Authorship Attribution: ”To attribute or not to attribute?”

Final Report by Lisa Becker, 775242

A project by Nina Harlacher, Joceline Ziegler and Lisa Becker in ”Advanced Natural Language Processing” by Prof. Dr. Tatjana Scheffler at University Potsdam.
Submitted March 15th, 2019.

1 Abstract

This project concerns authorship attribution, comparing the performance of three models with each other: a generative model with a naïve Bayes classifier, a generative model with a support vector machine and n-gram-tracing. The best result achieved the generative model with naïve Bayes (85.7%) and was used to attribute one of five possible authors to each of three different plays of the Elizabethan epoch.

2 Approach and Main Results

The project[1] addressed the question of authorship for plays of the Elizabethan epoch since some of their authorship is still debated. We decided to analyse the following plays of which the authorship is still contested: *The Second Maiden’s Tragedy (1611)*, *The Puritan (1607)* and *A Yorkshire Tragedy (1608)*. We tested the three plays of question on five authors: William Shakespeare, Thomas Middleton, George Chapman, Ben Jonson and Christopher Marlowe. To address our research question we employed the following two steps: firstly, we started with implementing three different models and to evaluate their performance we measured the accuracy of these models. According to our evaluation the generative model with naïve Bayes scored best. Secondly, based on our results we proceeded with improving this model by combining it with additional linguistic features.

We reconstructed a bag of words and a generative model by Fox and colleagues (2012) as a baseline to improve over with various features, compared its accuracy with naïve Bayes with that of a support vector machine and with a new model called n-gram-tracing (Fox et al., 2012; Grieve et al., 2018). The generative model with naïve Bayes proved to be the best one with a baseline accuracy of 83.1% which was still lower than Fox and colleagues’ baseline accuracy of 84.4% (see Table 1). The support vector machine and n-gram-tracing did not prove useful for our project. Thus, we decided to implement features to improve over the generative model with a naïve Bayes classifier. The features which showed best results with the generative model were lemmatizing, stemming and keyword extraction, pushing the naïve Bayes accuracy to 84.4% (with lexical features) and 85.7% (with structural features). Only with those features we managed to achieve an accuracy which was as high as the baseline from Fox and colleagues. Other features worsened the accuracy. Our best model attributed *The Second Maiden’s Tragedy* and *A Yorkshire Tragedy* to Thomas Middleton and *The Puritan* to George Chapman.

The project showed that developing methods for authorship attribution is necessary since it seems there is still room for improvement. Furthermore, the authorship of many documents is still to be clarified, not only from the Elizabethan epoch, but in various areas of literature and domains such as forensics.

3 Data and Tools

We used the same data as Fox and colleagues as they made it available online[2]. Since the plays were already pre-processed, we did not apply any further changes to them in order to reproduce Fox and colleagues’ findings as close as possible. The pre-processing included the removal of stage directions, character labels, prologues, act and scene breaks, footnotes and others. With Fox and colleagues’ corpus, we had at least seven

[1] Code and data of the project (last accessed March 15th, 2019).

[2] Corpus used for ”Statistical Stylometrics and the Marlowe-Shakespeare Authorship Debate” by Fox et al. 2012 (last accessed January 9th, 2019).

plays per author (Early Shakespeare: 23, Late Shakespeare: 13, Middleton: 9, Marlowe: 7, Jonson: 14, Chapman: 11). The separation of early and late Shakespeare is reasoned by the amount of plays that we have from Shakespeare and big stylistic differences between the so-called earlier (until 1601) and later (from 1602) works of Shakespeare. Thus, splitting the data not only balances out the data, but also reflects this stylistic shift.

For our generative model, we needed a list of stop words, as Fox and colleagues manually created a list with stop words based on the available data. Doing this would have been outside the scope of this project, therefore, we created our own list from various stop word sources[3] since Fox and colleagues did not provide theirs online. We used packages from NLTK (Natural Language Toolkit) for various types of language processing and the Pandas data frame to store the data.
While Fox and colleagues used the Stanford POS (Part of Speech) tagger, we used the spaCy POS tagger due to technical reasons, although the former proved to work well for historic texts by a drop of only 10% in accuracy and might thus produce better results (Archer et al., 2007).

4 Models

We implemented three different models to find the most suitable approach for authorship attribution. Firstly, a bag of words model was implemented as a baseline. A generative model was implemented and we experimented with two different classifiers: one evaluated with naïve Bayes and one with support vector machines. The bag of words as well as the generative model with naïve Bayes with its baseline features are based on Fox and colleagues' paper. Furthermore, a fairly new approach called n-gram-tracing was tested. This model was based on a paper by Grieve and colleagues (2018). We measured the performances of the models by using a leave-one-out cross validation. Our goal was to compare their accuracy and improve over the best model. All implementations are explained and evaluated below.

