Classification of Time Period and Author Age in Fiction

ABIGAIL HODGE, Northeastern University
SAMANTHA PRICE, Northeastern University
ZIJIN HUANG, Northeastern University

ACM Reference Format:
Abigail Hodge, Samantha Price, and Zijin Huang. 2019. Classification of Time Period and Author Age in Fiction. 1, 1 (March 2019), 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 CHANGES

There have been no changes made to any aspect of the project since the initial proposal (as of yet). We have currently established a baseline linear regression model to determine age (discussed below). Based on the preliminary results we have received and future research, we will ultimately determine if we want to keep the linear regression model or choose an alternative approach. Otherwise, the goal of the project, the methodology, the datasets, and the evaluation techniques are still consistent with what we outlined in the initial project proposal.

2 PREPROCESSING

2.1 Extraction

The first step in preprocessing the dataset was to analyze the Project Gutenberg [1] catalog and extract relevant authors and books according to the following criteria: originally written in English, fictional, single book (not collection), known publication date, known birth year of the author and undisputed authorship. After selecting appropriate authors and texts from the catalog, texts were pulled from the Gutenberg database with `wget` and canonical URLs for books. Basic data cleaning such as removing unnecessary separators and page numbers followed.

2.2 Tagging

Then, the author and books were tagged with age and time periods from Wikipedia summaries. We modified the age categories from PAN at CLEF to better suit our corpus for fictional books and arrived at the following age categories: 18-24, 25-34, 35-49, 50-64, 65-xx. We adjusted the time periods based on past research and historical literature movements: 1700 - 1750, 1751 - 1800, 1801-1810, 1811-1820, 1821-1830, 1831-1840, 1841-1850, 1851-1860, 1861-1870, 1871-1880, 1881-1890, 1911-1920. Wikipedia summaries were extracted automatically using a Python package.

2.3 Issues and Directions

One potential issue we noticed was the large gap between the year a book is written and the year a book is published. We will be discussing which to exclude, and how to remove irrelevant year information. Moreover, since there is no well-formatted, extractable data from Wikipedia, the publication year has to be either manually tagged or implemented with a question-answer system. We will expand our database to a larger corpus that is well distributed across the above time periods and age categories in the future.

3 METHODS

3.1 Feature Selection

For this report we only analyzed age and therefore used features that have been successfully implemented in age profiling research. Argamon et al. (2009)[5] and Nguyen et al. (2011)[3] demonstrated that unigram counts and POS-tag counts are powerful tools in this task, so we decided to focus on these for our initial feature vectors. We extracted the 50 most common unigrams in our training set using the frequency distribution functionality in the Natural Language Toolkit, and used NLTK's default POS-tagger to get counts for 35 tags. Each count was normalized by the length of the document.

3.2 Model Description

At this point in the project, we have chosen a simple linear regression model to predict the age of a given author. The general structure of the model was developed based on a variety of sources that demonstrated how to implement basic linear regression with PyTorch, the machine learning library [4] [6]. The model utilizes Stochastic Gradient Descent to optimize the weights in the linear regression formula and reduce the loss between the predicted ages and target ages. The loss is represented by the Mean Squared Error.

3.3 Rationale for Model

Two major factors influenced the choice of linear regression: ease of implementation and proven efficacy. Regarding ease of implementation, PyTorch has convenient functionality to create a linear regression model, optimizer, and loss calculator. We may decide that a more complex model like a neural network will be necessary when we endeavor to predict both age and time period; for now, we wanted to observe the accuracy of the results we could achieve with the simplest possible model. The analysis of our present results is delineated in Section 4. In terms of efficacy, the utilization of the linear regression model for age prediction in nonfiction writing has

Authors' addresses: Abigail Hodge, abigailhodge98@gmail.com, Northeastern University; Samantha Price, price.sam@husky.neu.edu, Northeastern University; Zijin Huang, elizazijinhuang@gmail.com, Northeastern University.

XXXX-XXXX/2019/3-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

been demonstrated in Nguyen et al. (2011)[3]. Due to their relatively reliable results, we decided that applying a similar strategy to fiction was a logical decision; thus we created a preliminary linear regression model that can be be fine-tuned (or changed completely) later on.

4 RESULTS AND EVALUATION

4.1 Results

Currently, the trained model can output a predicted age for a selected author given a piece of text by that author. The authors used to train the model were Jane Austen and Charles Dickens. These authors were chosen because they were prolific during their lifetime and had distinct writing styles. Also, Project Gutenberg possessed numerous novels from both Austen and Dickens that could be easily accessed. All novels from the two authors available on Project Gutenberg were used for training, except for Austen's *Mansfield Park* and Dickens' *Oliver Twist*, which were used for testing. Examples are illustrated below demonstrating the predicted ages produced by the model with the test text of each author employed as input.

```
Jane Austen
Age predicted for Mansfield Park: 35.5429

Charles Dickens
Age predicted for Oliver Twist: 35.5576
```

The result for Austen is somewhat promising, as the author was 39 when *Mansfield Park* was published. On the other hand, the result for Dickens is skewed dramatically, since he was around 25 years old when *Oliver Twist* was first published. A quantitative evaluation of the results is discussed in the next section, and the major reasons for the results are discussed in Section 5.

