

Automated Text Categorization and Metabarcoding Reinforces *Cryptococcus Neoformans*–Woody Decomposition Association

DeAndre Tomlinson

Publication Date

21-12-2023

License

This work is made available under a Public Domain Mark 1.0 (No Copyright) license and should only be used in accordance with that license.

Citation for this work (American Psychological Association 7th edition)

Tomlinson, D. (2020). *Automated Text Categorization and Metabarcoding Reinforces *Cryptococcus Neoformans*–Woody Decomposition Association* (Version 1). University of Notre Dame.
<https://doi.org/10.7274/24884850.v1>

This work was downloaded from CurateND, the University of Notre Dame's institutional repository.

For more information about this work, to report or an issue, or to preserve and share your original work, please contact the CurateND team for assistance at curate@nd.edu.

Automated Text Categorization and Metabarcoding Reinforces *Cryptococcus Neoformans*-Woody Decomposition Association

Predicting Presence of *Cryptococcus Neoformans* Using Topic Modelling on Academic Literature

DeAndre Tomlinson

ABSTRACT

This paper displays the results for predicting whether or not *Cryptococcus Neoformans* was found in different DNA samples based entirely upon the contents of associated academic papers. This is not the published version of this report.

KEYWORDS

Genomics, Clustering, *Cryptococcus Neoformans*, Text Mining, Natural Language Processing, Latent Dirichlet Allocation, Topic Modelling, Random Forest

ACM Reference format:

DeAndre Tomlinson. 2019. Academic Analysis Using Natural Language Processing and Clustering: Predicting Presence of *C. Neoformans* Using Topic Modelling on Academic Literature, *Unpublished*.

1 Introduction

Cryptococcus Neoformans is a fungus that lives all around the world and is fairly ubiquitous. It can be inhaled by anyone and is rarely studied or discovered. For a normal, healthy person with a fully functioning immune system, *C. Neoformans* is relatively harmless. However, bodies with immunodeficiencies, in particular those with HIV/AIDS, can develop cryptococcal meningitis, resulting in nearly 200,000 deaths per year. The problem is particularly pervasive in developing countries, affecting many in sub-Saharan Africa. There is a lack of targeted research due to the fungus having little to no effect on healthy individuals or those individuals in wealthy countries.

Though little research has focused directly on finding *C. Neoformans*, there has been a large amount of research that analyzes DNA samples from varying environments. These environments can be anything from soil samples to fecal samples. Researchers use a technique called barcoding to allow the DNA to be analyzed; this technique isolates and magnifies a certain segment of DNA between organisms that

is similar enough to identify the region, but is different enough to distinguish the organisms. The researchers then release this barcoding data into the public domain through easily accessed databases.

The research articles analyzed in this paper did not directly mention *C. Neoformans*, but it is possible to use DNA analysis pipelines to detect if their barcoding data contained the fungus. I hypothesized that we could predict whether or not one of these datasets contained the fungus through only analyzing the contents of the papers through topic modelling and clustering. Accomplishing this goal allows researchers looking into *C. Neoformans* to narrow down their search for the bacteria through eliminating some papers and their associated datasets and therefore reducing time spent doing DNA analysis.

2 Related Work

Natural language processing and topic modelling have been used in a wide range of applications. Other research has applied this area of data mining to analyze research papers. Primarily, other work has been in the area of attempting to identify the topics of various academic papers in order to make research more efficient. The closest work to ours in the biological sphere is a paper in which researchers used text mining to find the presence of bacteria in various papers and assess their pathogenicity [1]. However, this work involved finding the actual name of the organism in the paper; no work that I have found has used topic modelling or natural language processing to assess undiscussed but related biological information.

There are instances of other researchers using topic modelling to predict undiscussed but related data from the content of papers. For instance, one group of researchers built an LDA model to try and predict the commenter response to political articles [2]. Though very different considerations had to be made to predict virulent bacterial presence than those made for predicting possibly virulent comment sections, I looked at that research as evidence that this sort of project could work.

3 Problem Definition

Can we predict which papers are associated with data sets that contain *C. Neoformans*? How well does this model perform?

4 Data

4.1 Data Collection

Most of the data was gathered and classified before I began working on this project. My colleague, David Molik, found microbiome papers that could plausibly have used a dataset that contained *C. Neoformans*, did not directly reference *C. Neoformans*, and had an associated publicly available dataset of bacteria on NCBI. My colleague then analyzed the DNA dataset and labelled the papers based on whether the data contained or did not contain *C. Neoformans*.

