CS-534
5 December 2016
Artificial Intelligence

Inferring Characteristic Traits of an Author using Text Analytics

Aditya Gupta, Anand Parwal, Namrita Madhusoodanan, Praneeta Mallela

Department of Robotics Engineering, Worcester Polytechnic Institute, MA, USA

Abstract
This paper presents a method to predict the characteristic traits of an author of a given book. Characteristic traits - like gender, race, age of the author at the time of publishing, the genre of the book, and other socioeconomic and psychological factors - which influence the content and style of writing of an author can be predicted given his/her work. Multiple feature vectors (tf-idf, n-gram, POS tagging) were extracted from texts using preprocessing techniques (stopword removal, stemming, lemmatization) and a comparative performance study for every technique was analyzed by applying six classification methods (namely Naïve Bayes, SVM using gradient descent, k-NN, Random Forest, Passive Aggressive, Perceptron). These analytic studies were performed with the aid of the sklearn machine learning library and the Natural Language Toolkit. The performance of the classifiers was further compared for data with and without preprocessing.

1. Introduction

Writing is often influenced by a myriad of factors such as the author's personal life, gender, race, socioeconomic status, their unique individual experiences and other literary and non literary influences. These factors cannot be easily deduced by the reader. With the advancement of technology these task are being efficiently done by computers using different machine learning techniques.

Authorship attribution is the task of linking a text to a particular author. While author attribution enables author identification, author profiling infers an author's characteristics using language/text as evidence and by retrieving information out of it. These two parts together are a research area of growing importance, with a wide range of applications in forensics, marketing in social media, internet security, threat identification, uncovering plagiarism and authorship. For example, if a text poses any threat, then having basic information about the author through profiling can act as prima facia for investigation. It can further be used to narrow down the list of possible culprits. In addition, with increasing popularity of e-commerce market, this information can be used by digital marketers and advertising company.

In this paper, we present a method for author profiling. Author profiling for Age and Gender was performed on two datasets - chat room conversations and blogs provided by the PAN 2013 Author Profiling organizers and the other on a custom dataset of published and edited books. By analyzing text data, extracting features like tf-idf (term frequency-inverse document frequency), n-grams and POS (part of speech) tagging and experimenting with classification algorithms, we predict the demographic profile of an author based on factors such as gender and age. We also compare the results obtained by using various combinations of data preprocessing and learning techniques.

2. Background

Authorship attribution is a challenging task, which dates back to the mid-twentieth century. Significant early work includes authorship attribution studies on Shakespeare's works [8] or the Federalist Papers [9]. This area has gained a lot of attention in the past couple of years. The conventional approach is to analyze a long anonymous text and attribute it to a small closed set of authors, whose writing samples exist as datasets [6].

In 2003, Koppel et al [10] used patterns in writing style to classify authors into specific demographics. They used a group of function words (i.e. prepositions, pronouns, auxiliary verbs) and part-of-speech analysis to predict the gender of the author and classify the text into fiction and nonfiction. He proposed combinations of simple lexical and syntactic features for this task, and achieved an accuracy of 0.80. In 2006, Scheler et al [13] performed classification of writers into groups based on age and gender by using over 71,000 blogs. For this purpose, they used stylistic features such as non-dictionary words, parts of speech, function words and hyperlinks, and certain content based features. They achieved an accuracy of 0.80 in identifying gender and 0.75 in identifying age. Content based features helped identify trends such as males write mostly about politics and technology and females write about relationships. In 2011, Peersman et al [12] carried out a similar study on social media networks, using features such as length of text, emoticons, sequence of words such as unigrams and bigrams, using an SVM classifier.

Other related work included using a cosine based similarity metric to measure the distance between two documents and using the same to attribute an anonymous document to an author whose known document it was most similar to [3]. In 2013, the PAN workshop series proposed a task to classify authors based on gender and age group. A set of blogs was given as the input. The best participant [11] achieved an accuracy of 0.59 while predicting gender and an accuracy of 0.65 while predicting age.

