**DATA ANALYTICS FOR IMPROVING PATHOLOGY TEST USE**

Monaesh Siva Kumar

Curtin University

Final Report for NPSC2001 Research, Leadership and Entrepreneurship in Science 1

Table of Contents

[Abstract 4](#_Toc57899952)

[DATA ANALYTICS FOR IMPROVING PATHOLOGY TEST USE 5](#_Toc57899953)

[Problem Statement and Background 5](#_Toc57899954)

[Objectives 6](#_Toc57899955)

[Project Objectives 6](#_Toc57899956)

[Personal Objectives 6](#_Toc57899957)

[Review of Literature 7](#_Toc57899958)

[Process and Methodology 8](#_Toc57899959)

[Attempt 1: Non-supervised clustering 8](#_Toc57899960)

[Method: 8](#_Toc57899961)

[Results: 8](#_Toc57899962)

[Lessons Learnt: 9](#_Toc57899963)

[Attempt 2: Manual Grading System 9](#_Toc57899964)

[Method: 9](#_Toc57899965)

[Results: 10](#_Toc57899966)

[Lessons Learnt: 10](#_Toc57899967)

[Attempt 3: Dictionary Model 11](#_Toc57899968)

[Method: 11](#_Toc57899969)

[Results: 11](#_Toc57899970)

[Results 12](#_Toc57899971)

[Recommendations & Discussion 13](#_Toc57899972)

[Personal Outcomes 14](#_Toc57899973)

[References 15](#_Toc57899974)

[Table. 1 16](#_Toc57899975)

[Table. 2 17](#_Toc57899976)

[Figure. 1 18](#_Toc57899977)

[R Code 19](#_Toc57899978)

Abstract

This project improves pathology test use by analysis of clinical history texts written by doctors. Several methods were explored, the best method was found to be an algorithm using a custom compiled dictionary. These were used to match terms and classify the texts within predefined categories. The algorithm was designed to be flexible and modifiable to any changes the user makes within the dictionary. The algorithm provides a matrix output that displays the input data into their respective categories. This allows pathologists to identify which doctors provide sufficient information.

**DATA ANALYTICS FOR IMPROVING PATHOLOGY TEST USE**

NPSC2001 is a unit designed for students of the Advanced Science course. It is intended to provide students with the opportunity to explore their chosen disciplines. Part of the unit is a yearlong project and that journey is encapsulated in this report. The chosen project was undertaken as one of the Curtin Data Science Pathology Partnership Projects 2020. This was undertaken with Susan Benson (Industry Supervisor), and Ritu Gupta (Data Science Academic Supervisor). The entire project was facilitated by Sonia Ferns (Unit Coordinator).

# Problem Statement and Background

Pathology is a study that specializes in the cause and nature of diseases. It helps medical professionals to prevent, diagnose, manage, and monitor many conditions through pathology tests. In Australia, pathology tests usage contributes significantly to the rising costs of healthcare (Zhuang et al., 2013). When medical practitioners order a pathology test, information about the patient should be provided with the request. This information assists the laboratory staff in ensuring the correct tests are ordered and the results are reported and interpreted appropriately. Problems can occur when doctors fail to report sufficient information to the laboratory which can result in additional tests being performed. This leads to a delay in the result and the overall costs of running the test causing a burden on the health care system and funding. This raises an issue as doctors often may not provide the required information. The problem is identifying which doctors consistently provide insufficient information for the pathologists to work off. By identifying these doctors, we can then request for more information to be provided in future pathology requests.

# Objectives

## Project Objectives1

The main outcome of the project is to develop a performance indicator to quantify the quality of information provided by doctors. Text extracted from de-identified pathology requests is analyzed. The aim is to help the pathologist identify which practitioners are providing sufficient information. The main scope of this project the analysis of the following datasets:

* Blood Culture
* Histology
* Wound Swab

## Personal Objectives2

Industry projects and experience can help to expand one’s skillset in more ways than traditional university can. My main goals were to gain experience and confidence in working within the industry. Improving my ability to effectively work and communicate with industry partners outside my domain is a target I intend to focus on. Furthermore, doing projects that have real world application is something I am interested in but have not explored much in University. Understanding the process involved in the steps it takes from inception of the idea to the deployment of the project is something I would like to be able to experience firsthand.

# Review of Literature

Measuring information in medical texts is a challenging topic as it can have multiple solutions. Due to the nature of the topic, it can differ wildly based on the context of the situation and furthermore, the type of performance indicator that is required also differs based on the requirements of the client.

