Who Wrote Hebrews? A Stylometric Analysis with Word Embeddings and Word Frequencies

Jeremy Lachowicz
Roanoke College
Salem, Virginia
jdlachowicz@mail.roanoke.edu

***Abstract*—Using computers to determine style and authorship is a thriving and intriguing field. This paper applies modern computing to attempt to solve the mystery of authorship with the Book of Hebrews. To solve this, the word embedding techniques of Word2Vec and Doc2Vec are tested with cosine similarity and authorship classifier tests. Additionally, frequency analyses tests are utilized, some formulated specifically for Ancient Greek. Out of the potential authors of Hebrews tested, the author of the Pastoral Epistles is the most likely to have written Hebrews. However, this author did not pass all the tests, so it is ultimately inconclusive. Clement and Luke can be confidently ruled out of the authorship of Hebrews. With the help of machine learning, much was learned about using word embedding and frequency analysis techniques to determine authorship with Ancient Greek.**

I. Introduction

Knowing the author of a text can inform a reader of their background, bias, and purpose. It helps the reader understand the motive and thoughts of the author, and ultimately enriches the reading experience. When the author of a text is unknown, questions of credibility and motives can be felt. Therefore, determining authorship is of critical importance. Techniques for determining authorship have been around for a long time, however new technology brings new approaches. These modern techniques can help solve age old questions of authorship in fresh ways. One of the most disputed authorship problems is the Letter to the Hebrews in the Bible. Even among the greatest bible scholars, authorship is widely undecided. But with new tools to determine authorship, now is the time to reexamine this mystery through the lens of modern computing power.

II. Literature Survey

A. What is Stylometry?

Stylometry started with Augustus de Morgan in 1851, who hypothesized authors could be determined by analyzing differences in word lengths. This hypothesis was tested with discouraging results for Morgan, but still, it was this hypothesis that laid the groundwork for future Stylometrty research. From this came the application of traditional statistics to determine style in a text. Early works included finding linear relationships of vocabulary count per author under logarithmic scales, and analyzing word frequencies and their distributions. In the 1960s, Stylometry made a significant surge due to the work of Frederick Mosteller and David Wallace from their success of analyzing authorship of some of the disputed works in The Federalist Papers. They used more advanced statistical techniques and their results agreed with the popular scholarship opinion. This research positively impacted the reputation and opinion of Stylometry. From here, many univariate and multivariate approaches were utilized, as well as cumulative sum charts. Finally, in recent years, Stylometry has been paired with artificial intelligence, specifically neural networks. As a whole Stylometry can be as simple as counting word lengths, to using advanced multivariate statistical methods and neural networks. Despite the numerous approaches, there is no general consensus on which ones are good or bad, so selecting the best methods is perhaps the greatest challenge posed by Stylometry. [1]

B. Word Embeddings

One of the most promising new methods is using word embedding techniques to determine style in a corpus. Word embeddings is the technique of computing vector representation of words. These representations can capture semantic and syntactic similarities of words in a corpus. To generate these embeddings there's multiple methods, but one of the best is Word2Vec. Word2Vec learns similarities by training a neural network with one hidden layer to execute a 'fake task'. The output of the neural network is not important, but the hidden layer contains the learned weights, which is the vector representation of the word. There are two methods of training a Word2Vec model. The first is Continuous Bag of Words (CBOW) (see figure 1). In this technique, the (fake) task of the neural network is to predict a word from the surrounding context words. The second is Continuous Skip-Gram, which is tasked with predicting the context words around a certain word; therefore, the two methods are essentially opposites (see figure 2). Once the vectors from each word in a corpus are generated, vector operations such as addition, subtraction, multiplication, etc. can be used to find interesting insights of the data. For example:

King - Man + Woman = Queen

This shows a simple example of the possibilities of vector operations. [2]

Another popular word embedding technique, Doc2Vec, generates vector representations of documents rather than words.

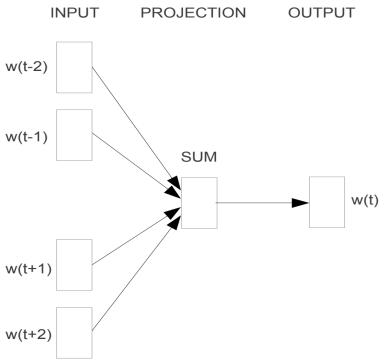

Fig. 1. Continuous bag of words model

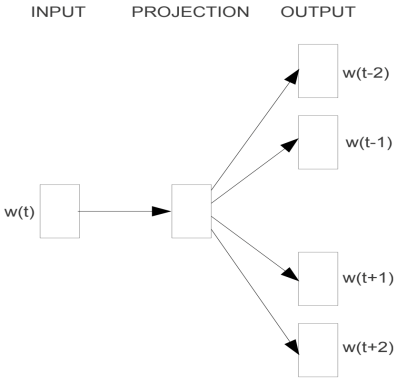

Fig. 2. Skip-gram model

Doc2Vec is based on the Word2Vec architecture and offers two types of training methods: Distributed memory (DM) and Distributed Bag of Words (DBOW) (see figure 3). DM is similar to SG from Word2Vec because the model generates vectors by trying to predict a target word from context words. However, the difference is the input, where DM gives an additional input of a document ID. This allows the model to compute a vector representation of a document, rather than just a word. Likewise, DBOW is like CBOW of Word2Vec. In this case, instead of tasking the model with predicting context words of a target word, it tasks the model with predicting context words of a document ID. [3]

