Bag of Authors

Abdallah Zaher, Cristina Mousinho and Gabriel Ravi
M20190684, M20190303, M20190925.

1 Introduction

If there's a trauma, what most portuguese adults will share is reading the +500 pages of "The Maias: Episodes of Romantic Life" by Eça de Queiroz. This year's Text Mining group project might just be what we need to extract what this book is talking about without reading the whole thing. The main goal here is to apply and develop a model capable of understanding those who wrote a text excerpt with 500 or 1000 words.

To accomplish that, we received two folders, one containing six random texts of 500 words and another containing six random texts of 1000 words. These texts were written by Almada Negreiros, Camilo Castelo Branco, Eça de Queiroz, José Saramago, José Rodrigues Dos Santos and Luísa Marques Silva, six portuguese authors.

With the aim of classifying six new texts, we built an algorithm based on NLP techniques that we learned during classes and their performances were then evaluated.

The system was implemented using Python 3, and more than 7 of its libraries.

2 Method/Approach

For this project, we will begin by researching the authors and their writing styles, and we will also clean the provided data, followed by preprocessing it, doing feature engineering, and building the machine learning algorithm, which we will then use to predict the authors of the texts. In addition, we will check its performance using evaluation metrics.

To clean the data, firstly, we removed all headers and footers, as they didn't contain information we considered useful and to prevent bias information to intervene in our algorithem's accuracy. Because we couldn't find a specific pattern to do it, the elimination had to be done manually.

By then, we had a dataframe containing 63 training texts and their respective labels. We proceeded to remove all portuguese stop-words and tags. Following this, we steamed, lemmatized and we lowercased the texts.

We went back and forth between keeping or getting rid of punctuation and numerical data. Since our research on the authors brought up that punctuation is very specific to each one of them (if you are familiar with José Saramago's writing style, you will for sure acknowledge the absence of punctuation), we decided to keep most of it, eliminating the ones we considered irrelevant. We used the same logic for numerical data.

Before any feature engineering, we decided to do a descriptive analysis on each text to understand a little more about each one of the authors. At this time, we had 20 texts by Camilo Castelo Branco but only 5 by Eça de Queiroz. We had about 300 words in each file by Almada Negreiros, but more than 9000 in Luísa's part.

Adding to that, using TF-IDF, we noticed that some authors contained unigrams, bigrams and trigrams more relevant than others. For example, the trigram "tertuliano maximo afonso" appears 600 times in José Saramago's texts, and these appearances are at least 10 times more than any text written by Eca de Queiroz.

Because of all of that, we decided to standardize the database, and to apply a **systematic sampling** (a probability sampling method in which sample members from a larger population are selected according to a random starting point but with a fixed, periodic interval). Before moving to the algorithm, we:

- Created new labels on the dataframe;
- Defined a function to break a big text in smaller segments (systematic sampling);
- Converted a list of segments being evenly distributed by the number of elements in each one of them.

The result of the last step is 6 dataframes, each one of them containing several texts with size between 500 and 1000 words.

We recleaned the obtained segments just to guarantee that there was a clean finalized dataframe without any stop words.

And finally, to guarantee that the final dataframe is fully homogeneous and to confirm that we can make inferences, we extracted some random segments, guaranteeing that each author contains a proportional amount of segments based on the size of the original text. This step is one of the most important steps because it'll reveal a massive difference regarding the predictions.

Eventually and after all the preprocessing and the separation of the rows, we ended up with a well distributed data frame composed of 600 rows.

With this new and clean dataframe, we extracted features using a bag of words technique to get a Countvectorizer matrix size equal to (600, 10000) and a label vector size of (600).

We distinguished that when sampling weights, such as words, a "simple k-fold" cross-validation will result in ignoring instances from the minority class. In our case every instance is significant and therefore it is better to apply **Stratified K-Fold.**

Afterwards, a comparison of accuracies will take place between **Train-Test split**, **Kfold CV** and **Stratified K-Fold.**

After cross validation we had a train set (75%) and a test set (25%). We wanted to find the algorithm that best fitted our case study, so we implemented the **K-Nearest Neighbors** and the **Naive Bayes algorithms.** Also, to find the best combination of parameters, GridSearch was also implemented.

Because we were interested in both true positives and true negatives, none of which more than the other, the statistical measure that we used to choose the best of the applied models was the **accuracy**. To calculate it, we resorted to the package *sklearn.metrics.*

At the end, to predict the authors of the 500 and the 1000 words texts, we applied the best method and the algorithm with the best accuracy.

3 Results and Discussion

As mentioned before, in this part, we will shed the light on the comparisons between the Cross Validations and the algorithms, and we will choose the model with the higher accuracy.

3.1 Results

Firstly, we would like to test which cross validation approach is the best one, so we applied a **basic** K-Nearest Neighbors with 3 neighbors and leaf size equaling 5.

	kFold	Stratified kFold	Train-Test
Accuracy	48%	63%	55%

After that, we decided to apply Stratified kFold. Our goal here was to verify the accuracy of this CV based on the number of splits. After using grid search, the three best combinations we achieved were 5, 4 and 2 splits. So we will use 5 splits for the next steps.