4.2 Evaluation

After intrinsically evaluating our model with test novels, we used the "goodness" formula (described in our project proposal) to determine our model's efficacy thus far. When we applied the formula to the model's predicted ages and the actual ages for Austen and Dickens, the outcome was the following:

```
Jane Austen
Goodness Metric: 0.2306610107421875

Charles Dickens
Goodness Metric: 1.0557559967041015
```

The goodness metric should be a normalized value between 0 and 1, where the closer to 0 the value is the more accurate the model is. In the case of Jane Austen, the metric is in the appropriate range and on the lower side, which is a positive sign for our model. However, the metric for Charles Dickens is out of range and high, an indication that there are definite changes that need to be made to both the model and other aspects of the project. Sections 5 and 6 describe the actions that we will take to improve this metric and the predictive power of our model for both age and time period.

5 WHAT IS WORKING AND WHAT IS WRONG

In terms of what is working, we have a model that can extract POS-tag and unigram frequencies, and leverage those frequencies to predict an author's age from a text. We have created a rudimentary pipeline to take in data, pre-process it, run necessary feature extraction, and run said features through training for a linear regression model. This base will be crucial in our future efforts to improve and expand our model.

Unfortunately, the predicted age is not yet accurate. There are three main causes for this. First, we do not have a large training set, which means that a single stylistic outlier could throw off our numbers by a wide margin. Second, we have only picked very simple features, POS-tags and unigrams, and could look into leveraging more complex syntactic or stylistic features going forward. Third, we utilized a simple linear regression model, which might not be the best model for this task. It might be better to use an SVM model or neural network, for example. Going forward we will be examining all three of these points of failure to improve our predictions.

6 FUTURE WORK

There are a number of ways in which the current model for age prediction could be improved. To begin with, the feature vector from the training data that is submitted to our model could be modified in order to increase overall accuracy. Currently, the only unigrams examined are the most frequent words found in the training set. However, small frequencies should not be removed entirely, as there could be valuable information revealed about an author in both the structures they chose not to use and the structures they chose to use. The feature vector produced from the training data will also be expanded with additional features. For example, sentiment analysis might be helpful, as per Nguyen et al. (2011) [3], younger writers tend to utilize more negative emotions.

The most obvious next step is to implement a model to predict the time period in which an author wrote a novel. First, we will need to develop feature vectors for this problem. We will likely create a word embeddings classifier to analyze the training data, as word embeddings can be valuable predictors of the time period in which a work was written. We will also be considering neologisms utilizing the techniques discussed in Garcia-Fernandez et al. (2011)[2]. Next we need to create a model (potentially a neural network) so we need to perform additional research about the most appropriate structure of the model. Based upon the success of the neural network for time period prediction, we may consider utilizing a neural network for age as well. In that case, we would have to modify the existing linear regression implementation. Once we officially modify our models, we will calculate additional evaluation metrics: precision, recall, accuracy, and the F-score for the combined prediction of age and time period.

REFERENCES

[1] [n. d.]. Project Gutenberg. Retrieved February 07, 2019 from http://www.gutenberg.org/

[2] Marco Dinarelli1 Anne Garcia-Fernandez, Anne-Laure Ligozat and Delphine Bernhard. 2011. When Was It Written? Automatically Determining Publication Dates. *String Processing and Information Retrieval Lecture Notes in Computer Science* (2011). https://perso.limsi.fr/annlor/docs/spire11.pdf

[3] Noah A. Smith Dong Nguyen and Carolyn P. Rosé. 2011. Author Age Prediction from Text using Linear Regression. *LaTeCH '11 Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities* (2011), 115–123. hhttp://delivery.acm.org/10.1145/2110000/2107651/p115-nguyen.pdf?ip=71.174.251.115&id=2107651&acc=OPEN&key=4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E6D218144511F3437&__acm__=1552696258_041e7e8ac84ce4b4aaf78893d9f88543

[4] Aakash N S. [n. d.]. PyTorch basics - Linear Regression from scratch. Retrieved March 15, 2019 from https://www.kaggle.com/aakashns/pytorch-basics-linear-regression-from-scratch/comments

[5] James W. Pennebaker Shlomo Argamon, Moshe Koppel and Jonathan Schler. 2009. Automatically Profiling the Author of an Anonymous Text). *Commun. ACM* 52, 2 (2009), 119–123. http://u.cs.biu.ac.il/~koppel/papers/AuthorshipProfiling-cacm-final.pdf

[6] Somnath. [n. d.]. Linear Regression using PyTorch. Retrieved March 15, 2019 from https://www.geeksforgeeks.org/linear-regression-using-pytorch/