4.1 Bag of Words: Naïve Bayes

We implemented a bag of words to reconstruct the approach of Fox and colleagues. One corpus was formed per author with all their plays in a profile-based approach instead of passing multiple instances per class as researchers have suggested (Stamatatos, 2009). We evaluated our baseline with a leave-one-out cross validation by separating one play per author per validation round. A sparse data countvector was created per train and test corpus set based on word counts. The naïve Bayes classifier was trained in each round with the training vector and calculated prediction for the test vector per round. The accuracy was calculated by the prediction compared to the canonical author of the play over all validation rounds. Our bag of words model achieved an accuracy of 83.1% which is close to baseline of Fox and colleagues with 84.4%. This difference can be explained by their use of Kullback-Leibler divergence instead of naïve Bayes. We decided for the latter to be able to compare to the generative model and the support vector machine and so we could apply our knowledge and experience concerning naïve Bayes classifiers from this class.

4.2 Generative Model: Naïve Bayes

Fox and colleagues built a generative model evaluated with naïve Bayes which we reconstructed. Its baseline already included three features: the frequency of POS tags of words which are not stop words, the frequency of stop words and the frequency of ngrams of POS tags and stop words combined. For achieving the best possible accuracy with the stop words feature, we tested different sets of stop words. The best result was achieved with the spaCy stop word list (containing 305 stop words), probably due to its combination with our use of the spaCy POS tagger. Our second best stop word list consisted of the biggest overlap from the various stop word sources that we used (710 stop words) which was the same size of stop word list as Fox and colleagues used.

4.3 Generative Model: Support Vector Machine

We chose a support vector machine since they are commonly used in authorship attribution for longer documents (Diederich et al., 2003; Sudheep Elayidom et al., 2013; Zheng et al., 2006). Support vector machines depend on getting passed mul-

[3] Packages: NLTK, spaCy, stop_words
Other: RANKS NL (last accessed March 15th, 2019), Semantic Similarity (last accessed March 15th, 2019) .

tiple instances per class. Since each class represented an author's corpus, we did not have many instances based on individual plays. To compromise this, we tried passing single sentences of an author's corpus as instances, but our model only achieved approximately 54.54% accuracy. Hence, we decided on using the data as done by Fox and colleagues which was one play being one instance. We achieved much better results but they still did not exceed the accuracy of the generative model with naïve Bayes. Adding features to the support vector machine did not improve the accuracy but did tend to worsen the performance of the model (see Table 1). Thus, we decided to improve over the generative model with naïve Bayes instead and compare it to the n-gram-tracing model.

4.4 Feature Engineering

Since the generative model with naïve Bayes classification achieved the highest accuracy, we tried to improve over it. We implemented different syntactical, lexical, structural and content-specific features as suggested for authorship attribution to improve over the baseline (Zheng et al., 2006). The best combination seemed to be the baseline with lexical stemming and lemmatizing (both 84.4%) and the baseline with content-specific keyword extraction (85.7%). My contribution to the feature engineering was the implementation of the hapax legomena ratio and the average word length as features. Neither one has proven to be useful in our implementation although research has shown that the word length has been used as an identifying feature for Elizabethan authors (Mendenhall, 1901; Pinsken, 2008). Other features that we implemented but which worsened the performance were the following: sentence length (lexical) and line length (structural). Reasons for them decreasing the accuracy rather them improving it can be changes done by the editors of the play regarding punctuation or line composition. The more features were combined together, the more the performance of our classifier decreased. Depending on what kind of data is used and the type of authorship detection, different features have to be taken into account. Thus, the feature selection is another very important issue of building a good classifier beside the choice of the right type of model.

Since the baseline features with the keyword extraction proved to be the best approach, we used this combination to classify the three plays in question. *The Second Maiden's Tragedy* and *The Puritan* were classified as being written by Thomas Middleton to whom these plays have been attributed before (Taylor and Lavagnino, 2014). Our attribution for *The Second Maiden's Tragedy's* aligns itself with the one from Fox and colleagues. Our model attributed *A Yorkshire Tragedy* to George Chapman who has not been amongst the debated authors for this play yet. Thus, our model is probably not right in this case, as its accuracy achieved only 85.7%.

4.5 N-Gram-Tracing

My main contribution to this project was the implementation of the n-gram-tracing approach based on the paper by Grieve and colleagues. It calculates an overlap coefficient of n-grams in a given corpus with that of another corpus of a similar size. In order to calculate the overlap coefficient of the plays in question with each author we took random lines from each play of a given author such that they added up to the size of the questioned play. N-gram-tracing works without any changes to the data or the use of features. The model was introduced for shorter corpora, such as a few hundred words. We decided to implement it to see whether it would give useful results for our project with much larger corpora. Grieve and colleagues achieved their best results with 7-grams and 8-grams on a character level (both 98%) and bigrams on a word level (96%). We compared our results to theirs and agreed that those three features achieve the best accuracy compared to other word- or character-n-grams, although the results from our implementation did not even get close to the accuracy of Grieve and colleagues; character 7-grams achieved 38.4%, 8-grams 60.8% and word bigrams 29.4%.