C. Word Embeddings for Authorship Attribution

Previous research indicates word embeddings can be a powerful tool to determining authorship of a document. [4] proposed an approach of using six word vectors generated by Word2Vec to represent an entire document. For this approach, the texts of a single author are grouped into a text file. Word2Vec is applied to all these files to generate vector representations of the words. From here, they compute similar words between all the documents, and randomly select six words to represent each document. To clarify, each document is represented by the same six words chosen randomly. Then,

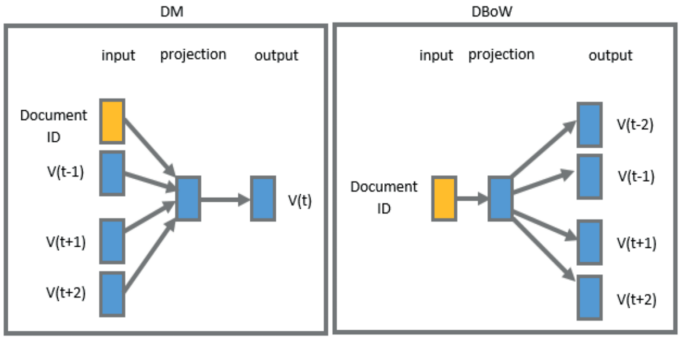

Fig. 3. Doc2Vec Models

each author's vectors representations of the selected words are given to a Multilayer Perceptron (MLP) classifier as a training set. The test set contains the six vector representations from the document whose authorship is in question. The MLP classifier returns the probability of the authors (classes) belonging to the test set. For Doc2Vec, the process is more straightforward. All of the texts of a single author are placed in a text file, then the text files of all authors are passed to the Doc2Vec model to generate document vector representations. From here, there are two options for testing. The first method for testing is by inferring a vector from the Doc2Vec model, and the second one is retraining the model with the test document. Inferring the vector restricts the vocabulary to the vocabulary of the training set, while retraining the model uses all the vocabulary. However, retraining is faulty machine learning as the model sees the test instances during training. Nonetheless, retraining is the more chosen method. [5] In this paper, inferring the vector is used rather than retraining.

For Word2Vec and Doc2Vec having large corpora is important. [6] used roughly 2,250,000 words per author. [4] used 72 documents per author, while [5] used 30 documents per author. These techniques have been applied to multiple languages, namely Bengali, English, Italian, Spanish, and French. [6] [5] However, they have not been applied to a reasonable degree to Ancient Greek, which is a highly inflected language. Likewise, Bengali is a highly inflected language that gave great results with Doc2Vec, which provides some comfort about using word embeddings in the largely untested Ancient Greek language.

Hyperparameters and input types for for Word2Vec or Doc2Vec receive conflicting research. [6] found the skip-gram model of Word2Vec to perform better, while [4] found the CBOW model to perform better. For Doc2Vec, [5] did not find a significant difference between the two methods of DBOW and DM. Due to this, CBOW, SG, DBOW, and DM are all tested. Furthermore, specific hyperparamters differ within papers, but one paper suggested the use of **LayerSize - 50, WindowSize - 5, MinWordFrequency - 1, Iterations - 3, LearningRate - 1.0E-4, Sampling - 1.0E-5** for W2V [4]. Lastly, [7] claimed improved results when combining models in an ensemble.

III. Who Wrote Hebrews?

The authorship of the Letter of Hebrews found in the New Testament is unknown. Traditionally, the primary candidate is the apostle Paul, who wrote numerous other New Testament epistles. In his previous works, he claimed authorship within the first sentence. No claim of authorship is found in Hebrews, however. Pauline authorship defenders argue he did not claim this because he was not well liked with the Jewish population at the time. They also cite a few early church fathers who claimed Paul wrote Hebrews. [8]

The evidence against Pauline authorship is rather large. First, vocabulary usage is different from his other works. Particularly, there are 154 words that appear only once in Hebrews, which is a much greater count than Paul's other texts. Secondly, the writing quality is more advanced, with more ornate and sophisticated Greek than Paul's other writings. This indicates a person who was highly educated in Greek would have written it. Thirdly, there are thematic differences. One example is the number of times the author talks about the resurrection of Jesus, which is a much smaller count in Hebrews than in his epistles. The evidence against Pauline authorship is quite large, therefore the popular scholarly opinion has swayed from Paul to other candidates. The second candidate is Clement. He authored a letter to the church in Corinth that contained similar themes and quotes from Hebrews. The third candidate is Apollos. He was highly educated and associated with Paul. He was also Jewish, so his knowledge of the Old Testament would coincide with the knowledge the author of Hebrews would require. Next is Barnabas, who was named the author by an early church father, but there is no other substantial evidence. The next candidate is Luke, who wrote other New Testament works around the same time period as Hebrews. The final candidate is the unknown author of the Pastoral Epistles in the New Testament. The Pastoral Epistles are a compilation of three works, and while authorship is unknown, it is believed they are written by the same person. For both Luke and the author of the Pastoral Epistles, there is not substantial evidence for authorship, however being in the time period and writing about similar themes are enough to warrant a test. [8]

