Authorship Verification in Classical Chinese Texts

Ellis Cain
Indiana University
escain@iu.edu

Abstract

Traditional methods consist of "close reading for stylistic detail" (Juola, 2012), which have been combined with statistical methods and natural language processing to increase the productivity. There are already basic non-NLP and non-computationally-intensive authorship verification models for Dutch, English, Greek, Spanish, German text documents (Halvani et.al., 2016) along with a copy detection models for modern Chinese (Bao et.al., 2003), however there does not appear to be any models for Classical Chinese. The designed program uses n-grams to represent the text documents, which are then transformed using a Tf-Idf vectorizer. K-Means clustering is then used to cluster the documents into author profiles to do authorship verification.

1 Introduction

When reading a research paper or book, the author(s) is (are) clearly displayed on the paper or book, and if you cannot find who is the author, you could search the title to find more information about the authorship. However, this is not always the case for ancient texts where they either did not record the author or there are multiple versions with different recorded authors. One well-known example of disputed authorship in Classical Chinese texts would be Zhuangzi (莊子), an ancient Daoist text, which will be used in this analysis. For the inner chapters (内篇), authorship is generally attributed to Zhuangzi, but since the Song dynasty (960-1279), there have been many doubts that the outer chapters (外篇) were actually written by Zhuangzi (Knechtges, 2014 and Roth, 1993).

The author verification problem includes taking a group of potential authors and a set of text samples whose authors are known which are used to train the program, and a target text with an unknown author which is analyzed. Based on *A survey of modern authorship attribution methods* by Efstathios Stamatatos, the two approaches to author verification are profile-based approaches and instance-based approaches (along with a hybrid between the two). Profile-based approaches consist of developing a representation of an author's profile, and then comparing the unknown text to this profile. Instance-based approaches differs in that each training text is separately represented as an individual instance of an author's style (Stamatatos, 2009).

The methods for author clustering, mainly character n-grams, set forth in Sari and Stevenson's *Exploring Word Embeddings and Character N-Grams for Author Clustering* (2016) will be used. Texts from Confucius (Kongzi), Mencius (Mengzi), Dong Zhongshu, Liu Xiang, and Zhuangzi's inner chapters will serve as the the different author profiles. Three of Zhuangzi's outer chapters will serve as the test documents of unknown authorship. A more detailed list of source texts can be found in the appendix

2 Methods

The texts were obtained from the Chinese Text Project (Sturgeon, 2019), around eight documents for each author, with three exerpts from Zhuangzi's outer books to serve as documents with an unknown author. The length of the documents vary from 674 characters to 8330 characters. The Classical Chinese Universal Dependencies Treebank (annotated and converted by the Institute for Research in Humanities, Kyoto University) was used to train a Classical Chinese UDpipe model. This UDpipe model was then used to segment and tokenize each of the text documents and convert them into conllu files.

Character *n*-grams were extracted from the text documents, with *n* ranging from 2 to 10. The Tf-Idf vectors were calculated using the CountVectorizer and TfidfTransformer from Scikit-Learn python library. The vectors were normalized as Pedregosa et al. (2011) stated that normalizing the Tf-Idf vector makes the KMeans python function behave as a spherical K-Means for better results. Next, the K-Means clustering implementation from the SciKit-Learn python library was used to cluster the documents.

Once the data was analyzed by the K-Means clustering algorithm, multiple metrics were used to evaluate the quality of the clustering, including: homogeneity, completeness, V-measure, adjusted rand-index, silhouette coefficient, and weighted F1 score. The final program and materials can be found on GitHub https://github.com/ellissc/LING-L545/tree/master/final_project.

3 Results

The program was able to extract n-grams from the digitized texts and run the K-Means clustering algorithm on the vector obtained from the TfIdfVectorizer. Homogeneity indicates that a given cluster contains only data points that are members of a given class. Completeness is the inverse; it indicates that all the data points of a given class are members of the same cluster. V-measure is the harmonic mean of homogeneity and completeness. Adjusted Rand Index is a similarity measure for two clusters by comparing differences in predicted and true clusters. The Silhouette Coefficient indicates how well the clusters are defined. Overall, metric values close to 1.0 indicate good performance.

