Authorship Verfication for Arabic Poems

1st Yousef Jallad
Department of Electrical and Computer Engineering
Birzeit University
Birzeit, PS
1190030@student.birzeit.edu

2nd Sary Hammad
Department of Electrical and Computer Engineering
Birzeit University
Birzeit, PS
1192698@student.birzeit.edu

3rd Laith Sharia
Department of Electrical and Computer Engineering
Birzeit University
Birzeit, PS
1190651@student.birzeit.edu

4th Tareq Ewaida
Department of Electrical and Computer Engineering
Birzeit University
Birzeit, PS
1192363@student.birzeit.edu

***Abstract*— In this Project, we considered the issue of author verification, many foundational and important documents throughout history have had anonymous or fuzzy authorship. identifying the authors of such documents would open new avenues for research and uncover previously unknown aspects of an author or reveal misattributed documents. We focused on Arabic, a language that is largely unexplored despite the importance of this issue. We had tried deep learning and the initial results are almost excellent.**

I. Introduction

Authorship verification is a quite common task that is important in the analysis of historic documents. Traditionally, language, style, and vocabulary, as well as other characteristics of the text such as punctuation and grammar were analyzed to identify the author. Historical and cultural backdrops that could reveal information about the author's identity were also taken into account. All of these techniques involve human labor, which is time-consuming and results in high error rates. Thus, with current technology, we can use Machine Learning techniques to assist us in completing these procedures quickly and accurately.[1]

Arabic is a language currently spoken by hundreds of millions as a first language, as well as many uses in religious contexts, and it has been used to document some of humanities most important works in philosophy, medicine, law and poetry. Authorship verification for Arabic texts is quite important, but has its own challenges given that it’s a complex and intricate language.[2]

Numerous efforts have been done on classification and authorship analysis of Arabic writings. While text classification typically focuses on identifying the topic or area of a text, such as sports, politics, etc. in contrast, authorship analysis focuses on identifying actual authorship.

What distinguishes our project from others, is that it can verify the authorship of a particular document. It is not only intended to identify known authors based on linguistic patterns in the poems they have produced, but also to make an informed guess as to which authors in the training set were closest.

II. Methodolgy

In order to create a decent authorship verification system, a systematic and comprehensive technique in the field of natural language process (NLP) were employed. And this shown in the stages below:

A. Data Collection:

Through doing the project were in front many methods of collecting data, but we found that web scrabbing is the best option for us. This decision was primarily due to the fact that the most required poems were not readily format.

B. Preprocessing and Tokenization:

1) Preprocessing:

For preprocessing “Arabet preprocessing” were used. This tool is responsible for normalize and cleaning Arabic words, It also performs number of other tasks, including eliminating diacritical symbols and punctuation and changing Arabic numbers to English ones.

2) Tokenization:

This is the process of breaking down a paragraph and sentences into individual words or tokens.[7]

C. Machine Learing:

The system was trained on a corpus that we collected using web scrapping as mentioned before. The system may also predict the author for any entered document.

III. Implementaion

In order to implement the system, The fact that the Python programming language includes so many helpful libraries is just one of the numerous reasons why it is employed.

A. GPU Availability:

Firstly, the System checks the availability of the GPU, because it can significantly speed up the calculations, also its ability to perform parallel operations on large block data.

B. Loading Dataset:

The code loads poetry data from a CSV file called "poems.csv" into a Pandas Data Frame called "df"

C. Split Data:

Using the train_test_split function from scikit-learn, the dataset is divided into training and testing sets. For reproducibility, the test set size is set to 30% of the original dataset and a random seed of 42 is employed. The resulting testing set is kept in test_df, whereas the training set is kept in train_df.

D. *Defining feautures and target:*

- The coding identifies the feature as the column name of the poem's content, which is a list with a single element: ['poem_content'].
- The variable "poet_name" is the target variable, and it refers to the column that contains the poet's name.

E. *Loading Arbert preprocessing and Tokenization:*

As explained in the methodology section.

F. *Encoding labels:*

- The LabelEncoder class from Scikit-Learn is used to encode the labels (poet names). Each individual poet's name is given a number in this way.
- The encoded labels are stored in train_labels and test_labels.

G. *Moving Data to GPU:*

- The to function of the torch.tensor class is used to transfer the training and testing encodings (input IDs and attention masks) to the GPU (if available).
- The encoded labels are also moved to the GPU.

H. *Defining the Model Architecture:*

- The program installs the Hugging Face model hub's pre-trained Arabert model for sequence classification (AutoModelForSequenceClassification).
- The model will divide inputs into 5 categories as indicated by the value of the num_labels parameter, which is set to 5.
- The model is moved to the GPU.

I. *Setting Up Training Parameters:*

- The model parameters are supplied to the optimizer (AdamW), which is defined with a learning rate of 1e-5.
- The input IDs, attention masks, and labels are combined to generate the training dataset.
- The testing dataset is created in a similar way.