To test these potential authors, there must be previous works from these authors to compare with Hebrews. Unfortunately, perhaps the strongest candidate, Apollos, has no written works, as well as Barnabas. For Paul, his works include Galatians, Thessalonians, First and Second Corinthians, Philippians, Philemon, and Romans. For Clement, there is First Clement. For Luke, there is the Gospel of Luke and Acts, For the Pastoral Epistles there is First and Second Timothy and Titus. The goal of this paper is to determine, between Paul, Luke, Clement, and the author of the Pastoral Epistles, if any could have written Hebrews, and if so, which one. If there is no clear answer to this mystery, at least it may be possible to find which candidates are least likely to have authored Hebrews.

IV. Methodology

The previous studies focused on a data set where the text could only be authored by one of the authors in the data set. However, in this case, all of the possible authors of the Book of Hebrews can not be tested. This is why it is absolutely crucial to use a mixed-method approach. In particular, the methods of cosine similarity, Word2Vec, Doc2Vec, and frequency analyses will be considered. To develop, tune, and test the methods, a testing set will be utilized where the author of the test document is known. In the training set, Ignatius is the author that will simulate Hebrews (see figure 4,5). Once the methods are properly tuned to work as best as possible for this testing set, only then will the methods be applied to the Hebrews data set. This is to ensure the methods can produce sufficient results for the Ancient Greek language and to ensure the models are tuned and shaped to develop a significantly powerful assessment for the Hebrews data set.

Author	Works	Length(words)
Mark	Gospel of Mark	11,288
Mathew	Gospel of Mathew	18,852
Longus	Daphnis and Chloe	20,306
Ignatius(train)	Letter to Romans, Smyrnaeans, Polycarp, Philadelphians	3,999
Ignatius(test)	Letter to Ephesians, Magnesians, Trallians	3,857

Fig. 4. Training Dataset

Author	Works	Length (words)
Clement	First Epistle of Clement	10,233
Luke	Gospel of Luke, Acts	37,956
Author of the Pastoral Epistles	I & II Timothy, Titus	3,493
Paul	I Thessalonians, Galatians, I & II Corinthians, Philippians, Philemon, Romans	24,122
Unknown*	**Hebrews**	**4,955**

Fig. 5. Hebrews Dataset

A. Preprocessing

Before any analysis of the corpora, the texts must be cleaned. Some of the text files have indicators of chapter and line, so that is removed as well as all numbers. From here, there is the option of getting rid of stop words. Stop words are the most common words in a specific language and are usually

filtered out. However, due to the lack of data, the Word2Vec and Doc2Vec classifier methods are tested with stop words and without stop words. Once the texts are cleaned, they are ready for frequency analysis and Doc2Vec, but not Word2Vec. For Word2Vec the texts are transformed into a list of sentences before passing on to the model.

B. Cosine Similarity

The classifier's used on Word2Vec and Doc2Vec output the percentages of which class the test document most likely belongs to. This is a problem if the author is not one of classes. For example, if the author of "Green Eggs and Ham" was tested with works from J.K Rowling and Charles Dickens as the training set, the classifier would still predict one of them, even though it is clear neither of them are the author. This is why similarity measures are so important in this case. Similarity measures are not forced to choose a class and they return an unbiased similarity score. In this respect, there is high dimensional data, so cosine similarity is the best measure because it computes the cosine angle between the vectors, rather than simply the distance between the vectors. [9] The cosine similarity with A and B as vectors can be computed as follows,

$$sim(A, B) = \frac{A * B}{\|A\|\|B\|} \tag{1}$$

If the two vectors are similar, they should be pointing in the same direction. A cosine similarity score of 1 would indicate strongly similar vectors, while a score of -1 would indicate strongly opposite vectors. For Word2Vec, the cosine similarity of the vectors will be tested in two ways. First, each word of the Word2Vec model of the test author is iterated. Each iteration, if the word is found in the model of all of the other authors (n = 4), the cosine similarity is computed between the word vector's of the author's and the test author's word vector. If the word is found in at least one of the author models, rather than all of them, that is tested separately. So $n > 1$ and n = 4 are tested, where n is the number of authors that contain the word in their model. Then, the cosine similarity is added to a list and averaged per author at the end of the test. So essentially, this method is an average cosine similarity (per author) of each shared word with the anonymous author's Word2Vec model. The second method is similar to the first, but instead of averaging, a count of which author's model had the most similar word for each word in the test document is kept. This count is used to generate a percentage of how many words in the document were most like author x, author y, etc. While these measures are straightforward, they still test the documents in a thorough manner. For Doc2Vec, due to the vectorization of the entire document, rather than the individual words of the document, cosine similarity is simple. Both of these methods include an extensive grid search with various hyperparameters.