4.2 Data Pre-Cleaning

Once the journal articles were collected, they were gathered into a text corpus. This corpus allowed for easy textual analysis. The corpus allowed for a three step data cleaning pre-processing. First, several algorithms, primarily from gensim and NLTK, were used to remove several articles of speech, symbols, punctuation, common English stop words, and unique words that the group has determined to be unimportant for the purposes of this project. An example of a unique word for this project was "http," which showed up in many of the references the papers that were analyzing. Also, for a more stringent approach, the minimum number of characters that a word must have was 4, so any word below 4 characters was removed during this step. The next step was the lemmatization and stemming of the text. Lemmatization is the transforming of word endings to their base part of speech, so any participle endings were changed to their normal form. Gensim was used in the lemmatization step. Afterwards, the NLTK stemming algorithm cut the lemmatized word into its essential root form. This improves computational processing time while preserving the meaning of each word. The last step of preprocessing was the creation of n-grams using gensim. Unigrams, bi-grams, and tri-grams were created to better capture the meaning from groups of words.

5 Methodology

5.1 Latent Dirichlet Allocation

After every paper was processed as described above, the remaining words were totaled with the results used to populate a matrix relating each paper to the number of times a given word was used. Gensim's Latent Dirichlet Allocation (LDA) model was then applied to the documents to create features. LDA models human language using the assumption that documents are created probabilistically using a Dirichlet

distribution. The model assumes that one starts with a list of words, fills out a group of topics with probabilities of words, and fills out a list of documents with probabilities of topics. Finally, it assumes that the documents are filled out with words given the set probabilities of the topics and associated words.

When the model was implemented in this case, the model produced a list of topics made up of different weights of words that are essentially clusters of words. The number of topics is specified beforehand by the user; the number of clusters that produced the best classifications in this case were $n=3$ and $n=4$. LDA then assigned weights to documents based on their adherence to the different topics between 0 and 1 (e.g. 0.6 for topic A and 0.4 for topic B). This assignment can be considered to be soft clustering. In topic modelling and natural language processing in general, this sort of soft clustering makes sense because it models how documents actually are in the real world. Real documents do not fit into just "science" or "math" or "logic"; they tend to be combinations of several different topics. I then used these topic scores as features to classify the documents.

6 Evaluation

After completing the LDA process, there was a list of documents reduced to an id number, the weights assigned to each topic, and the ground truth label for each document. This core data was used to train and test every classifier that I used to predict labels on the dataset. Since this dataset was small, and due to the computationally and highly manual process of acquiring more good quality documents, we decided to use all of the positive and negative documents we could, even though it resulted in an unbalanced dataset. The full set had 117 documents in total. 83 documents, or roughly 70%, were used as training data for both the LDA clustering model and the final classifiers. The other 34 documents, roughly 30%, were reserved to validate the training for the final classifier. These documents assigned topic weights using the same LDA model trained on the larger portion of documents, but were not included in the training of that model so as to keep the model unbiased on the testing data. Finally, both the training set and the testing set were composed of roughly 70% documents associated with evidence *C. Neoformans* (positives) and 30% documents which found no evidence of *C. Neoformans* (negatives).

7 Results

7.1 Random Assignment

Since no other group has set out to solve this problem, I had no state-of-the-art baseline to measure against. To get a sense of how good or bad a classifier is at predicting labels, I compared it to a random, balanced assignment of a positive or negative labels to the testing dataset. This random assignment resulted in the following performance:

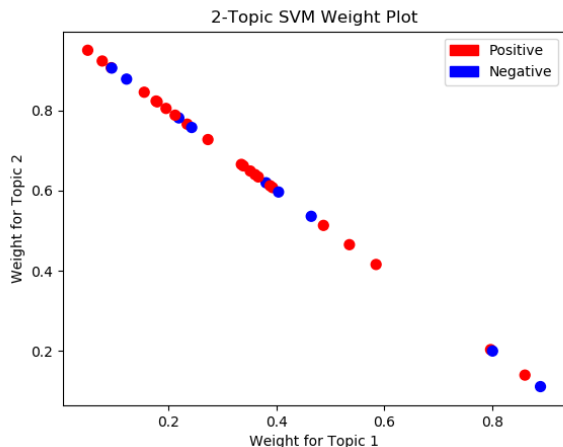
Precision	30.1%
Accuracy	54.2%
Recall	46.6%
F1 Score	0.366

Table 1: Random Assignment Performance

Clearly this method is lacking, but it provides an adequate baseline against which to judge the next classifier, the Support Vector Machine.

