POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Data Science and Engineering

Data Science Lab: Process and Methods Report
Exam session: Winter 2020

A double approach to review Sentiment Analysis

Emanuele Fasce
Student ID: S277983

INTRODUCTION

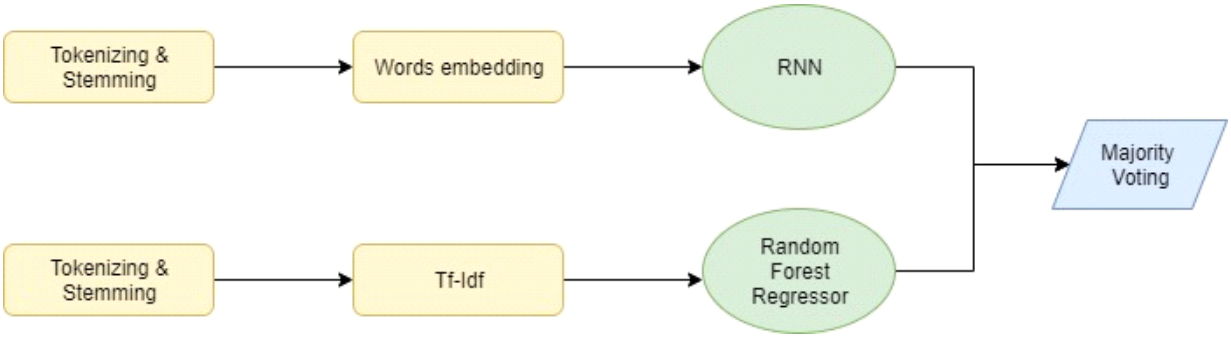
In this project I have built two sentiment analysis predictive models that classify Italian Trip Advisor hotel reviews as either positive or negative.

After reviewing some literature on sentiment analysis[1], I decided to focus my work on trying to extract "meaning" from the reviews, analyzing not only the occurrence of words but also the surrounding context.

Therefore, I have decided to present two approaches in this report: Approach 1 uses a Dense Neural Network and gets the best accuracy on the test set, Approach 2 combines a Recurrent Neural Network (RNN) and a Random Forest Regressor with the goal of combining the pattern recognition capabilities of the RNN with the frequency analysis provided by the regressor. Here I schematize my approaches for clarity.

APPROACH 1

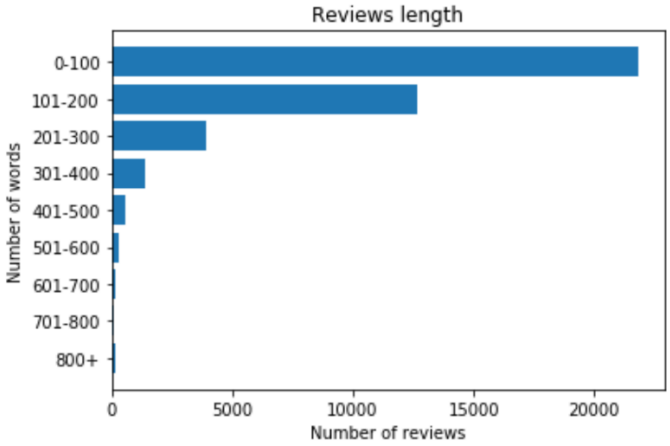
APPROACH 2



DATA EXPLORATION

The training set contains 28754 labeled reviews, 67.93% of which are labeled as positive while 32.07% as negative. The test set contains 12322 unlabeled reviews.

The length of the reviews is variable, as it can be seen from the graph below. This will be taken in consideration in the data processing step, when choosing the best data structure for the algorithms.

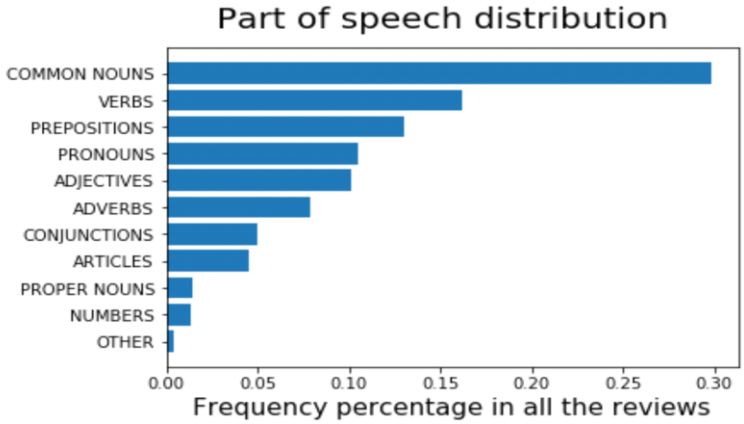


Every review is written in Italian, however some of them seem to have grammar mistakes as they were translated with Google Translate, probably because they were written by foreigners.