C. Word2Vec and Doc2Vec Classifier Tests

For the Word2Vec classifier tests, the preprocessed texts are vectorized and the words to represent the documents are chosen. There are three approaches to selecting these words. First, a small sample of the most common word frequencies in all the texts are chosen. Secondly, a larger sample of the most common word frequencies in the texts are chosen. Third, all the similar words are chosen. Once these word vectors are combined to generate a matrix to represent the author, a grid search is ran with an MLP classifier. The results from the cosine similarity Word2Vec grid search are used to determine a baseline of hyperparameter values to include in the grid. The closest iterations to 100% probability of the anonymous text belonging to Ignatius are recorded. Furthermore, the hyperparameters used for Word2Vec are alpha, min count, sample, sg, and size. [10] These are specifically chosen due to the success of these hyperparameters in previous research. [7]

For the Doc2Vec classifier tests, the preprocessed texts are vectorized and the document vectors are tested with an MLP classifier in two ways. First, the documents will be split by paragraph. To determine what constitutes a paragraph, groups of 3, 5, and 7 sentences will be tested. Secondly, the entire document will be used, so each author only has one training sample. The second method is the traditional approach, but due to the severe lack of data, generating more training samples as paragraphs could be important. Both of these methods are then ran through a grid search. The results from the cosine similarity Doc2Vec grid search are used to determine a baseline of hyperparameter values to include in the grid. The closest iterations to 100% probability of the anonymous text belonging to Ignatius are recorded. Lastly, the hyperparamters tested are alpha, min count, vector size, and epochs. [11] Previous research did not indicate specific hyperparameters that were effective. Therefore, alpha, min count, and vector size were chosen due to the success with Word2Vec. Epochs was chosen because it could make a large difference for the limited data in the data sets because it controls the number of iterations of the corpus during training.

The final Word2Vec and Doc2Vec method is using part of speech (POS) tags instead of words as the input to the models. POS tags are simply the part of speech of the word. So instead of the input to the model being "the young man was standing", it would be "article adjective noun verb". This would let the model learn about how the author uses specific nouns, verbs, etc., which could be a great indicator of style. With limited data, there is a limited vocabulary for the model to learn, and little instances of each word. This technique would supply the model with less vocabulary words overall, but more instances for each word. To represent the documents, three word list options are tested. The first is all the similar words in their full format, so their part of speech, number, gender, tense, etc. are tested. The next option is a small sample of the most frequent words between the documents in their full format. Then, a larger sample of words will be tested. For the next options, the tags will be transformed into simplified format, so instead of representing the words in their full description, a simplified description is used. For example, the full format for a word could be "pronoun, plural, neuter, accusative", but the

simplified format would just be ”pronoun”. For the simplified versions, all similar tags will be tested and a small sample of only important tags (noun, verb, etc.) will be tested.

Finally, once the best Word2Vec and Doc2Vec models are determined from the above methods, they are concatenated into a combined model. To concatenate the models, Word2Vec and Doc2Vec must have the same vector size. The best hyperparameters for the same vector sizes are chosen, and the Doc2Vec document models are added to the Word2Vec document models. Then, an MLP classifier is used as previously described to determine the predicted class.

D. Frequency Analysis Tests

For the following frequency analyses, a G-test is used to determine statistical differences in word frequencies, which is recommended for use with word frequency proportions. [12]

The first test is a comprehensive word frequency analysis. First, each word of the test corpus is iterated. Each iteration, if the word is found in all of the other documents, the word frequency for all documents is calculated, and the G-test computes statistical significance of these frequencies. This process is repeated until all the words of the test corpus are iterated. Due to the uneven document sizes, the counts of the significant words will be divided by the number of words tested. Additionally, the differences in frequencies between the training authors' words and the test author's words will be calculated, and a count of which author's word frequency difference is the smallest will be counted per word.

The next test requires the usage of a Greek tagger. [13] This tagger computes information about the word's part of speech, person, number, tense, mood, voice, gender, case, and degree by taking into account the word's usage in the sentence and known information about the word. With this information and with advice from Dr. Jennifer Berenson, multiple Ancient Greek specific measures of style were created. The frequencies of the optative mood indicates a higher style, as well as the frequency of the future and pluperfect tenses. These frequencies are calculated and then tested for statistical significance with the G-test. The next test is the frequency of participles before or after the main verb of the sentence. The main verb is the indicative and imperative verb. Due to the lack of punctuation usage in Ancient Greek, it is also important to split the sentence when a conjunction word is encountered. From here the frequency of participles before and after the main verb is calculated and the frequencies are tested using the G-test.

V. Results

A. Cosine Similarity

For the training data set, the cosine similarity tests produced promising results. The best hyperparameters return a 0.99 mean similarity per word for Ignatius (see figure 6). The best three hyperparameter combinations were applied to the Hebrews data set. These results disclosed the author of the Pastoral Epistles as the highest in mean cosine similarity and word count measures, with Clement lagging slightly behind (see figure 7).