Author attribution using the techniques that were employed in the above mentioned approaches may not always be practical, as the author may not belong to the known set of authors used for training. Furthermore, the writing samples for each author in the dataset may be inadequate. Attribution is generally carried out on small datasets, and gets really difficult for larger datasets. As a result, attribution is not very useful in a real life scenario such as in the field of forensics [5]. In this paper, we use an alternative to this approach, namely author profiling and compare its accuracy across different classifiers and different preprocessing techniques. We attempt to predict characteristic traits of an author which can provide clues to help uncover the author's identity. Author profiling tackles the challenges faced by authorship attribution, as it is possible to find these clues about the author even when the data specific to the author in question is absent from the training dataset [1][2].

The challenges faced during author profiling are in the data retrieval phase, such as acquiring labelled data, as manual labelling needs to be done quite often to fill in gaps in the data [7]. Furthermore, acquiring a large database of text data is a mammoth task, as there are not many free sources of data. The datasets also tend to be highly skewed. In addition, the analysis is limited to the books written in the English language.

In this paper, a brief description of the dataset and libraries used is given in Section 3. The preprocessing techniques used on the raw dataset, the feature selection techniques and the classifiers used are described in Section 4. The experiments performed are listed in Section 4.4. and Section 4.5 discusses the metric used to measure prediction accuracy. A discussion of the results obtained from these experiments is given in section 5. Finally, conclusions are drawn in Section 6.

3. Corpus and Libraries

3.1 Corpus

For the purpose of training and testing, a custom dataset TEMO, was designed. This is a corpus of 245 books written in the English language by over 100 different authors. A database of these books was created tabulating the name of the book, author name, author gender, author race, genre of the book, year of publishing of book and year of birth of author. The age attribute of the author was taken to be the age at the time of publishing the book. This is simply the difference between the year of publishing of the book and the author's year of birth. These are author attributes that are used for author profiling in the training phase. The genre of each book was categorized into 18 categories like fiction, romance, philosophy and classics to name a few.

The PAN corpus consisted of blogs and multiple conversations of over 300,000 people.

3.2 Libraries used

The open source machine learning library, scikit-learn [21], was used for text analytics. Another open source library, Natural Language Toolkit library [22], was used for the preprocessing of the data.

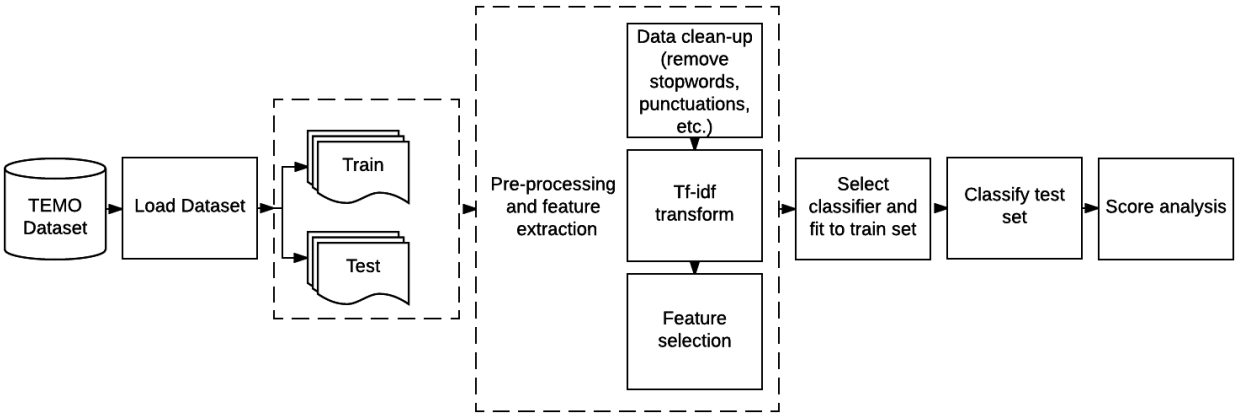

Fig. 1. **Data Processing Pipeline** This figure represents a block diagram of the generalized flow of the data through six processing steps. The dataset is loaded into the train and test folders, after which it goes through preprocessing stage which comprises of Stop words removal ,lemmatizing, punctuation removal, stemming. After feature extraction from the preprocessed data, classification models are trained and evaluated on the test set.

