Machine Learning Engineer Nanodegree

Capstone Proposal

Thiago Vieira May 31st, 2019

Proposal

Domain Background

The goal of this project it's to build a model for authorship attribution classification using as a background the work developed in VJO2011 e OOJ2013 that is available in The Laboratory of Vision, Robotics and Imaging of Federal University of Parana.

The use of electronic documents like e-mails continue to grow exponentially, and even though reliable technology is available to trace a particular computer/or IP address where the document has been produced, the fundamental problem is to identify who was behind the keyboard when the document was produced (OOJ2013). Practical applications for author identification have grown in several different areas such as criminal law (identifying writers of ransom notes and harassment letters), civil law (copyright and estate disputes), and computer security (mining email content).

This problem is very relevant for my work since I'm developing several NLP classification models to label court decisions and the importance of a case to the federal attorney. I intend to apply the knowledge obtain from this project to my work and share it, as well.

Problem Statement

Authorship attribution can be defined as the task of inferring characteristics of a document's author from the textual characteristics of the document itself. The challenge here is to estimate how similar two documents are from each other, based on patterns of linguistic behavior in documents of known and unknown authorship. This is known in the literature as authorship attribution or authorship analysis.

Datasets and Inputs

The Authorship Attribution Database (AAD) contains short articles from 100 different authors whose texts were uniformly distributed over 10 different subjects:

- Miscellaneous;
- Law;
- Economics;
- Sports;
- Gastronomy;
- Literature;
- Politics;
- Health;
- Technology;
- Tourism;

The sources were 15 Brazilian newspapers located all over the country. We have chosen 30 short articles from each author, thus summing up 3000 pieces of documents. The articles usually deal with polemic subjects and express the author's personal opinion. In average, the articles have 600 tokens (words) and 350 Hapax (words occurring once). One aspect worth of remark is that this kind of articles can go through some revision process, which can remove some personal characteristics of the texts. Besides, authorship attribution using short articles poses an extra challenge since the number of features that can be extracted are directly related to the size of the text.

The dataset can be downloaded in this link.

Solution Statement

The solutions for this problem will be based on several supervised learning methods and by testing different approaches for feature extractions/representation. The best solution will be the one that has the best score on selected metrics. It's going to be an experimentation analysis to find the best solution. Since the work that this project is based is before the boom of CNN and RNN, I hope to get some interesting results.

Benchmark Model

In the image below, it's shown some works on authorship attribution published in the literature. Comparing different works is not a

straightforward task since most of the works use different databases and classifiers.

Published works on authorship attribution.

Ref.	Classifier	Database	Rec. rate (%)
[23]	SVM	Web pages	66–80
[7]	SVM	German newspaper	80
[10]	SVM	3 sister's letters	75
[24]	kNN	Novels	66–76
[5]	Distance	Brazilian novels	78
[19]	SVM	Brazilian newspaper	72
[6]	Bayes	Mexican poems	60–80
[21]	Bayes	Turkish newspaper	80
[25]	SVM	Brazilian newspaper	74

For this project, I'll use as a benchmark model the accuracy of the work in VJO2011 e OOJ2013, which were a 74% and 77%.

Since we have 100 authors in the database, analyzing the confusion matrix would be complicated, these works provided the analysis of the confusion matrix grouped by subject. Such a matrix can show that the recognition rate in terms of subjects is about 86% in VJO2011 and 80% in OOJ2013.

Evaluation Metrics

Based on the context of NLP and the multiclass problem property, I'll use the following metrics to compare models:

- Accuracy
- F1 score (trade-off between TP and FP);
- F1 score average;
- Recall;
- Precision
- Confusion Matrix.

Given the context of the problem and to replicate the same metrics used in the base references, I think this set of metrics is suitable to the problem because it's the most used in the NLP field.

Project Design

The project will be organized based on cookiecutter-data-science:

- A `data` folder containing the CSV files
- A `notebooks` folder containing the notebooks on insights and model training
- A `model` folder containing the final version of the model
- A `requirements.txt` with software requirements for this project

The methodology consists of a classic NLP classification pipeline:

- Text Exploratory Data Analysis (EDA);
- Text Preprocessing (Stopwords/Unnecessary tokens removal, Stemming, Lemmatization, etc...) + Features Extraction (BoW, TF-IDF, Hashing, Embeddings, etc);
- Run some classic methods (Logistic Regression, Naive Bayes, KNN, etc);
- Run some more advanced methods (Word2Vec, Doc2Vec, CNN, RNN, etc.)
- Evaluate the models against the benchmark;
- Discuss the results.

It's essential to notice that maybe not all algorithms described above are going to be tested, as I the goal of this project is not to test all of them.

Initially, I'll use the following Python libraries:

- Jupyter Notebook;
- Scikit Learn;
- NLTK;
- Spacy;
- pandas;
- Gensim;
- Google Colab;

- also, others that may be necessary.