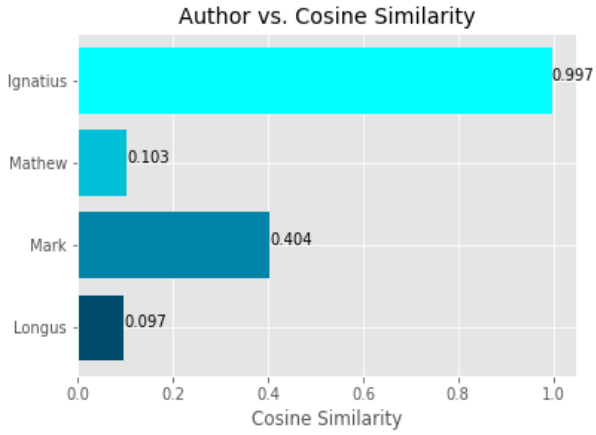

Fig. 6. Training Set: Mean Word Cosine Similarity

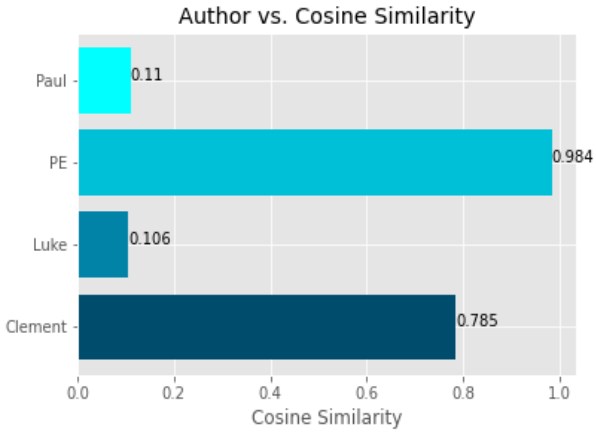

Fig. 7. Hebrews Set: Mean Word Cosine Similarity

It is interesting that the words in the Pastoral Epistles are strongly correlated with the words from Hebrews, while Paul and Luke's words are not. Upon changing the minimum number of authors that contain the word in their model from $n > 1$ to $n > 3$, the results for word count are even more staggering for the Pastoral Epistles (see table 1).

TABLE I
PERCENTAGE OF WORDS CLOSEST TO THE TEST SET

Author	Percentage		
	n>1	n>2	n>3
Clement	33.887	37.255	0
Luke	10.299	0	0
Pastoral Epistles	45.515	62.745	100
Paul	10.299	0	0

The cosine similarity measures for Doc2Vec with the training data also returned promising results (see figure 8 for best

grid search iteration).

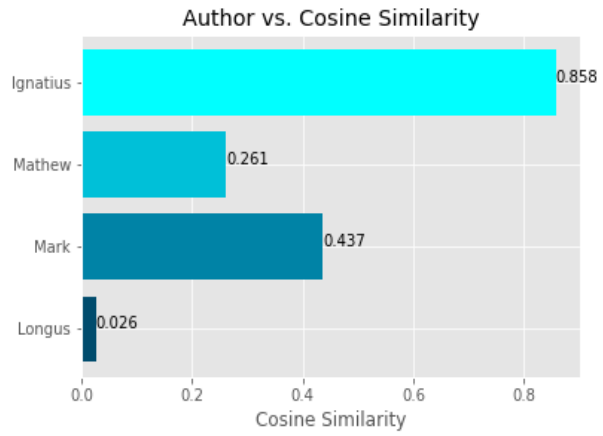

Fig. 8. Training Set: Mean Document Cosine Similarity

However, there was some random variation with inferring the vector. Therefore, once the best grid search results were determined, the cosine similarity for the Hebrews data set ran 20 times for the best three hyperparameter combinations to account for the random variation in the infer vector algorithm. From this, the results showed slightly similar document vectors for Clement, Luke, and the author of the Pastoral Epistles with all of them fairly close together except for Paul (see figure 9 for results from best hyperparameters). Also note all three hyperparameter sets tested returned almost identical results.

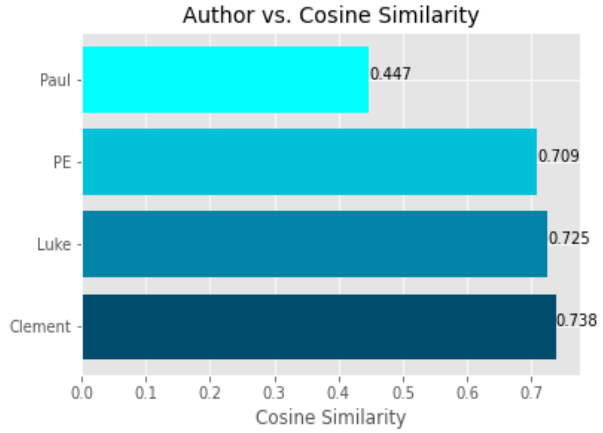

Fig. 9. Test Set: Mean Document Cosine Similarity

B. Word2Vec and Doc2Vec Classifier Tests

For the training data set, the Word2Vec classifier tests produced remarkable results. The small sample of the most frequent words worked better than the larger sample and all similar words, so the small sample was chosen for the grid search. Additionally, when stop words were removed from the model, the percentages hardly changed, so the inclusion or exclusion of stop words was not significant. In the grid search, the best iteration classified Ignatius as 94% likely, and there were multiple other hyperparameter combinations that produced >90% results for Ignatius (see figure 10). When