Instance	F1 Score
Average	0.176
Maximum	0.424
Minimum	0.000
St dev	0.092

Table 1: F1 Scores with tf-idf normalization

Instance	F1 Score
Average	0.225
Maximum	0.976
Minimum	0.000
St dev	0.194

Table 2: F1 Scores without tf-idf normalization

The results for running the K-Means algorithm with normalization of the tf-idf vector follow. The K-Means clustering algorithm was able to run 75 instances in 330.733s. It averaged an F1 score of 0.176, with the minimum value of 0.000 and a maximum value of 0.424, which can be found in Table 1. Standard deviation was 0.092. The homogeneity, completeness, V-measure, and adjusted rand-index were all between 0.18 and 0.54, while the silhouette coefficient was between 0.52 and 0.69, which can be found in table 3.

The results for running the K-Means algorithm without normalization of the tf-idf vector follow. The K-Means clustering algorithm was

able to run 75 instances in 1910.730s. It averaged an F1 score of 0.225, with the minimum value of 0.000 and a maximum value of 0.976, which can be found in Table 2. The standard deviation was 0.194. The homogeneity, completeness, V-measure, and adjusted rand-index were all above 0.5, while the silhouette coefficient was around 0.003, which can be found in table 4.

4 Discussion

Generally, normalizing the tf-idf vector led to quicker processing and metric values around 0.5, but the F1 score was often below 0.5. When normalizing was not used, the K-Means produced much higher metric values and a higher range of F1 scores, but the silhouette coefficient of 0.003 suggested that some of the clusters are overlapping.

Pedregosa et al. (2011) noted that normalization is common for text classification or clustering tasks and other models used by the information retrieval community. When the tf-idf vector was not normalized, however, the K-Means clustering algorithm performed better in all of the metrics except for the Silhouette Coefficient. While some of the clusters may be overlapping, the increased F1 score may be worth the trade-off.

While it is understandable that the K-Means clustering performance heavily relies on the initialization of the starting points for the clusters, the performance was not very consistent. This could be due to a couple factors, such as the n-gram range or document length. However, varying the range of n-grams also had minimal affect on the performance. There was a large disparity in the document lengths, more precisely, ranging from 674 characters to 8330 characters. Some author's works were shorter on average than others, which could have an affect on the clustering performance. Document type could potentially influence the performance, as all of the authors besides Zhuangzi were famous Confucian thinkers, while Zhuangzi was a Daoist thinker. This may not drastically change the results, but the different writing styles may be distinct enough to slightly impact the clustering.

5 Conclusion

This relatively straightforward method of using n-grams and K-Means clustering has potential, with the right setup, to be an effective method for author verification for Classical Chinese. As mentioned previously, at the cost of processing time, using tf-idf vectors that are not normalized for the K-Means clustering algorithm can garner high F1 scores. Looking at Zhuangzi's inner and outer chapters, however, the program did not reach a clear consensus. For further study, it would be interesting to look at a word2vec model for Classical Chinese and see if there would be any improvements to the K-Means clustering performance. In addition to this, it would also be interesting to analyze the potential authors for Zhuangzi's outer chapters, instead of other famous authors.

References

[1] Bao, J. P., Shen, J. Y., Liu, X. D., Song, Q. B. (2003). A survey on natural language text copy detection. Journal of software, 14(10), 1753-1760.

[2] Halvani, O., Winter, C., Pflug, A. (2016). Authorship verification for different languages, genres and topics. Digital Investigation, 16, S33-S43.

[3] Juola, P. (2012). An Overview of the Traditional Authorship Attribution Subtask, Notebook for PAN at CLEF 2012.

[4] Knechtges, David R. (2014). Zhuangzi 莊子. Ancient and Early Medieval Chinese Literature: A Reference Guide, Part Four. 2314-23.

[5] Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python, JMLR 12, 2825-2830.

[6] Roth, H.D. (1993). Chuang Tzu 莊子. Early Chinese Texts: A Bibliographical Guide. Berkely., 55-66.