4. Methods

Fig. 1. depicts the data processing pipeline that is adopted for the implementation of experiments performed in this paper

4.1 Data Pre-processing and feature extraction

The data corresponding to the train set is first loaded from the dataset. Feature vectors are retrieved from each document based on a numerical statistic called term frequency-inverse document frequency (tf-idf).
Term-frequency is obtained by using a bag-of-words model. In the bag-of-words model, text is represented as a multiset of its composite words, without taking grammar and word order into consideration but retaining multiplicity.The idf is defined as:

$$idf(t) = log(|D| / 1 + |\{d : t \in d\}|)$$

$$tfidf(t) = tf(t,d) * idf(t)$$

Where $D = \{d_1, d_2, \ldots\ldots, d_n\}$ defines the document space where n is the number of documents in corpus. In *(1)*, $|\{d : t \in d\}|$ defines the number of documents where t appears. We add *1* in the denominator so as to avoid zero-division.

Stopwords like “a”, “the”, “in”, etc. have a high term frequency in any given text (sklearn has about 318 words in the stop set). These stopwords are removed from the data (both train and test) to bring focus to the more important words/features. Thus, this step is crucial to enable the feature extraction engine to retrieve important words of interest. Further, punctuations and numbers (both in numeric and word forms) are removed as a data clean-up process. Other pre-processing techniques using the Natural Language Toolkit were also performed. The three processing techniques used are - word and punctuation tokenizing, stemming and lemmatizing. Tokenizing reduces a given text into individual words or punctuations. Individual words are called tokens and these tokens undergo further processing. The stem of a token is the largest part of the token that does not contain prefixes or suffixes. Lemmatization of a token results in the base form of the token (word). For example, the token “communities” becomes “community” after lemmatization and “commun” after stemming. Each word is then attributed to a category of the list of parts of speech categories. This is achieved using a parts of speech tagger or a POS-tagger (from NLTK). Thus both NLTK and sklearn libraries are used to pare the database while adding additional meaning to it.

4.2 Feature selection

A chi-square analysis is performed for feature selection for each of the features. Chi-square ranks features with respect to their usefulness. If there exists a relationship between the feature and the respective target variable, the feature variable and the target variable are dependent thus making the feature variable important. If the feature variable and the target variable are independent (chi-square value is high), the feature is discarded.

4.3 Classification techniques

Six different classifiers are used for the purpose of performance comparison. A classifier (Multinomial Naive Bayes, Random Forest, Perceptron, Passive Aggressive, Ridge Classifier, kNN) is fit to the features extracted from the training data that has been pre-processed by the steps mentioned above. For predicting the age of the author two approaches were used. First was to generate a continuous regression model and another was to classify into discrete age groups.

Multinomial Naive Bayes classifier is a simple probabilistic classifier with the assumption that there is no correlation between features. It is a high bias/low variance model (underfitting) and is not computationally intensive. Thus it is a popular choice in the case of small training sets. It works on basic formula of Bayes Theorem:

$$P(c/x) = [P(x/c) * P(c)] / P(x)$$

Where *P(c|x)* is posterior probability of the class given predictor, *P(c) is* prior of class, *P(x|c)* is likelihood and P(x*)* is prior of predictor [15].

Random Forests is an ensemble classifier that fits multiple decision trees on several sub-samples of dataset simultaneously using averaging to augment prediction and control overfitting and gives an output corresponding to the most frequent class predicted by the individual trees [18].

Perceptron classification performs a binary classification of the given data by combining a set of weights with the feature vector and runs efficiently on large datasets [16].

Ridge classifier is a regression method which is converted into a classifier by adding a threshold to it. For linearly separable data, it works similar to linear classifier by defining a hyperplane. But for non-linear case the 'ridge' parameter comes into play which makes it work for them [20].

kNN classification is a type of instance based non-parametric, non-generalizing learning technique. It finds the k-most covariant instances around the test data and predicts the label from it [19].