J. *Training the Model:*

- The model is trained for 5 epochs. In each epoch, the model is put in the training mode (model.train()), and the total loss is initialized to 0.
- The training dataset is iterated over, and for each batch, the optimizer gradients are set to zero (optimizer.zero_grad()).
- The input IDs, attention masks, and labels are moved to the GPU and then passed to the model.
- The forward pass of the model is run, and the loss is calculated.
- The gradients are computed and updated using backpropagation (loss.backward() and optimizer.step()).
- The average loss per batch is calculated and printed for each epoch.

K. *Evaluating the Model:*

- After training, the model is put in evaluation mode (model.eval())
- The total number of correct predictions is initialized to 0.
- The input IDs and attention masks are transferred to the GPU for each batch while the testing dataset is iterated over.
- To obtain the predictions, the model's forward pass is used.
- The frequency of accurate predictions is increased after a comparison between the predicted and actual labels.
- The accuracy is computed by dividing the total number of correct predictions by the length of the testing dataset, and it is printed at the end.

IV. Results and Evaluation

A Number of tests are conducted on the collected dataset to evaluate the accuracy, precision, and recall for each poet. And the result will be shown on report and confusion matrix.

The evaluation will be determined using the following equations:

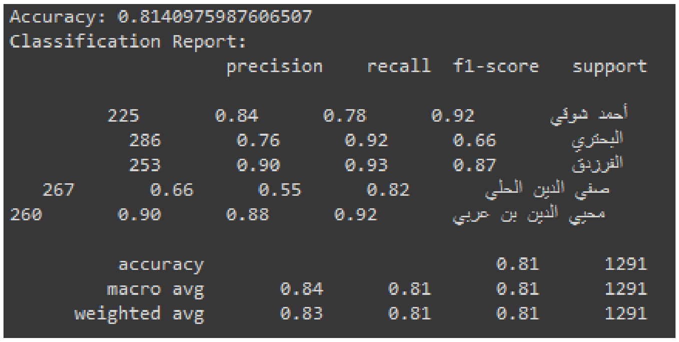
$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Where: TP, FP, TN, and FN are FN are the numbers of true positives, false positives, true negatives, and false negatives, respectively.

Also, the results for running our code were as follows:



```
Accuracy: 0.8140975987606507
Classification Report:
                    precision    recall  f1-score   support

         225       0.84      0.78      0.92       أحمد شوقي
           286       0.76      0.92      0.66        البحتري
           253       0.90      0.93      0.87        الفرزدق
   267       0.66      0.55      0.82       صفي الدين الحلي
260       0.90      0.88      0.92       محيي الدين بن عربي

          accuracy                          0.81      1291
         macro avg       0.84      0.81      0.81      1291
      weighted avg       0.83      0.81      0.81      1291
```

Figure 1: Results of simulation

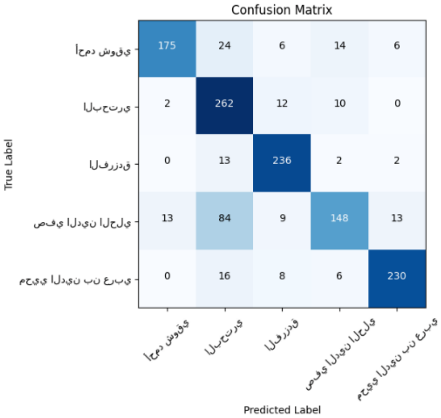


Figure 2: Confusion matrix

This confusion matrix The accuracy scores for identifying a particular author of a poem. he first document indicates that the accuracy for authorship by Ahmad Shawqi is '175' - a relatively high number. on the other hand, Al-Bahtari, the accuracy is only '2', suggesting a low likelihood of authorship.

V. Conclusion

In this paper, only a narrow avenue of author verification was explored using a narrow set of features that don't explore the particular linguistic aspects of the texts, even with these small features and using ARBERT pre trained model we were able to obtain an accuracy of 81.4%. There is certainly room for improvement in regards to feature extraction.

VI. Possible Improvements

The main avenue for improvement in this project would revolve around feature extraction. Features pertaining to the subject matter in the documents could be utilized by collecting word sets for common poetry subjects and finding the frequency of these subjects in the texts. Many papers on this subject have suggested using deep-learning methods to skip expert feature extraction which could be time consuming or very specific.

VII. References

[1] https://aclanthology.org/N04-4038.pdf

[2] http://www.med.mcgill.ca/epidemiology/hanley/bios601/Applications/StatisticsGuideToUnknown3rdEdition/part%203,%20our%20social%20world/mosteller,wallace_deciding_authorship(115).pdf

[3] https://www.tokenex.com/blog/ab-what-is-nlp-natural-language-processing-tokenization/#:~:text=Tokenization%20is%20used%20in%20natural,into%20understandable%20parts%20(words).

[4] https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall#:~:text=Accuracy%20shows%20how%20often%20a,when%20choosing%20the%20suitable%20metric.

[5] https://www.aldiwan.net/

[6] https://huggingface.co/aubmindlab/bert-base-arabert

[7] ENCS 5342 Lecture Notes.

[8] https://github.com/aub-mind/arabert