# Authorship Attribution for Ancient Greek Text Fragments Using Support Vector Machines

**Anonymous Author(s)**
Affiliation
Address
`email`

## Abstract

Determining the author of a disputed Ancient Greek text fragment is a very labour intensive and hard problem, currently carried out by classicists only. We have developed two machine learning models to assist with this classification problem. To this end we used the data from the Ancient Greek and Latin Dependency Treebanks. These offer a large number of tagged Ancient Greek texts to which no machine learning methods have been applied to date. As a first step, support vector machines (SVMs) are trained on two models. The first model is based on vocabulary, the second one on more complex syntactical features. We trained binary and multiclass SVMs on text fragments consisting of four sentences and found that the models based on vocabulary outperform the syntax-based models in every situation. The binary model produced precisions ranging from 91.9 to 99.7 percent, whereas the precisions for the multiclass model are in the 83.1 to 96.5 percent range. The results for the latter indicate that more precision might be gained by using different methods to handle the data imbalance.

## 1 Introduction

Determining the author of text fragments is a longstanding problem in classical scholarship. The *Homeric Hymns* and the ending of Sophocles' *Oedipus Tyrannus* are famous examples of disputed texts [1, 2]. Often, however, the authenticity of only small parts of a text is questioned. Because the manuscripts were copied manually for over two thousand years, it is unlikely that any of the extant texts is exactly the same as the original copy. For example, texts can be altered when a note in a margin is accidentally inserted by a scribe, or when a different author takes it upon himself to edit the text, perhaps for a new performance of an old play. Usually, disputed passages are only a couple of lines in length and examples of this can be found in commentaries on almost any ancient text.

Judgments on the authenticity of a fragment have so far been based on the expertise of experienced classical scholars. This is the first study that aims to use machine learning models for authorship attribution for classical texts and also the first study to apply support vector machines to texts in the Greek language. As a first step, two models are created to choose an author from a predefined set of possible authors. The first model is based on vocabulary, while the second model uses syntactical features. These models correspond to the two lines of argument that are usually taken when determining the author of a fragment.

## 2 Support Vector Machines for authorship attribution

The problem of assigning an author from a predefined set is part of a broader class of problems called text categorization. Many methods have been proposed, including Naive Bayes classifiers, decision tree classifiers, DNF rule learners, regression methods, neural networks, memory-based reasoning methods and support vector machines (SVMs) [3]. Joachims argues that SVMs are particularly well suited for text categorization problems. One reason is that SVMs can handle high dimensional inputs, hence eliminating the need for feature selection. Furthermore, most text categorization problems are linearly separable, thereby satisfying one of the SVM model assumptions. When compared to the most popular machine learning methods on several benchmark corpora, SVMs were found to perform best [4]. Since then, SVMs have been applied to text categorization problems with good results, see for example [5]-[7].

First consider a model with two classes. SVMs with a linear kernel construct the separating hyperplane that results in the widest possible margin between the two classes. This is illustrated in figure 1. If the data are not separable, a cost parameter $C$ can be introduced. A larger $C$ corresponds to assigning a higher penalty to errors [8]. By using nonlinear kernels the SVM can be extended to nonlinear models by mapping the input space into a high-dimensional feature space chosen a priori. However, a study by Diederich found that for authorship attribution, the choice of the kernel function has little or no effect on the performance [6]. The linear kernel was found to perform best in most cases. An additional advantage of the linear kernel over other common kernels is that there is only one parameter, $C$, to be chosen. Therefore, in this study attention is restricted to SVMs with linear kernels.

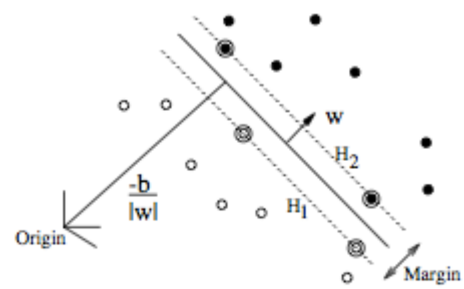



Figure 1: Linear separating hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$ for the separable case. The support vectors are circled. Figure reproduced from [8].

In case of more than two classes, the multiclass problem is usually reduced to multiple binary classification problems. Two common methods are the one-against-all and the one-against-one scheme. In the one-against-all method, $k$ SVMs are trained, where $k$ is the number of classes. The $i$th SVM is trained with all the examples in the $i$th class with positive labels, and all the other examples with negative labels. If SVM $j$ assigns the highest output to new data, this data is classified as $j$. For the one-against-one method, $k(k-1)/2$ classifiers are trained, each on data from two classes. The following voting strategy is used: a new example is given to all SVMs and each assigns a class. The class that is assigned the most is the prediction for the class [9].