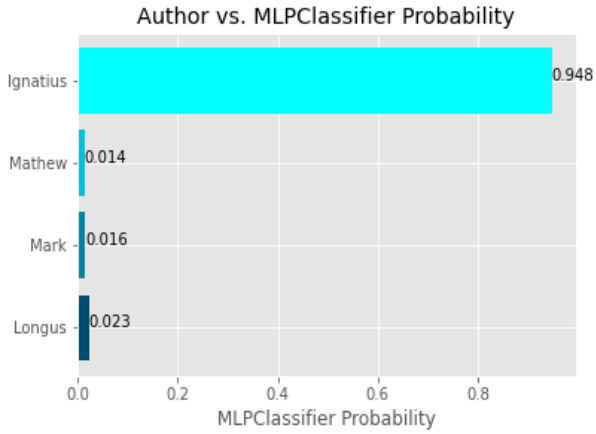

Fig. 10. Training Set: Word2Vec Classifier Test

using this method for the Hebrews data set, the Pastoral epistles were classified as 89% likely, with the other authors much lower (see table 11).

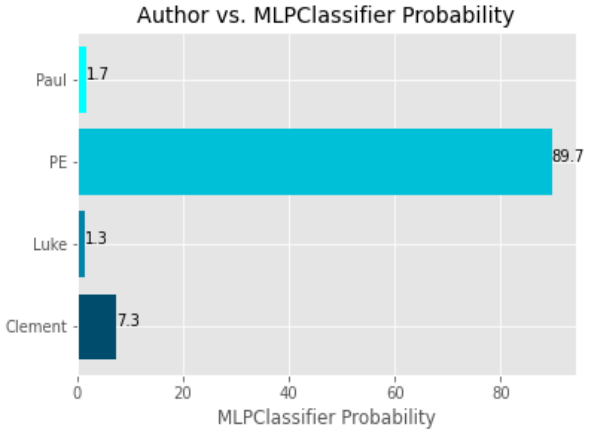

Fig. 11. Test Set: Word2Vec Classifier Test

The next test was using POS tags as input. For this test on the training data, the results dropped slightly from the previous 94% from using words as input to 88% Ignatius. The best option for the chosen words to represent the documents was all similar words in full format, however the other options were not significantly worse (see figure 12). When testing this on the Hebrews data set, the Pastoral Epistles were again found to be the most likely class by far, but the percentage dropped from 89% to 76% (see figure 13).

The Doc2Vec model did not perform as well as the Word2Vec model. One problem encountered when performing the tests was the randomization of the MLP Classifier. This

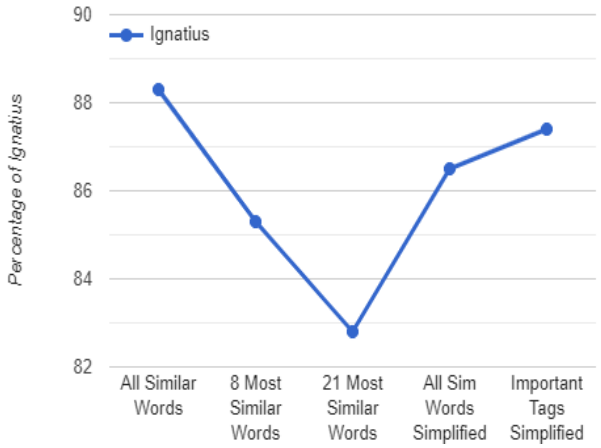

Fig. 12. Training Dataset: POS Tags Word2Vec MLPClassifier Ignatius Probability

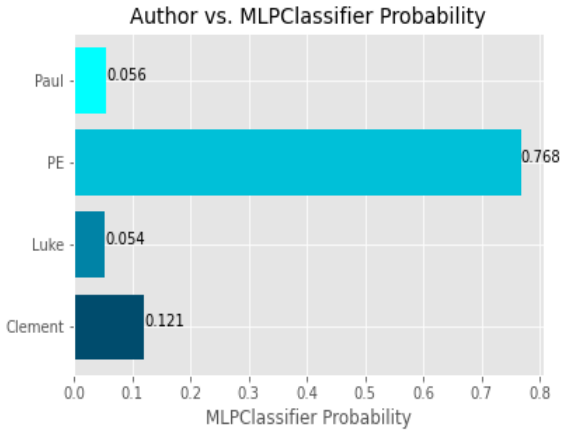

Fig. 13. Hebrews set: POS Tags test

produced almost completely random results each time with the Doc2Vec models. Therefore, Decision Tree, SVC, Linear SVC, and a Logisitic Regression classifier were tested and a logisitic regression classifier worked the best. The logistic regression classifier produced stable and accurate results. From here, the Doc2Vec model as the entire document as the input was tested. Then, the document with paragraph sizes of 3, 5, and 7 were tested (see table 2).