Since Italian is a complex language in which words can have several different suffixes, the usage of a stemmer seems to be required. The stemming process was implemented through Treetagger, a Python library that also recognizes different lemma types.

It could be interesting to analyze the frequency distribution of each lemma type and compare it with the distribution used in standard Italian.

From a 2004 paper[2], we can observe that the written Italian language averages at a noun frequency percentage of 25% and a verb frequency percentage of 15.8%, while the spoken Italian language averages at 15.7% and 20%, respectively. This contrasts with the available data, that shows instead a lemma type distribution skewed towards nouns, as shown in the graph below.



There are two possible explanations for this difference:

1) Treetagger has a tendency to wrongly classify past participles as nouns.
2) Since reviews are evaluations, people tend to use a highly descriptive language.

The frequency of interjections, which could help in classifying reviews, is unfortunately very low.

By analyzing the reviews singularly, many of them seem hard to classify even for a human reader, given how many of them list both positive and negative aspects regardless of the final vote given by the user. For this reason, an approach that takes the order of the words in consideration could be effective.

PREPROCESSING

Punctuation was removed in both approaches, and the Treetagger library was used to extract the stem of every word in each review.

- **Approach 1:** following the first step, a Tf-Idf matrix was by including as many features as possible, considering hardware constraints, so that the neural network itself could decide whether a word is useful for the analysis or not. A useful attribute of the Tf-Idf vectorizer is *ngram_range*, an attribute that takes into account groups of contiguous words as well as single words. The *ngram_range* was used to count the frequency of couples of words, as taking more would have been computationally unfeasible given the available hardware. After many trials, including and excluding stop words, selecting only the adjectives or the nouns, excluding the prepositions and conjunctions, it was observed that, in order to achieve a high accuracy on this dataset, including every word pays off more.
- **Approach 2:** a Tf-Idf matrix was built as described in approach one, using a *ngram_range* of (1,1) due to the higher hardware workload linked with the random forest algorithm. The resulting matrix was classified with a random forest in order to be able to analyze the most significant words. In order to feed the RNN, a vocabulary of all the words was created and each review was embedded in an array, with each number in the array representing a word. The selected size for the embedded arrays is 1837, given that such is the length of the longest review. Shorter arrays were padded with zeroes. Padding was chosen instead of shortening reviews to a fixed size because the approach proved to be computationally feasible.

ALGORITHM CHOICE

During the first trials, the TF-IDF and CountVectorizer classes were used with several different standard classifiers, but, even though some worked satisfactorily, none of them seemed to reach a high enough accuracy for the leaderboard, thus pushing the implementation towards more complex approaches that could better model the complexity of the Italian language. This was achieved by a review of the existing literature on sentiment analysis.

Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), often used in conjunction with the Word2Vec library, were quickly highlighted as two of the most widely used and successful approaches. Such approaches were replicated to see if they would yield similarly significant results.
The CNN implementation started by building Numpy arrays of size 700x50, each used to store a review, through Word2Vec, with each row representing a word in the review. The convolutional part was simple, involving a single convolutional layer and a single average pooling layer. The CNN approach, however, was discarded due to long training times, only reaching a maximum accuracy of 90%. The simple RNN approach yielded a similar result, while the Long Short-Term Memory (LSTM) approach performed better.

A simple, yet effective model that was implemented following the previous approach was a dense Multi-Layer Perceptron (MLP) network formed by two hidden layers, used in conjunction with the Tf-Idf matrix. Having therefore experimented with several different approaches, two different ones were selected for this report. The first one is the MLP dense neural network, the one which yielded the highest accuracy, and the second one is the RNN/random forest combination, where the two models were used together to build the predictions. Since all models were doing regression, the predictions were summed, choosing 1 as the cutoff value.
This approach attempts to model the complexity of the Italian language and to find the most useful words when doing a prediction thanks to the regression model, in order to be able to interpret a part of the results.