## 3 Method

### 3.1 Data collection and fragmentation

Data on a selection of Ancient Greek texts was obtained in XML format from The Ancient Greek and Latin Dependency Treebanks [10]. These treebanks contain linguistic annotations for every word in a corpus of texts. For each word, the exact form, the stem, morphological information and its relation to other words in the same sentence are recorded. An example can be found in figure 2.

```
<word id="1" cid="32749174" form="qeou\s"
lemma="qeo/s1" postag="n-p---ma-"
head="3" relation="OBJ" />
```

Figure 2: Example annotation from Aeschylus' *Agamemnon*. It is word 1 of sentence 32749174, the word is ϑεοὺς with stem ϑεός, it is a noun (n), plural (p), masculine (m) in the accusative case (a), and it is an object (OBJ) depending on word 3.

Publications on this dataset have so far only been concerned with the construction of the treebanks, the only exception being a paper on using part of the Latin data for detecting textual allusions [11]. In our study, the data for all available Ancient Greek texts were used. The exact authors and works can be found in table 1. Each text was divided into fragments of four consecutive sentences. This fragment length was selected because it is representative of the length of many disputed passages.

Table 1: Texts considered in this study.

| AUTHOR | WORKS | WORD COUNT | FRAGMENTS |
|---|---|---|---|
| Aeschylus | *Agamemnon, Eumenides, Libation Bearers, Persians, Prometheus Bound, Seven Against Thebes, Suppliants* | 48.172 | 1008 |
| Hesiod | *Shield of Heracles, Theogony, Works and Days* | 18.881 | 296 |
| Homer | *Iliad, Odyssey* | 232.569 | 3794 |
| Sophocles | *Ajax* | 9.474 | 197 |

### 3.2 Features and transformation of data

Arguments used for authorship attribution are often based on either vocabulary or syntax complexity. Models for both lines of argument were constructed. For the first model, only the lemmas occurring in each fragment were extracted as features. This type of model is known as a 'bag-of-words' model, as it does not take the order of the words or any other syntactical information into account [3]. In the remainder of this paper we refer to this model as the 'lemmas model'. The vectors containing the lemmas were then converted to a document-term matrix. To reduce computational effort, features with over 99.9% sparsity were discarded. This corresponds to removing those words that occur five times or less in the entire corpus. The remaining matrix was of size 5295 by 3821, with the rows and columns representing the fragments and the lemmas respectively. Diederich compared several methods of transforming and normalizing the frequency vectors. He found that logarithmic relative frequencies with $L_1$ normalization have the overall best performance [6]. Let $w_k$ denote lemma $k$, $d_i$ fragment $i$, $f(w_k, d_i)$ the frequency of lemma $w_k$ in fragment $d_i$ and $f(d_i)$ the number of words in fragment $d_i$. The logarithmic relative frequency is then:

$$F_{log}(w_k, d_i) = \log\left(1 + \frac{f(w_k, d_i)}{f(d_i)}\right). \tag{1}$$

Each row of the document-term matrix was transformed according to (1) and then $L_1$-normalized.

For the second model, which will be called the 'syntax model', the information contained in `postag` and `relation` as well as the total number of words in the fragment were used. For each fragment, the number of times each possible component of `postag` occured was counted, resulting in counts of 50 morphological features. A `relation` can consist of multiple building blocks, for example `OBJ_AP_CO`. These were split up into their respective elements, of which there were 33. After extracting all these features, columns with a maximum of one entry were removed. This resulted in a matrix of size 5295 by 80. The standard scaling implemented in LIBSVM was used, which scales the training data to zero mean and unit variance [12]. The training center and scaling values are used for later predictions.

### 3.3 Models, choice of kernel and parameters

Initially, three types of SVMs were trained for both models: binary SVMs on each author pair, binary SVMs for each author against all other authors and finally, multiclass SVMs. The SVMs for author pairs and the multiclass models can be applied if a set of possible authors of a text fragment has been identified. The one-against-rest model can be used to answer the question: is this text by author $X$ or not? The implementation of

LIBSVM in the R package e1071 was used [12, 13]. This implementation uses the one-against-one method for multiclass problems. Because a linear kernel was used for all models, only the cost parameter $C$ needed to be chosen. For each model, an SVM with $C \in \{1, 10, 100, 1000\}$ was constructed. As can be seen in table 1, the data is unbalanced. Therefore, a high-accuracy classifier might be produced by classifying any example to the majority class Homer. To prevent this, the total misclassification cost $C$ can be replaced by as many terms as there are classes. If there are $k$ classes with $n_k$ examples each, a method of choosing the $C_i, i = 1, \ldots, k$, is by setting:

$$C_i = \frac{C}{n_i}, \tag{2}$$

where $C$ is a constant. This ensures that $C_i n_i = C_j n_j$ for all $i, j$ [14]. SVMs were also trained using these class weights, with $C \in \{1000, 10000, 100000\}$. These values were chosen to make them comparable to the fixed values of $C$.