And now we need to decide the best algorithm aligned with the most suitable parameters. We did a GridSearch in the KNeighborsClassifier parameters (n_neighbors and leaf_size) and the best three combinations were:

Parameters (neig,leaf)	(5,30)	(4,20)	(1,5)
Accuracy	95%	89%	88%

When using kNN, the best accuracy we got was 95%.

But now we want to compare our obtained results by testing the same dataframe using Naive Bayes algorithm and the best accuracy acquired was 81%.

We then applied the final model to the texts, in order to predict which text belongs to which author.

The result of the folder that contains 6 texts with 500 words was: the first text was written by José Saramago, the second by Almada Negreiros, the third and fourth by Luísa Marques Silva, the fifth by Camilo Castelo Branco and the sixth by José Rodrigues dos Santos.

And the result of the folder that contains 6 texts with 1000 words had almost the same results, with the exception of the fourth text, now associated with Eça Queiroz, instead of with Luísa Marques Silva.

3.2 Discussion

We took some time to realize that cleaning, sampling and extracting texts are the most important factors to find good accuracy.

When we applied different algorithms with different parameters itselves, the accuracy didn't change more than when you start with a good sampling and cleaning.

Finally, for text databases, the **Stratified kFold** CV is a good technique because it forces each fold to have at least m instances of each class. The **KNN** is a useful approach because it can detect non-linear distributed words and it tends to perform very well with a lot of data points.In addition, when you combine an effective algorithm with the most suitable parameters (GridSearch), you will achieve a good final model and consequently good prediction.

4 Conclusion

The best approach to associate texts to their authors turned out to be: doing systematic sampling to extract segments of the data set, using Stratified Kfold to build train/test/validation sets and applying the KNN algorithm with neighbor size 5 and leaf size 30.

References

"Almada Negreiros." *Wikipedia.Org*, Fundação Wikimedia, Inc., 14 June 2005, pt.wikipedia.org/wiki/Almada_Negreiros. Accessed 22 Feb. 2020.

"Almada Negreiros.". *Instituto-Camoes.Pt*, 2020, cvc.instituto-camoes.pt/literatura/almada.htm. Accessed 22 Feb. 2020.

"Anti-Clericalism." *Wikipedia*, Wikimedia Foundation, 16 Mar. 2020, en.wikipedia.org/wiki/Anti-clericalism. Accessed 22 Feb. 2020.

"Camilo Castelo Branco." *Wikipedia.Org*, Fundação Wikimedia, Inc., 30 June 2004, pt.wikipedia.org/wiki/Camilo_Castelo_Branco. Accessed 22 Feb. 2020.

Ciberdúvidas/ISCTE-IUL. "Saramago, o Escritor Que Brinca Com a Pontuação - O Nosso Idioma - Ciberdúvidas Da Língua Portuguesa." *Iscte-Iul.Pt*, 23 Apr. 2008, ciberduvidas.iscte-iul.pt/artigos/rubricas/idioma/saramago-o-escritor-que-brinca-com-a-pontuacao/1691. Accessed 23 Feb. 2020.

"Eça de Queiroz." *Wikipedia.Org*, Fundação Wikimedia, Inc., 8 Mar. 2004, pt.wikipedia.org/wiki/E%C3%A7a_de_Queiroz. Accessed 23 Feb. 2020.

"Futurismo." *Wikipedia.Org*, Fundação Wikimedia, Inc., 5 Nov. 2004, pt.wikipedia.org/wiki/Futurismo. Accessed 22 Feb. 2020.

Gh. "Futurismo,Como Surgiu ?" *Blogspot.Com*, 2013, futurismobbd.blogspot.com/2013/09/o-futurismo.html. Accessed 22 Feb. 2020.

"José Rodrigues Dos Santos." *Wikipedia.Org*, Fundação Wikimedia, Inc., 14 Feb. 2006, pt.wikipedia.org/wiki/Jos%C3%A9_

Rodrigues_dos_Santos. Accessed 23 Feb. 2020.

"José Saramago." *Wikipedia.Org*, Fundação Wikimedia, Inc., 9 Apr. 2004, pt.wikipedia.org/wiki/Jos%C3%A9_Saramago. Accessed 23 Feb. 2020.

"Luísa Marques Da Silva - Saída de Emergência." *Saidadeemergencia.Com*, 2020, www.saidadeemergencia.com/autor/luisa-marques-da-silva/. Accessed 23 Feb. 2020.

Marques Silva, Luísa. "Histórias Para Ti!" *My-Free.Website*, 2020, wnfe.my-free.website/. Accessed 23 Feb. 2020.

"Modernismo Em Portugal." *Wikipedia.Org*, Fundação Wikimedia, Inc., 10 May 2008, pt.wikipedia.org/wiki/Modernismo_em_Portugal. Accessed 22 Feb. 2020.

"Romanticism." *Wikipedia*, Wikimedia Foundation, 18 Mar. 2019, en.wikipedia.org/wiki/Romanticism. Accessed 22 Feb. 2020.