Passive Aggressive classifier is an online algorithm which learns from massive streams of data. Intuitively, it passively stores the model if it correctly predicts the label and aggressively corrects the model if the prediction is wrong [17].

The sklearn library was used to import the above libraries necessary for training and classification. The test data is transformed by a classifier to obtain the predicted class/labels. To overcome the challenge posed by a small dataset, we use cross validation to average the measures of fit to get a better model and make more accurate predictions. In a k-fold cross validation,the data is split into k groups. The prediction function is obtained using k-1 groups, and the kth dataset, called the validation set, is used as a test set in the training phase. Cross validation helps approach problems like overfitting. Using a five fold cross validation approach, we split the dataset into five groups. The prediction function is learned using four groups, and the fifth group is used for testing.

4.4 Test cases and Experiments

A comparative study of the six classification techniques with the following combinations of preprocessing was made:

1. Tokenizing using sklearn (with and without stopwords)
2. Stemming as tokenizer (with and without stopwords)
3. Lemmatization as tokenizer (with and without stopwords)
4. Cross-validation

4.5 Evaluation and Performance Score

Following classification, a performance score is computed to measure the correctness of the predicted labels. A one-to-one comparison is made between the true and predicted classes and the "accuracy" is simply the number of correct classifications. The score is normalized in order to acquire a score in the range of [0,1] for easy interpretation of the results. The mathematical form of the scoring metric is given by :

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}-1} 1(\hat{y}_i = y_i)$$

where, $n_{samples}$ is the total number of test samples, y_i is the true class of the ith sample and $\hat{y}_i$ is the predicted class of the ith sample.

5. Results and Discussions

The results of the methodology described in Section 4 applied on the PAN corpus and custom dataset of books are discussed below.

5.1 Performance on PAN corpus

The following results were obtained on PAN dataset for author profiling. Fig 2 shows the performance of four

classifiers in predicting gender from the conversations. Random Forest performed the best with a score of 0.745 while kNN failed to yield results due to memory constraints. Age prediction gave similar results with random forest out performing other classifiers with a score of 0.761 (Fig 2).

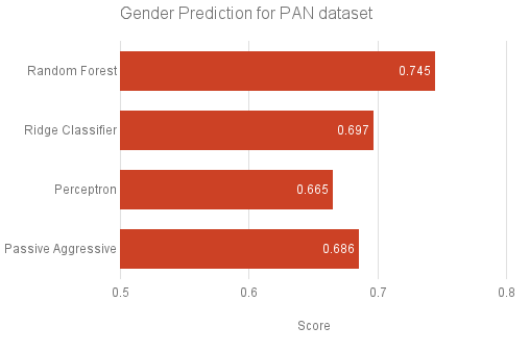

Fig. 2. Gender Prediction on PAN corpus. The graph above represents the performance score across 4 classifiers on the PAN corpus for gender prediction. The Random forest classifier performed the best giving an accuracy of 74.5%.

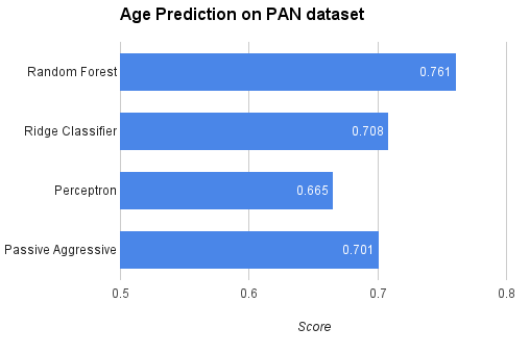

Fig. 3. Age prediction on PAN corpus. The chart above depicts the performance score of four classification techniques on the PAN corpus for Age prediction. The Random forest classifier gave the best classification accuracy of 76.10%.

5.2 Performance on corpus of books

The following results were obtained on the published books dataset described in section 2.1 . The performance on classification based on gender was similar to those observed on the PAN dataset, but the classification based on age groups gave poor results.