In “A method of measuring information in language, applied to medical texts”, a similar problem is explored. The method they chose to explore used complexity scores. The assumption made was that a higher complexity of words correlates to a higher information density which could be a valid approach to this problem. However, they used a static complexity score which does not represent differences in context of each problem. Different datasets would have different definitions of useful information. My approach would have to adapt a variation of this.

The chosen language for development of the algorithm is R. Due to the rich ecosystem of R packages, many text analysis tasks can be simplified. One such package was developed by Benoit et al is *quanteda* (Quantitative Analysis of Textual Data). This is a package is used to analyse, create and process text within R. The nature of working with textual data is complex as it contains large amounts of memory and requires longer computing times. By working through developed packages, methods have already been implemented to simplify time consuming tasks (Benoit et al., 2018).

Overall, there is not much current research done on the topic and there are numerous gaps in the existing literature. While research has been done on quantifying information and analysing medical texts separately, it has not been practised or tested. This is due to the volume of medical words that exist. Furthermore, classifying and measuring their value is dependent on the context of the sentence. Hence my project aims to target this gap in the existing literature.

# Process and Methodology

There were several approaches attempted. These methods produced a variation of semi-successful results. The results for the first two attempts will be discussed below as contextual information is needed to understand the issues and facilitate the improvement necessary in the following iterations. These attempts are shown over the next few pages.

## Method 1: Non-Supervised Clustering1

Method:

The Histology dataset was cleaned. Numbers and punctuation were removed, and words were stemmed. Which means longer words are converted to their shorter form. E.g., the word ‘clinical’ is stemmed to ‘clinic’. This helps reduce the number of required words to compute.

This cleaned dataset was then converted into a document term frequency matrix. This is a matrix that measures the frequency of terms that occur in a collection of texts. In the matrix, rows correspond to each text in the collection and columns correspond to terms within the texts (Miner & Elder, 2012). The frequency of each word in each document was recorded in this. This was then used in the K-means clustering algorithm (Zwitch, 2013).

Results:

This appeared to be an unsuccessful attempt purely due to the nature of K-means clustering. The algorithm had some successes, but it provides clustering based on words and not actually value of the words to the pathologist. Furthermore, due to size of the document term matrix, the method is not scalable to the blood culture data as it is a much larger dataset.

Takeaway:

Non-supervised clustering does not work. The computer cannot quantify usefulness of the text without professional supervision. For next attempt manual grading will be necessary in order to quantify text usefulness.

## Method 2: Manual Grading System2

Method:

The blood culture data was used this time. The dataset was not cleaned. Instead, a new column called ranking was added to the dataset. Ranking represents the usefulness of information, going from 1 being least useful and 5 being most useful. The idea is the algorithm can be run on a clinical history and decide how useful it will be to the pathologist receiving it on a scale of 1 to 5. The ranks were decided with collaboration from Industry experience.

The ranking and their compositions are shown below:

1. **Rank 1:** Any clinical history text with only punctuation, no value entered, or less than two characters.
2. **Rank 2:** Any reference to external information was considered. Any text with words, ‘see’, ‘form’, ‘notes’, ‘refer’, ‘clinical’,’ request’, ‘scanned’.
3. **Rank 3 to 5**: Rest of information is ranked using a regression model based on the data inputted by the pathologist

The issue with the previous model was lack of human expertise. To amend that issue this algorithm requires manual grading.

2% of the input data set is selected at random and saved to a csv file with a blank Ranking column. This column is intended for the pathologist using the algorithm to enter a number from 1 to 10 based on how useful that particular text seemed to be. For example, ‘Fever Urine MCS no growth Was o Cefuroxime for 4 days stopped 5 December Has had One dose of Trimethoprim’ can be ranked as 9/10 while just ‘sepsis’ is ranked as a 1/10.

The user ranked csv file is read in after professional input. Numeric features are then produced for each row. E.g., Word count, number of syllables, Character count etc.

This is then used to create a model. An example formula that we can expect is:

‘Rank = 0.5\* wordCount + 0.3\* sentenceLength + 0.6\* SyllableCount’

This is determined by forward selection of variables. Due to the use of a data driven modeling technique, the algorithm is adaptable for many kinds of data sets. By applying the same feature extraction on the remainder of the dataset that was unranked and applying the same formula to it we can get an estimation of rank from the pathologist. This is then scaled into a range of 1 to 5 and applied to the dataset.