7.2 Support Vector Machine

The SVM, from sklearn, that I tested gave lackluster results. While SVMs can be very powerful for some problems, the LDA model output was not well suited to an SVM classifier. Refer to Fig. 1 for a scatter plot of the weights of documents with two topics plotted in two dimensions.

**Fig. 1: Scatter Plot of LDA Weights for k=2**

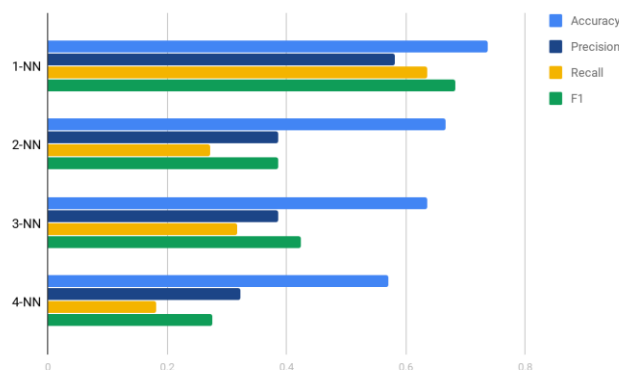
Since LDA assumes that documents are composed probabilistically from core topics, it assigns weights to the topics that it derives from the document such that the weights add up to equal one. This constraint on weight values results in all weight values in the testing set lying on a (n-1) dimensional hyperplane, where n is the number of topics. A linear SVM, of course, seeks to find a (n-1) dimensional hyperplane that separates all of the points into separate groups. This is obviously impossible in this situation. While an SVM with a polynomial kernel of degree 4 did yield some results, this classifier could do little better than random assignment.

Precision	56.3%
Accuracy	35.5%
Recall	40.9%
F1 Score	0.474

Table 2: SVM Performance

7.3 K Nearest Neighbors

The next classifier I tested was K Nearest Neighbors. I found sklearn's KNN to be a significant improvement on both SVM and random assignment, although the results are highly dependent on the number of neighbors that are taken into account for the classification. Figure 2 shows the respective results of KNN prediction with different values of K.

**Fig 2: KNN Results with Varying K**

It became clear very quickly that the most effective KNN classifier used only the single nearest neighbor to predict a new test document. This can be explained by the size of the dataset. Since the total number of documents is only 117, and just 84 documents are used for the prediction of each new test document, the move from one neighbor to two neighbor represents a significant increase in the portion of the data set that is influencing the prediction. As the proportion of the dataset used to classify a point grows, the amount of noise in the prediction process can grow as well.

The scatter plot of points in Fig 1 also give insight into this problem: positive and negative documents are not forming large groups. Since the groupings are small, with some negatives grouped with only one or two other negatives, an increase in the number of neighbors influencing the prediction can quickly include members from outside a local group. Still, KNN using a single neighbor yielded a significant improvement over SVM and random assignment, as shown below.

Precision	73.7%
Accuracy	58.1%
Recall	63.6%
F1 Score	0.683

Table 3: KNN (K=1) Performance

7.4 Decision Tree

After KNN, I moved on to a decision tree model. I used sklearn, a python library, to generate a CART decision tree using Gini as the uncertainty measure. The resulting decision

tree, after being fit to the training data had over 130 nodes. The results were very encouraging, in table 3.

Precision	88.2%
Accuracy	70.9%
Recall	68.2%
F1 Score	0.769

Table 4: CART Decision Tree Performance

This decision tree was the best performing single classifier, and it was only outdone by the following classifier, the random forest.

7.5 Random Forest

"If one is good, more must be better," as the saying goes, and it can be true to a certain extent. I used sklearn's random forest in this step. The optimal number of estimators for the random forest was seven. When I aggregated seven decision trees into one classifier, I saw yet another jump in performance.

Precision	83.3%
Accuracy	80.6%
Recall	90.9%
F1 Score	0.869

Table 5: Random Forest Performance

One of the goals from the beginning was to match or beat the unpublished results of my graduate adviser, David Molik. His reported accuracy in classifying the testing dataset was 77%. With this random forest I was able to surpass random assignment by a large margin and slightly beat the best unpublished results on which I have information.

7.6 Feed Forward Neural Network

The last classification algorithm used in this project was a keras implementation of a feed forward neural network. The neural network created consisted of three layers and all layers had non-linear activation functions to determine the unknown relationship between the topics and classification label. The results of the neural network were not ideal for several reasons based on the model accuracy, lost, and training problems.