Instance	Homogeneity	Completeness	V-measure	Adjusted Rand-index	Silhouette Coefficient
Average	0.353	0.369	0.360	0.179	0.598
Maximum	0.533	0.550	0.541	0.361	0.681
Minimum	0.187	0.208	0.197	0.043	0.523
St dev	0.074	0.076	0.074	0.074	0.036

Table 3: Metrics used to evaluate the quality of the clustering, with tf-idf normalization

Instance	Homogeneity	Completeness	V-measure	Adjusted Rand-index	Silhouette Coefficient
Average	0.853	0.866	0.859	0.778	0.003
Maximum	1.000	1.000	1.000	1.000	0.004
Minimum	0.688	0.710	0.689	0.512	0.002
St dev	0.071	0.062	0.066	0.112	0.000

Table 4: Metrics used to evaluate the quality of the clustering, without tf-idf normalization

[7] Sari, Y., Stevenson, M. (2016). Exploring Word Embeddings and Character N-Grams for Author Clustering. In CLEF (Working Notes), 984-991.

[8] Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, Xiaoyong Du. (2018). Analogical Reasoning on Chinese Morphological and Semantic Relations, ACL 2018.

[9] Stamatatos, E. (2009). A survey of modern authorship attribution methods. Journal of the American Society for information Science and Technology, 60(3), 538-556.

[10] Sturgeon, D. (2019). Chinese Text Project: a dynamic digital library of premodern Chinese, Digital Scholarship in the Humanities.

[11] Yasuoka, Koichi. (2019) Universal Dependencies Treebank of the Four Books in Classical Chinese, DADH 2019: 10th International Conference of Digital Archives and Digital Humanities.

A Source Texts

Kong zi, selected from The Analects (論語): Xue Er (學而), Wei Zheng (🄵政) Ba Yi (八佾), Li Ren (里仁), Gong Ye Chang (公冶長), Yong ye (雍也), Shu Er (述而), Tai Bo (泰伯).

Liu xiang, selected from Shuo Yuan (🄵苑): Jun Dao (君道), Chen Shu (臣術), Jian Ben (建本), Li Jie (立節), Gui De (貴德), Fu En (復恩), Zheng Li (政理), Zun Xian (尊賢).

Dong Zhongshu, selected from Chun Qiu Fan Lu (春秋繁露): Chu Zhuang Wang (楚莊王), Yu Bei (玉杯), Zhu Lin (竹林), Yu Ying (玉英), Jing Hua (精華), Wang Dao (王道), Mie Guo (滅國), Suiben Xiaoxi (隨本消息).

Meng zi, selected from Meng Zi (孟子): Liang Hui Wang Shang and Xia (梁惠王上, 梁惠王下), Gong Sun Chou Shang (公孫丑上), Teng Wen Gong Shang (滕文公上), Li Lou Shang (離婁上), Wan Zhang Shang (萬章上), Gaozi Shang (告子上), Jin Xin Shang (盡心上).

Zhuang zi, selected from Zhuangzi inner chapters (莊子內篇): Enjoyment in Untroubled Ease (逍🄵🄵), The Adjustment of Controversies (齊物論), Nourishing the Lord of Life (養生主), Man in the World, Associated with other Men (人間世), The Seal of Virtue Complete (德充符), The Great and Most Honoured Master (大宗師), The Normal Course for Rulers and Kings (應帝王)

Zhuang zi, selected from Zhuangzi outer chapters (外篇): Webbed Toes (🄵拇), Horses' Hoofs (馬蹄), Cutting open Satchels (🄵🄵)

B Data Policy

For the author profile generation, the training and testing texts will be gathered from the Chinese Text Project (https://ctext.org/), an

open-access library of digitized Classical Chinese texts. The license can be found at https://ctext.org/faq#copyright. To train the UD pipe segmenter and tokenizer, the Classical Chinese Universal Dependencies Treebank (annotated and converted by the Institute for Research in Humanities, Kyoto University), which can be found at https://github.com/UniversalDependencies/UD_Classical_Chinese-Kyoto. It is under the creative commons license. For the word embeddings, I plan to follow the instructions listed at https://github.com/Embedding/Chinese-Word-Vectors, which use creative commons data for training the models. Other tools from the Scikit-learn library will also be used throughout the data analysis, which can be found at http://scikit-learn.org/.