Results:

The end result of this attempt was promising and did somewhat mimic a performance metric for clinical history texts. However, this may not be sufficient as it doesn’t take into consideration the usefulness of terms and only their frequency in the text. This does provide a valid approach to classification and can be expanded upon with custom useful words data suited more towards this dataset. By using a more custom useful word dataset it can greatly improve prediction of the algorithm.

Takeaway:

The next approach requires a custom data base of words in order to produce a valuable output. Without a definition, the model is quite vague.

## Method 3: Dictionary Model3

Method:

A more customized approach had to be taken for valuable results to be extracted. The first step was to build a customized dictionary of terms. The terms were collected and separated into groups based on the pathologists needs. Using the custom dictionary, the words were matched within the text. If a certain word existed in the text, it was classified among that category. E.g., ‘Sepsis’ when matched in the text is classified under ‘Fever and otherwise unwell’ category.

To deal with overlapping categories, a matrix like output format is adopted. This means that a document in the collection can belong to more than one category. While the idea behind the method is much simpler than previous. Matching and handling 400k+ documents each with a varying amount words within it is quite computationally intensive. This is where *quanteda* was used. By converting the Dictionary to a *quanteda* dictionary, the tokens lookup function could then be used. This cuts the compute time to a fraction of what it would otherwise be due to the faster methods used by the package.

Results:

The output is a matrix. Rows represent number of documents, and columns represent categories. By providing this matrix format of output this allows the pathologist to explore the data output and sort the text based on category. While the result is not what was initially set out to do, this still provides the pathologist an enhanced method to separate the texts.

# Final Results

The resulting output of the algorithm is a category matrix. Rows represent each document from the inputted dataset, each column represents each category in the input dictionary.

Based on the trial dictionary that was compiled, the algorithm had a 93.3% classification rate. This means that only 6.7% of the data points could not be grouped into any of the categories. When only non-blank texts are considered, we observe a classification rate of 87.9%. This is quite good as it means a majority of the texts shared these common terms. This is means that the idea of matching terms is a valid approach to classify the data. The algorithm was designed in a way that allowed the dictionary to be flexible. This means that the classification rate can only get better as more words are added to its lexicon. A graph of the categories and their respective counts Is shown in *figure. 1*. However, none of the numbers are set in stone. The dictionary is something that can be updated when necessary. It can also be completely remade for a new data set if necessary. This allows its applications to be flexible. An example of a possible dictionary is shown in Table 1.

The resulting dictionary would then match words based on what is specified by the user. An example of this is shown in *figure. 2*. Keep in mind the algorithm itself does not have any hardcoded categories or words. Examples used are just some of the testing categories that were used.

Examples of the possible cases are presented:

1. ‘Transplant patient’ would be put into immunosuppression group as ‘transplant’ is a matched word.
2. ‘SOB, return Mexico’ would be put into respiratory as ‘SOB’ is a matched word, it would also be put into travel related as ‘return’ is a matched word.
3. ‘Nil’ would be put into No clinical details as it is a matched word.
4. ‘return from abroad, temp Nil’ would be put into travel-related but not into No clinical details. This is because the term ‘Nil’ has the condition to be an exact match.

# Recommendations & Discussion

Overall, the goal of the project seems to have been achieved. The final algorithm addresses the issues faced by the pathologist. However, another goal of the project was to quantify information within the text. Since this was not achieved, a future version could incorporate Attempt 2 and 3 together. Using the expert database that were compiled, words could be matched and used as regressors to a linear model. This provides a more reasonable choice of regressors compared to the standard text features used in attempt 2. This method could provide a numeric estimation for ‘usefulness’ of the clinical history text.

A recommendation of usage of the algorithm is to have several dictionaries made for different datasets. This will allow the user to easily use the same algorithm for different databases. Since the final algorithm was only tested and compiled with the Blood Culture dataset

While the output matrix of the algorithm can be explored easily with excel, a more elegant solution could be to use R Shiny for the data output and input. Not only will this make the entire algorithm integrated, it will appear more polished during usage. However, these are merely superficial changes to make the algorithm more appealing for pathologists to use. Therefore, they are not necessary at this stage.

This project has the potential to be explored further. While most of the groundwork has already been set, the overall feel of the program is rough. There is plenty of scope for a more polished design in possible future attempts.