### 3.4 Evaluation of performance and cross-validation

There are multiple measures of text categorization effectiveness. One of these is precision, the fraction of fragments classified as written by $X$ that are indeed written by author $X$. For this application, precision was used as the measure of effectiveness since classicists are interested in the probability that a text classified as written by for example Homer, is actually written by Homer. Depending on the model, we have two or four precisions. To combine this into a single measure of effectiveness, we use macroaveraging. If we have $k$ authors and denote the macroaverage by $\hat{\pi}$ and the precision per author by $\hat{\pi}_i$, the macroaverage is computed as:

$$\hat{\pi} = \frac{1}{k} \sum_{i=1}^{k} \hat{\pi}_i. \tag{3}$$

This way of averaging assigns equal importance to the classification of each author [3].

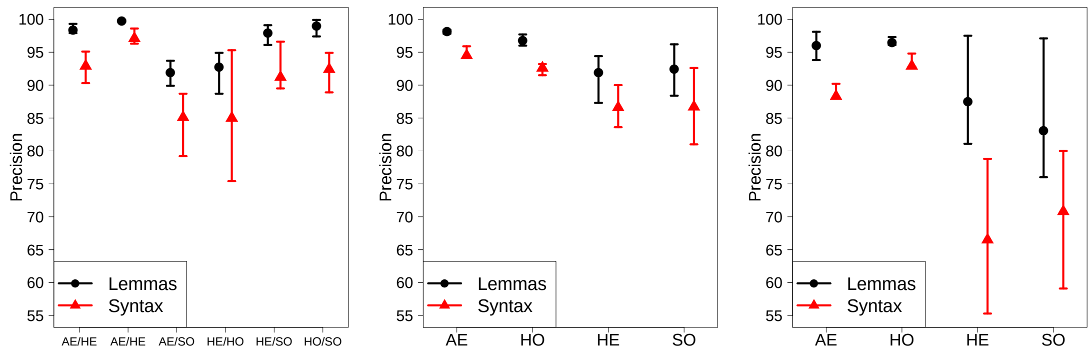
For each SVM, cross-validation was performed by partitioning the data $D$ into five random subsets $D_i, i = 1, \ldots, 5$. Then for each of the seven cost parameters as defined in 3.3, five SVMs were trained on $D \backslash D_i$ and tested on $D_i$ for $i = 1, \ldots, 5$. The macroaverage of the precision on each of the test sets was computed and averaged per cost parameter. The model with the cost parameter with the highest average precision was selected as the best model.

## 4 Results

The results for the binary models for each author pair can be found in figure 3(a). Both models perform well, with the precisions (averaged over the five test sets) ranging from 91.9 to 99.7 for the lemmas model and from 85.0 to 97.1 for the syntax model. The average precision attained by the lemmas model is 96.6, the one for the syntax model is 90.6. For each author, the lemmas model performed best, with the difference in percentage points ranging from 2.6 to 7.7.

The results for the one-against-rest models are available in figure 3(b). The precisions (averaged over the five tests sets) range from 91.9 to 98.1 for the lemmas model and from 86.6 to 94.5 for the syntax model. The average precision for the lemmas model is 94.8, the one for the syntax model is 90.1. The lemmas model again performs best for each author, with the difference in precision in percentage-points ranging from 3.6 to 5.7.

Results for the multiclass problem are reported in figure 3(c). Again the lemmas model performs best, but both models attain less precision than for the binary classification problems. The average precisions were 79.8 and 90.8 for the syntax and the lemmas models respectively. The average precisions per author range from 83.1 to 96.5 for the lemmas model and from 66.5 to 92.9 for the syntax model.




(a) Precision for pairs of Greek authors. (b) Precision for one-vs-rest models. (c) Precision per author for the multiclass model.

Figure 3: Precision per author for the three models. AE = Aeschylus, HO = Homer, HE = Hesiod, SO = Sophocles. The symbols represent the mean and the bars the lowest and highest precision from the test sets.

## 5 Discussion

These results show that the lemmas model performs better than the syntax model in every case. This was to be expected, as all words in the model were used, including topic-specific ones. If these models were tested on texts by the same authors writing about different topics, they might not perform as well. However, as all the extant work of these authors, except for Hesiod, is on topics related to the Trojan cycle and will hence contain similar words, the influence of these topic-specific words is hard to test. An advantage of the syntax model is that it is completely robust to this problem, as it only uses counts of grammatical features. A disadvantage of the syntax model is that it requires features that are more complex to extract compared to the lemmas model.