TUNING AND VALIDATION

Since complex models that are not completely covered in this course have been used, they were implemented with the goal of limiting overfitting while respecting standard implementations without deviating too much from standard, pre-existing models.
Overfitting can be avoided by using a contained number of neurons, layers and epochs.
The criteria used to decide when to stop training are the difference between the validation set accuracy and the training set accuracy, and the decreasing rate of the loss function.
For every neural network, 20% of the training set was used as a validation set.

- **Approach 1:** a very simple MLP was used, with 128 neurons in the input layer and 64 in the hidden layer, separated by dropout layers with dropout rate 0.5, with the Rectified Linear Unit (ReLU) activation function. The loss function decrease is very steep in the first epoch and the validation set accuracy grows quickly for the first epoch, but in the second one they both reach a plateau, thus leading to the decision to stop the training phase.
- **Approach 2:** Standard hyperparameters where chosen instead of computationally intense grid-search, since the main goal was to build a meaningful, properly working model. Therefore, a bidirectional LSTM with 64 cells and an input layer of 128 neurons was used. Due to hardware constraints, the training was limited to one epoch, as the provided results were satisfactory. The random forest was built with 200 decision trees, leaving all the other parameters as default.

DISCUSSION OF THE RESULTS

APPROACH 1

The Dense Neural Network has the best F-score (0.972), thus appearing to be a good approach to model the non-linearity found in this kind of problems.
An analysis of the results built with the WordCloud library is reported below. The first picture represents the most frequently used adjectives and nouns in the reviews labeled as positive, while the second one represents the adjectives and the nouns in the reviews labeled as negative.

APPROACH 2

The F-score reached with the Random Forest Regressor is 0.90.
However, this approach was chosen since it allows to have a look at the most significant words used by the algorithm.

While one could expect the most important features to be adjectives, several verbs and nouns appear to be meaningful as well. A similar analysis carried by only considering adjectives showed that the most significant ones were also the ones with the most extreme meaning. The same can be said for nouns and verbs.

Reported below is an unordered list of the words detected as being most meaningful by the random forest regressor, in both a positive and a negative way:

[immacolato, interfacciare, affiliare, disperatamente, stufetta, vanto, trafficatissimo, praga, senonche, rado, raggrinzire, antisettico, thriller, buonanotte, navata, scippo, viette, vandalico, commovente, indicibile, sarcastico, alchimista, neve, poveretto, ventoso, stimato, rinfrescante, strombazzare, abbuffare, nonostante, miliare, ottantenne, meridionale, mutevole, concorrenza, incertezza, nemmeno, supertecnologico, autonomo, autoinfliggere, whatsapp, esterno, semipulite, splendere, navigazione, rivalutare, snervante, banchetto, blogger, carcere, riparatore, hit, italiano, audace, presentabile, avvertire, terrorista, immangiabile, sconfinare, asfalto, sopravvivere, suntuosi, prosperare, inserviente, marginale, immigrato, polverose, orrendo, mortale, spacciare, alterare, ardentemente, mascalzone, sbavatura, amore, scricchiolio, voglioso, prode, esorcista, munire, inefficienza, sollecitare, portachiavi, neonato, strisciante, bottiglietta, fusibile, estendere, mercante, apparecchio, pellegrinaggio, trentenne, rigore, intervallo, nuovamente, deplorevole, sterilmente, news, ovation, ripetuto, caloria, cenno, sovrintendere, scassatissimo, lubrificazione, ciuffo, indolente, scollato, scalzo, adoperare, tempaccio, inclassificabile, atterrire, svilire, outsourcing, spazzino, caciarona, carcerato, inquadrare, cavolata, abusare]

Even though the random forest and the LSTM performed averagely when used separately, they reached an accuracy of 0.943 when used together. This shows that such an approach is viable and offers reliable results.

REFERENCES

1) https://www.researchgate.net/publication/334518326_Sentiment_Analysis_from_Movie_Reviews_Using_LSTMs
2) http://www.parlaritaliano.it/attachments/article/581/Voghera_Parti_del_discorso_2004.pdf