5 Conclusion

We did not manage to reach the same accuracy as of the model by Fox and colleagues. This might be due to the lack of detail given by the paper we based our approach on. Another reason might be that we did not use the same list of stop words as Fox and colleagues and the spaCy POS tagger instead of the Stanford one. Moreover, we showed that n-gram-tracing does not seem to be an appropriate approach to classify longer texts than the ones used by Grieve and colleagues. Using our best model, we produced further evidence that Thomas Middleton can be the author of *The Second Maiden's Tragedy* and *The Puritan*. However, our model did not give a plausible candidate for *A Yorkshire Tragedy*. Thus, our model still can be improved further.

5.1 Personal learning outcomes

Working on this project improved my Python skills since I had no prior background in programming. It was my first time working on a programming project as a group which required soft skills such as task distribution, coordination and developing the project together as a team. The project improved my skills in reading and understanding computerlinguistic papers, since we did not only draw from them theoretically but I implemented a whole model based on Grieve and colleagues, which required an even more thorough understanding of the paper. I strengthened a similar skill by using Python packages by just reading the documentation for the first time. On a more theoretical basis, I got more experience with stylometric features in authorship attribution by not only implementing them but coming up with the features and evaluating them and their influence on the accuracy of our model. Furthermore, I learned how to work with data frames and vectorizers, how to test a model and evaluate its outcome. All those skills will probably be helpful in other courses, my Master's thesis and my future career.

References

Dawn Archer, Paul Rayson, Alistair Baron, Jonathan Culpeper, and Nicholas Ross Smith. Tagging the bard: evaluating the accuracy of a modern pos tagger on early modern english corpora. 2007.

Joachim Diederich, Jörg Kindermann, Edda Leopold, and Gerhard Paass. Authorship attribution with support vector machines. *Applied Intelligence*, 19(1-2): 109–123, 2003.

Neal Fox, Omran Ehmoda, and Eugene Charniak. Statistical stylometrics and the marlowe-shakespeare authorship debate, 2012. URL http://cs.brown.edu/research/pubs/theses/masters/2012/ehmoda.pdf.

Jack Grieve, Isobelle Clarke, Emily Chiang, Hannah Gideon, Annina Heini, Andrea Nini, and Emily Waibel. Attributing the bixby letter using n-gram tracing. *Digital Scholarship in the Humanities*, 16: 26, 2018.

Thomas Corwin Mendenhall. A mechanical solution of a literary problem. *The Popular Science Monthly*, 60:98–105, 1901.

Daryl Pinsken. Marlowe's ghost: The blacklisting of the man who was shakespeare. *iUniverse*, 60:55, 2008.

Efstathios Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3): 538–556, 2009.

M. Sudheep Elayidom, Chinchu Jose, Anitta Puthussery, and Neenu K Sasi. Text classification for authorship attribution analysis. *Advanced Computing: An International Journal*, 4(5):1–10, 2013.

Gary Taylor and John Lavagnino. *Thomas Middleton and Early Modern Textual Culture*. Oxford University Press: Oxford, 2014.

Rong Zheng, Jiexun Li, Hsinchun Chen, and Zan Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American Society for Information Science and Technology*, 57(3): 378–393, 2006.

Features	**Naive Bayes**	**SVM**	**Fox et al.**	**N-Gram Tracing**
Word Frequency (Baseline)	83.1		84.4	38,4 (7-char-ngrams) 60,8 (8-char-ngrams) 29,4 (2-word-ngrams)
Syntactic				
POS tags + Function Words + n-grams of POS and Function Words	81.8 (305) 81.8 (710) 79.2 (747) 79.2 (507)	81.8 (305) 81.8 (710) 80.5 (747) 81.8 (507)	86	
Lexical				
Word Frequency	81.8 (305) 80.5 (710)	81.8 (305) 81.8 (305)		
Stemming	83.1 (305) **84.4** (710)	81.8 (305) 81.8 (507)		
Lemmatizing	**84.4** (305) 84.4 (710) 84.4 (747)	81.8 (305) 80.5 (507)		
Hapax Legomena Avg.	81.8 (305)			
Word Length Avg.	81.8 (305)			
Sentence Length	45.4 (305) 46.7 (710)	44.1 (507)		
Structural				
Line Length	71.4 (305) 70.1 (710)	68.8 (507)		
Content-specific				
Keyword Extraction	**85.7** (305) 84.4 (710)	81.8 (305) 80.5 (507)		

Table 1: Best achieved accuracies in percent. Number of function words used indicated in brackets.