Gender prediction performance , depicted in Fig4, was the highest for the Passive Aggressive classifier with cross-validation with a score of 0.88. The scores obtained with the use of cross validation were, in majority of the cases, notably better than those without the use of cross validation, with the multinomial Naive Bayes Classifier acquiring an increase of 22 % in its score.

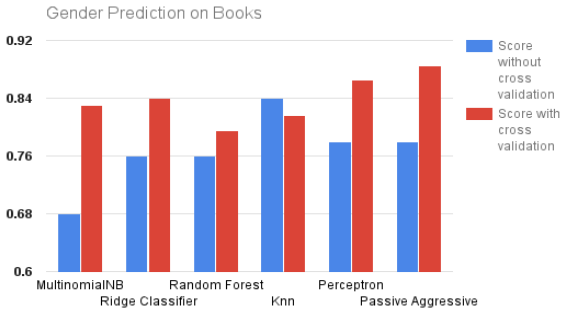

Fig. 4. Performance on gender prediction for classifiers with and without cross-validation The above chart depicts the performance of each classifier with and without cross-validation for predicting gender of authors of novels and published books. A five-fold cross-validation was chosen.

For the ease of classification of age, the range of ages of all authors was split into sub-ranges of 10-years.The classifiers performed very poorly in this task,with the maximum score achieved being only 0.45. We believe that this was due to the inherent nature of how age affects writing.

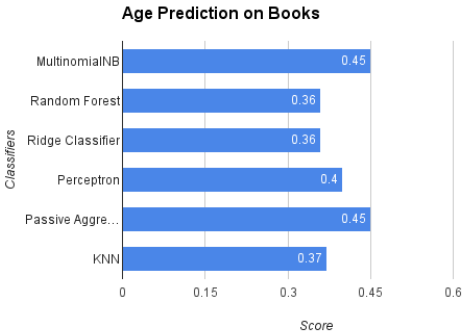

Fig. 5. Performance for Age Prediction on Books The graph above depicts the performance score for 6 classifiers for the label "age". The maximum score achieved was only 0.45

Data preprocessing techniques such as the removal of stop words, stemming and lemmatization result in negligible changes in the score. Since the corpus was small, we hypothesised that inclusion of stop words during feature extraction would depreciate the performance of the classifier. In contradiction to this, the performance measures we obtained were comparable and in most cases identical. It could be possible that the corpus is not large but big enough to be minimally affected by the presence of the stop words. The stop words could prove to

affect the performance score negatively if the document has no meaningful "content" or is "content-less"[14] (which could be the case in a small document).

6. Conclusions

In this paper we explored the profiling of an author based on extracted feature vectors like tf-idf, n-gram and POS tagging. The characteristic traits used for profiling were gender, race, genre of the book, age at the time of publishing, economic status, circumstantial influences and substance abuse history.

A higher performance was achieved for gender and age classification on PAN dataset than the participants (with max 0.65), which could be due to inclusion of tf idf and n-gram analysis of POS tags

For the purpose of further exploration, we constructed a custom dataset of books of 138 authors. The manually labelled data was used in the data processing pipeline (Fig. 1.) for different test cases.

A comparative study was done across six classifiers as mentioned in Section 4.5. Passive Aggressive and kNN classifiers performed markedly better on our dataset as compared to Naive Bayes, Random Forest, Passive Aggressive, and Ridge Regression classifiers. The results of classification with cross-validation followed the same trend as without cross-validation. The performance score without stop words did not deviate widely from the performance score with stop words across different preprocessing techniques. Performance score was improved significantly upon using five-fold cross-validation as opposed to the standard classification technique.

The performance on classification based on age on the custom dataset of books was poor, contradictory to what was observed in the case of the PAN dataset. This was due to the fact that the dataset composed of books had authors of the same age groups belonging to different eras, thus making grouping a difficult task due to the differences in writing across different eras. Whereas the PAN dataset is composed of blog entries, with the focus on conversations between people, where era doesn't come into the picture.