TABLE II
TRAINING SET: DOC2VEC CLASSIFIER TEST

Author	Average Logistic Regression Percentage (100 iterations)			
	Para. 3	Para. 5	Para. 7	Entire Document
Longus	0.03	0.02	0.003	0.02
Mark	0.3	0.3	0.29	0.28
Mathew	14.068	11.70	15.095	11.91
Ignatius	85.597	87.978	84.6	87.780

The paragraph size of 5 was similar in result to the entire document as input, while 7 and 3 sentences were slightly worse. All of these options returned a high likelihood of Ignatius. When testing the Hebrews data set, the entire document as input and paragraph size of 5 as input were considered. The logistic regression found Paul to be the predicted author of both input types (see table 3). Ultimately, the results were favorable for Paul, however it was not intensely confident as the results for the Word2Vec.

TABLE III
HEBREWS SET: DOC2VEC CLASSIFIER TEST

Author	Average Logistic Regression Percentage (40 iter.)	
	Para. 5	Entire Document
Clement	2.943	2.920
Luke	17.489	15.997
Pastoral Epistle	7.817	7.473
Paul	71.749	73.609

An important note was that when removing stop words from the texts, the Doc2Vec model improved, so stop words were removed for all the D2V tests. The final test for the Doc2Vec was using POS tags as input. This did not work well (see table 4), so it was not tested with the Hebrews data.

TABLE IV
TRAINING SET: POS DOC2VEC CLASSIFIER TEST

Author	Average Logisitic Regression Percentage (40 iterations)	
	Full POS Tags	Simplified POS Tags
Longus	21.982	26.325
Mark	35.301	21.283
Mathew	30.039	42.761
Ignatius	12.677	9.629

Finally, when concatenating the best performing Word2Vec and Doc2Vec models, the probability of Ignatius after 20 iterations was 87.1% with an MLP Classifier and 86.6% with a Logisitic Regression Classifier. This result is worse than the Word2Vec test, but better than the Doc2Vec test. When testing the concatenation on the Hebrews data, the results barely changed from the original Word2Vec classifier test when using an MLP Classifier. When using a Logisitic Regression classifier, the results were mixed (see table 5).

TABLE V
WORD2VEC AND DOC2VEC CONCATENATION TEST

Author	Probability	
	MLP Classifier	Logisitic Regression Classifier
Clement	10.212	35.007
Luke	0.481	11.172
Pastoral	89.151	36.110
Paul	0.155	17.710

C. Frequency Analysis

The first word frequency analysis method is the frequency of significantly different words for all the words in the test document. The next measure is the count of which author had the smallest frequency difference per word. Ignatius had the lowest frequency of significantly different words and the

highest percentage of smallest frequency differences (see table 6). Also, keeping stop words in the text helped the results slightly.

TABLE VI
TRAINING SET: SIGNIFICANTLY DIFFERENT WORDS

Author	Occurrences	Frequency	Smallest Diff. in Frequencies
Longus	91	0.4739	22.396%
Mark	90	0.469	20.312%
Mathew	66	0.344	6.250%
Ignatius	13	0.0671	51.042%

The next tests were testing for frequency differences in optative, future perfect tense, and pluperfect tense usage among the document. There were only two future perfect tense usages found within all the documents, so this is not a valid comparison. For all three of these, there was not a significant difference for Ignatius, but there was also not a clear indication that the others were significantly different across the tests. For the participle frequency analysis, Ignatius was found to have significantly different participle usage than the test document. So overall these results were okay at differentiating between the styles of the documents, but not great. When testing on the Hebrews data set, the author of the Pastoral Epistles had the lowest frequency of significantly different words and the highest percentage of the smallest word differences. (see table 7).

TABLE VII
HEBREWS SET: SIGNIFICANTLY DIFFERENT WORDS

Author	Occurrences	Frequency	Smallest Diff. in Freq.
Clement	57	0.244	40.171%
Luke	124	0.530	6.838%
Pastoral Epistles	37	0.158	50.427%
Paul	113	0.483	2.564%

For the optative mood usage test, Luke and Paul used it significantly different than Hebrews, while there was no future perfect usage between the documents. The pluperfect tests indicate none of the authors use it significantly different than Hebrews. For the participle analysis test, all the authors except for Clement used participles differently than Hebrews. Overall, the test for tenses are mixed, and do not tell much about the authorship of Hebrews.

VI. DISCUSSION

It is interesting that the common word lists used when representing a document for classification in Word2Vec have relatively little impact on the accuracy. It was hypothesized that this would make a meaningful difference than prior research which just used a small sample of random words to represent the document. Further studies should investigate combinations of words to represent a Word2Vec model for classification. Another hypothesis that was false was the impact of using POS tags on the model. This did not improve the models, but made them slightly worse. Furthermore, the hypothesis of using paragraphs as input to the Doc2Vec model, rather than an entire document was also slightly wrong. Paragraph sizes of 5 were better than the entire document at predicting Ignatius, but only by 0.198%, while the paragraph sizes of 3 and 7 made the model worse by about 2%. This shows using one large piece of data as input per author is comparable to using the same piece of data sliced into smaller pieces. Future research on different types of inputs to the Doc2Vec model and different sizes of input could explain the little impact seen here.