Another result is that the SVMs perform best on the binary classification problems. The results for the multiclass problem suggest that this may be due to the unbalanced data, as both multiclass models perform best on Homer and Aeschylus, the authors with the highest number of text fragments. Furthermore, for every model, a cost parameter that is constant for all classes was found to perform best. These two facts combined suggest that the class weights as defined in (2) are not effective and better results may be obtained by using different class weights. To test this hypothesis, another multiclass model was trained using 197 randomly selected text fragments from each author. The results, depicted in figure 4, are on average slightly better for the lemmas model and worse for the grammar model, with average precisions of 92.5 and 75.9. A possible explanation is that the benefits of the balanced data are negated by the drawbacks of having significantly less training data. However, a key aspect of figure 4 is the 100% precision for the minority class Sophocles in the lemmas model, indicating that the data imbalance did affect the results for the multiclass model using all data.

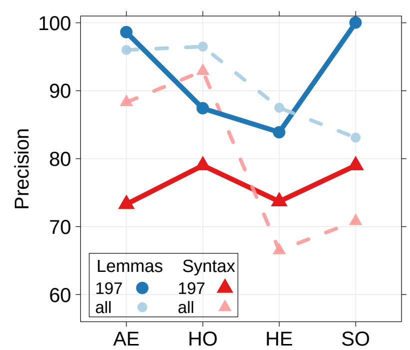



Figure 4: Precision for the multiclass model with all fragments and 197 fragments per author. The symbols represent the mean precision from the five test sets.

For the author pair models, class imbalance seems to be less of an issue, as for example 99.0% precision is obtained for the most imbalanced model of all, that of Homer and Sophocles. The precisions correspond surprisingly well to scholarly opinions on similarity of authors. The two pairs with the lowest precision, Aeschylus/Sophocles and Hesiod/Homer, both wrote their works in approximately the same time period and in the same genre: Aeschylus and Sophocles were 5th century BC tragedians, while Hesiod and Homer wrote (epic) prose in the 8th century BC. This suggests that the features in the models indeed manage to capture the style of the authors.

## 6 Conclusions and future research

Currently, the field of Classics relies on scholarly opinions only. This study introduces data-driven methods to assist classicists in classifying texts for which a maximum of two authors have been suggested. Our results are very encouraging. We have shown that even simple SVMs with features that are easily obtained and do not require complex fine-tuning do well on such binary comparisons. The models that merely use the words in a fragment outperform those that include more complex morphological information in every case. As there are many publicly available databases of classical texts from which the lemmas can be extracted, the potential is enormous. The multiclass model appears to be less useful however since it suffers from a too severe data imbalance. The results for the binary models might be improved by exploring different features and kernels, and expanded upon by including more authors and using different fragment lengths. For the multiclass model, other voting schemes and class weights can be tried. Both topics are considered as part of future research.

### References

[1] Athanassakis, A.N. (2004) *The Homeric Hymns.* Baltimore, MD: JHU Press.

[2] Kovacs, D. (2009) Do We Have the End of Sophocles' *Oedipus Tyrannus*? *The Journal of Hellenic Studies* **129**:53-70.

[3] Sebastiani, F. (2002) Machine Learning in Automated Text Categorization. *ACM Computing Surveys* **34**(1):1-47.

[4] Joachims, T. (1998) Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proceedings of the European Conference on Machine Learning*, Berlin: Springer.

[5] Leopold, E., Kindermann, J. (2002) Text Categorization with Support Vector Machines. How to Represent Texts in Input Space? *Machine Learning* **46**:423-444.

[6] Diederich, J. (2003) Authorship Attribution with Support Vector Machines. *Applied Intelligence* **19**:109-123.

[7] Koppel, M., Schler, J., Argamon, S. (2009) Computational Methods in Authorship Attribution. *Journal of the American Society for Information Science and Technology* **60**(1):9-26.

[8] Burges, C.J. (1998) A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* **2**:121-167.

[9] Hsu, C.-W., Lin, C.-J. (2002) A Comparison of Methods for Multiclass Support Vector Machines. *IEEE Transactions on Neural Networks* **13**(2):415-425.

[10] The Ancient Greek and Latin Dependency Treebanks, available from `http://nlp.perseus.tufts.edu/syntax/treebank/`. Accessed on 17-10-2011.

[11] Bamman, D., Crane, G. (2008) The Logic and Discovery of Textual Allusion. In *Proceedings of the Second Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2008)*, Marrakech.

[12] Chang, C.-C., Lin, C.-J. (2001) LIBSVM: a Library for Support Vector Machines. Maintained at `http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf`. Accessed on 21-10-2011.

[13] Dimitriadou, E., Hornik et al. (2011) e1071: Misc functions of the Department of Statistics (e1071), TU Wien. R package version 1.6.

[14] Hur, A. B., Weston, J. A (2010) A User's Guide to Support Vector Machines. *Methods in Molecular Biology* **609**(2):223-239.