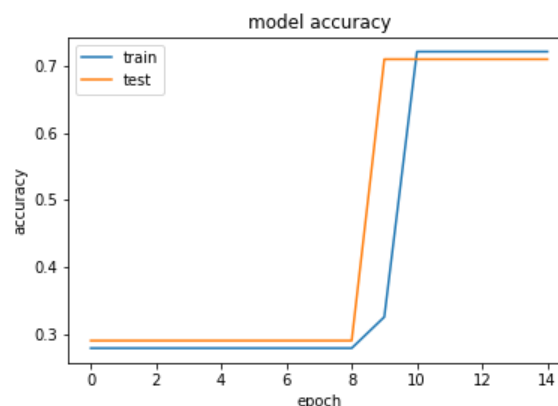


Figure 3: Neural Network Model Accuracy

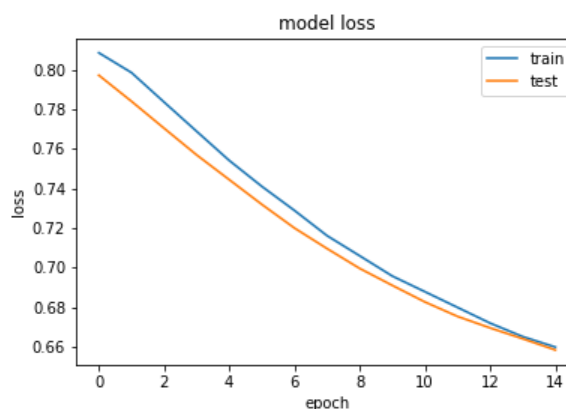


Figure 4: Neural Network Model Loss

The model accuracy was a major indicator that there is a need for readjustment in the training parameters. For both the training and testing sets, the model has a stagnant accuracy which immediately rises to a new plateau level. The model loss converges for both the training and testing sets, but further analysis revealed this result is misleading. As the model begins training, it classifies every paper as negative, and then after a certain number of epochs, it switches to classifying every paper as positive. This would account for the massive and sudden switch in accuracy, because the positive-negative ratio was a 70-30 split.

Neural networks have the potential to become the best classifier in this experiment, but there are several aspects preventing it from achieving this status. Firstly, the sheer amount of data points and training parameters is low for the network to correct find the relationship. Secondly, a higher degree of knowledge of neural network architecture, nodes, and layer formation is necessary as well to unlock the true potential of neural networks.

Precision	71.0%
Accuracy	70.9%
Recall	100%
F1 Score	0.83

Table 6: Neural Network Performance

<https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-10-177>

[2] - Predicting Response to Political Blog Posts with Topic Models, <https://www.aclweb.org/anthology/N09-1054.pdf>

8 Conclusion and Future Work

The experimental methods in this paper has led to the creation of several models that display optimal results for this project's purposes. The LDA-Random Forest classifier combination provided remarkable results in accurately predicting which journal article was a positively labeled document. However, there are several recommendations and future improvements for this project.

The most pressing improvement is to expand the dataset. Currently, we have far more negative papers (34) than positive result papers (83). Although this project factored in the imbalance of positive and negative papers, the ideal would be to equalize the positive-negative ratio and further increase the total amount of papers for classification. The increase of papers will increase the total amount of possible words and associations, which would benefit classifiers such as the neural network immensely with richer data values. This improvement will require significant work in the future, as DNA barcoding analysis can be time consuming, but is certainly one of the biggest possible value adds to this sort of project. Another goal is the creation of a co-word map. This is one of the new areas of exploration with LDA, and it involves determining relationships between documents through the co-occurrences and co-absences of words in each document. The end product from this method would be a visual map that shows the relationship connection strength between various words and phrases in a network-like map. This has been performed on other biological papers in regards to authorship and citations, but has clear implications in this research as well.

The last method to try in the future would be a reverse recall of this project. Currently, the models are tailored towards classifying papers that directly mention *C. Neoformans*. A reversal recall model would be trained to classify papers that do not directly mention *C. Neoformans*. This could reveal another relationship between the positive and negative papers while also testing the robustness of the current model as well.

This project represented the combination of two sciences: biology and data. As the future of biology moves towards advance computation and data collection, more projects such as ours will be necessary to aid in the progression and merger of these two areas.

9 References

[1] - Text-mining of PubMed abstracts by natural language processing...,