Natural Language Processing and Machine Learning for Stylometry
Problem (9)

Mohamed Shawky Zaky AbdelAal Sabae
Section:2,BN:15
mohamed.sabae99@eng-st.cu.edu.eg

***Abstract*— Stylometry is the application of the study of linguistic style, usually to written language. In this work, we discuss the proper methods of using *natural language processing* and *machine learning* for binary classification of authors' writings.**

I. INTRODUCTION

The report contains the discussion of stylometry problem. We discuss the means of dataset collection, feature extraction and language modeling. Moreover, we mention the advantage of the submitted solutions over the other possible approaches. Finally, we review the code structure and usage. The provided code mainly contains two solutions. **First,** *Naive Bayes* classifier with *TF-IDF* features. **Second,** a simple *neural network* with *word embeddings*.

II. APPROACH

This section contains the discussion of *dataset collection*, *feature extraction* and *language modeling*.

A. Dataset Collection

Dataset is collected from *Kaggle's* **Spooky author identification problem**, as it contains authors of the same era and genre. Other authors are not considered mainly because no available data for multiple authors of the same era. The dataset contains three authors :

- **Edgar Allan Poe** *(EAP)* : 7900 phrases.
- **HP Lovecraft** *(HPL)* : 5635 phrases.
- **Mary Wollstonecraft Shelley** *(MWS)* : 6044 phrases.

B. Feature Extraction

Before feature extraction stage, some text processing is performed on the input phrases. Basically, **tokenization**, **stemming** and **lemmatization** are performed. **stemming** performs better than **lemmatization** in our case.
For feature extraction, five methods are considered :

- **Bag of Words** *(BoW)* : yields the worst performance.
- **n-grams** : offers moderate performance based on the classifier.
- **TF-IDF vectorization** : *n-grams + term frequency - inverse document frequency*, offers decent performance with most classifiers.
- **PCA / Truncated SVD on previous features** : using *principal component analysis* or *truncated singular value decomposition* on our features does not seem to perform well.
- **Word Embeddings** : basically using *GloVe* pretrained embeddings for *neural networks* training.

C. Language Modeling

The submitted solution only offers two methods *Naive Bayes* classifier as a classical language model and a *deep neural network* as a deep learning language model.

- **Classical language modeling : Naive Bayes** is used with *TF-IDF* vectorization features. This is basically because it yields better results than all the other considered classical approaches. The other classical approaches, considered in this work, are **support vector machines** *(SVM)*, **logistic regression** and **gradient boosting**.
- **Deep learning-based language modeling :** a simple *neural network* is used with *word embeddings*, in order to test the ability of neural networks on our dataset. The network consists of one **bidirectional GRU** layer followed by two **fully-connected** layer with **dropouts**.

III. CODE STRUCTURE

IV. CODE USAGE

V. EXPERIMENTAL RESULTS

VI. CONCLUSION

REFERENCES