One of the main challenges we faced was during the data retrieval phase. There were limited sources that offered free data thus making data acquisition difficult. Out of the corpus of books, most of the books were authored by people of Caucasian origin, thus biasing the dataset.This bias made it difficult to use the "race" as a reliable label to attribute to the author's profile.

Future work is to conduct a thorough analysis of individual misclassifications in our results to provide more insight into edge cases. In addition, it would be interesting to extend this approach to the task of author identification by matching profile labels. Another area to explore is to generate a dynamic stop word set which could help in the enhancement of the performance score.

7. Acknowledgements

We would like to acknowledge Professor Dimitry Korkin for his valuable inputs provided during the course of the project.

8. References

1. Luyckx, K., & Daelemans, W. (2011). The effect of author set size and data size in authorship attribution. Literary and linguistic Computing, 26(1), 35-55.
2. Lin, J. (2007). Automatic author profiling of online chat logs.
   Chicago
3. Koppel, M., Schler, J., & Argamon, S. (2011). Authorship attribution in the wild. Language Resources and Evaluation, 45(1), 83-94.
4. Weren, E. R., Kauer, A. U., Mizusaki, L., Moreira, V. P., de Oliveira, J. P. M., & Wives, L. K. (2014). Examining multiple features for author profiling. Journal of Information and Data Management, 5(3), 266.
5. Sanderson, C., & Guenter, S. (2006, July). Short text authorship attribution via sequence kernels, Markov chains and author unmasking: An investigation. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (pp. 482-491). Association for Computational Linguistics.
6. Kešelj, V., Peng, F., Cercone, N., & Thomas, C. (2003, August). N-gram-based author profiles for authorship attribution. In Proceedings of the conference pacific association for computational linguistics, PACLING (Vol. 3, pp. 255-264).
7. Estival, D., Gaustad, T., Pham, S. B., Radford, W., & Hutchinson, B. (2007). Author profiling for English emails. In Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING'07) (pp. 263-272).
8. Ledger, G., & Merriam, T. (1994). Shakespeare, fletcher, and the two noble kinsmen. Literary and linguistic computing, 9(3), 235-248.
9. Mosteller, Frederick, and David Wallace. "Inference and disputed authorship: The Federalist." (1964).
10. Argamon, S., Koppel, M., Fine, J., & Shimoni, A. R. (2003). Gender, genre, and writing style in formal written texts. TEXT-THE HAGUE THEN AMSTERDAM THEN BERLIN-, 23(3), 321-346.
11. Rangel, F., Rosso, P., Moshe Koppel, M., Stamatatos, E., & Inches, G. (2013). Overview of the author profiling task at pan 2013. In CLEF Conference on Multilingual and Multimodal Information Access Evaluation (pp. 352-365). CELCT.

12. Peersman, C., Daelemans, W., & Van Vaerenbergh, L. (2011, October). Predicting age and gender in online social networks. In Proceedings of the 3rd international workshop on Search and mining user-generated contents (pp. 37-44). ACM.
13. Schler, J., Koppel, M., Argamon, S., & Pennebaker, J. W. (2006, March). Effects of Age and Gender on +Blogging. In AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs (Vol. 6, pp. 199-205).
14. Zhong, N., Liu, J., Yao, Y., Wu, J., Lu, S., & Li, K. (Eds.). (2007). Web Intelligence Meets Brain Informatics: First WICI International Workshop, WImBI 2006, Beijing, China, December 15-16, 2006, Revised Selected and Invited Papers (Vol. 4845). Springer.
15. Zhang, H. (2004). The optimality of naive Bayes. AA, 1(2), 3.
16. Gallant, S. I. (1990). Perceptron-based learning algorithms. IEEE Transactions on neural networks, 1(2), 179-191.
17. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). Online passive-aggressive algorithms. Journal of Machine Learning Research, 7(Mar), 551-585.
18. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
19. Dasarathy, B. V. (1991). Nearest neighbor ({NN}) norms:{NN} pattern classification techniques.
20. Hoerl, A. E. (1962). Application of ridge analysis to regression problems. Chemical Engineering Progress, 58(3), 54-59
21. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.
22. Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with Python. " O'Reilly Media, Inc.".