Ultimately, the results are not clear. Across the measures, there was variation in the predicted authorship for the Hebrews set, which was not seen in the training set. The W2V cosine similarity and the W2V classifier tests produced very strong evidence for the author of the Pastoral Epistles. The Doc2Vec cosine similarity results did not reveal a strong predicted author, while the Doc2Vec classifier tests revealed Paul as the predicted author, although with not as much confidence as W2V. Finally, for the frequency analyses, it could not agree on a predicted author. However, even for the training set, some of the frequency analysis tests were not satisfactory with predicting Ignatius, except for the test of significantly different words and count of smallest word difference. This could be because of the lack of textual data. With more words there would be more data to differentiate between tense usage across the texts. The two satisfactory tests for word frequency predicted the author of the Pastoral Epistles as the author of Hebrews. When comparing Word2Vec and Doc2Vec, Word2Vec predicted Ignatius with more confidence in the cosine similarity and classifier tests. Thus, the Word2Vec classifier test is the best indicator of style from all of the training tests. When analyzing and weighing the precision of each test, the author of the Pastoral Epistles is the predicted author. This author had favorable results with Word2Vec and the frequency analyses tests. In fact, the only test the Pastoral Epistles clearly lost was the Doc2Vec classifier tests. With that being said, claiming the author of the Pastoral Epistles is the author of Hebrews with this information is a stretch. Clearly, in the training set, Ignatius was predicted in all of the tests fairly significantly. There is no reason to think this would not be the same result if the author of Hebrews was in the Hebrews data set.

As discussed previously, the cosine similarity and frequency analyses tests do not have to ”chose” an author like the classifier tests, so if the author were not in the Hebrews set, the author would not perform well in these tests. Nonetheless, the author of the Pastoral Epistles was the most similar to Hebrews with a Word2Vec vector representation. In fact, the author scored a 0.984 on the similarity scale, while Ignatius scored a similar 0.997. The best test from the frequency analyses show significant different words in Hebrews with a 0.158 frequency, which is slightly greater than Ignatius' 0.0671. In both of these tests, the author of the Pastoral Epistles won and received similar scores to Ignatius. Despite this, there is still the Doc2Vec classifier test that disturbs this authorship claim. Although we might not be able to deduce for certain the author

of Hebrews, we did learn that Clement and Luke are highly unlikely to be the author, while Paul is slightly unlikely.

REFERENCES

[1] D. I. HOLMES, "The evolution of stylometry in humanities scholarship," *Literary and Linguistic Computing*, vol. 13, no. 3, p. 111–117, 1998.

[2] D. Jurafsky and J. H. Martin, *Vector Semantics and Embeddings*. Pearson, 2020, p. 1–11.

[3] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *CoRR*, vol. abs/1405.4053, 2014. [Online]. Available: http://arxiv.org/abs/1405.4053

[4] N. E. Benzebouchi, N. Azizi, N. E. Hammami, D. Schwab, M. C. E. Khelaifia, and M. Aldwairi, "Authors' writing styles based authorship identification system using the text representation vector," in *2019 16th International Multi-Conference on Systems, Signals 'I& Devices (SSD)*. IEEE, Mar 2019, p. 371–376. [Online]. Available: https://ieeexplore.ieee.org/document/8894872/

[5] H. Gómez-Adorno, J.-P. Posadas-Durán, G. Sidorov, and D. Pinto, "Document embeddings learned on various types of n-grams for cross-topic authorship attribution," *Computing*, vol. 100, no. 7, p. 741–756, Jul 2018. [Online]. Available: http://link.springer.com/10.1007/s00607-018-0587-8

[6] H. Ahmed Chowdhury, M. A. Haque Imon, and M. S. Islam, "A comparative analysis of word embedding representations in authorship attribution of bengali literature," in *2018 21st International Conference of Computer and Information Technology (ICCIT)*. IEEE, Dec 2018, p. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8631977/

[7] M. Rahgouy, H. Babaei Giglou, T. Rahgooy, M. Karami, and E. Mohammadzadeh, "Cross-domain authorship attribution: Author identification using a multi-aspect ensemble approach notebook for pan at clef 2019," 07 2020.

[8] K. P. Jackson, F. F. Judd, A. B. Morrison, K. P. Jackson, K. Muhlestein, J. C. Lane, C. W. Griffin, C. F. Ellertson, T. A. Wayment, G. Strathearn, and et al., *Authorship of the Epistle to the Hebrews*. Religious Studies Center, Brigham Young University, 2006, p. 243–255.

[9] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[10] "Gensim: Topic modelling for humans," Dec 2021. [Online]. Available: https://radimrehurek.com/gensim/models/word2vec.html

[11] Metin Bilgin, "Doc2vec (dm and dbow)," [Online; accessed April 8, 2022]. [Online]. Available: https://www.researchgate.net/figure/Doc2Vec-DM-and-DBoW$_{fig3_320829283}$

[12] "Statistics in corpus linguistics," 2021. [Online]. Available: http://corpora.lancs.ac.uk/clmtp/2-stat.php

[13] Chrisdrymon, "Chrisdrymon/angel: An ancient greek morphology tagger." [Online]. Available: https://github.com/chrisdrymon/angel