# Personal Outcomes

The structure of the project was mostly self-guided. Any decisions that I had to take in the creation of the algorithm, were done with minimal assistance, with the exception of a few technical discussions with my Academic Supervisor. Furthermore, my industry supervisor was not from a data science background, which meant I had to formulate my own solutions. This was hugely beneficial for me as it was much more challenging and allowed for a more enriched learning experience. While I initially found it quite challenging, I felt as though I adapted quite well to the challenges and issues that arised. When approached with a problem, I had to determine a solution in most cases without knowing what I was looking for. This type of project is something I felt helped my confidence and expertise in handling industry projects. This experience helped my project management skills. However, there were some areas I felt I could still improve upon. Communication skills being my main drawback. While communication with someone within your domain is a simple task, relaying information about methods and processes you have used with someone who has no expertise in the area is challenging. This is something I feel I could benefit from expanding upon.

Overall, I am pleased with the outcome of the project. However, due to the sheer number of exploratory models and trials required, I did not spend as much time as I would like on the final method.

References

A gentle introduction to cluster analysis using R. (2015, July 22). *Eight to Late*. <https://eight2late.wordpress.com/2015/07/22/a-gentle-introduction-to-cluster-analysis-using-r/>

Australia, H. (2020, August 12). *Understanding pathology tests* [Text/html]. Healthdirect Australia. <https://www.healthdirect.gov.au/understanding-pathology-tests>

Benoit, K., Watanabe, K., Wang, H., Nulty, P., Obeng, A., Müller, S., & Matsuo, A. (2018). quanteda: An R package for the quantitative analysis of textual data. *Journal of Open Source Software*, *3*(30), 774. <https://doi.org/10.21105/joss.00774>

Dotson, M. (n.d.). *Introduction to Text Analysis in R*. DataCamp. Retrieved October 12, 2020, from <https://www.datacamp.com/courses/introduction-to-text-analysis-in-r?embedded=true>

*Economic value of pathology\_June 2019 APPROVED.pdf*. (n.d.). Retrieved December 1, 2020, from <https://www.pathologytechnology.org.au/resources/Documents/Economic%20value%20of%20pathology_June%202019%20APPROVED.pdf>

Gordon, D. B., & Sager, N. (1985). A method of measuring information in language, applied to medical texts. *Information Processing & Management*, *21*(4), 269–289. <https://doi.org/10.1016/0306-4573(85)90053-6>

Jensen, K., Soguero-Ruiz, C., Oyvind Mikalsen, K., Lindsetmo, R.-O., Kouskoumvekaki, I., Girolami, M., Olav Skrovseth, S., & Augestad, K. M. (2017). Analysis of free text in electronic health records for identification of cancer patient trajectories. *Scientific Reports*, *7*(1), 46226. <https://doi.org/10.1038/srep46226>

Li, Y. H., & Jain, A. K. (1998). Classification of Text Documents. *The Computer Journal*, *41*(8), 537–546. <https://doi.org/10.1093/comjnl/41.8.537>

Miner, G., & Elder, J., IV. (2012). *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. Elsevier Science & Technology Books. <http://sbiproxy.uqac.ca/login?url=http://international.scholarvox.com/book/88812229>

Napolitano, G., Marshall, A., Hamilton, P., & Gavin, A. T. (2016). Machine learning classification of surgical pathology reports and chunk recognition for information extraction noise reduction. *Artificial Intelligence in Medicine*, *70*, 77–83. <https://doi.org/10.1016/j.artmed.2016.06.001>

SAP Conversational AI. (2018, July 17). *Text Clustering with R: An Introduction for Data Scientists*. Medium. <https://medium.com/@SAPCAI/text-clustering-with-r-an-introduction-for-data-scientists-c406e7454e76>

*Summarizing a clinical history for pathology sign-out*. (n.d.). Department of Pathology. Retrieved December 1, 2020, from <https://med.uottawa.ca/pathology/summarizing-clinical-history-pathology-sign-out>

Zhuang, Z. Y., Wilkin, C. L., & Ceglowski, A. (2013). A framework for an intelligent decision support system: A case in pathology test ordering. *Decision Support Systems*, *55*(2), 476–487. <https://doi.org/10.1016/j.dss.2012.10.006>

Zwitch, R. (2013, September 17). Clustering Search Keywords Using K-Means Clustering. *R-Bloggers*. <https://www.r-bloggers.com/2013/09/clustering-search-keywords-using-k-means-clustering/>

Table. 1

Example of dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| Terms | Contains or Match | Group | Category |
| Nil | match | No clinical details | uninformative |
| transplant | contains | immunosuppression | informative |
| SOB | match | respiratory | informative |
| chemo | contains | immunosuppression | informative |
| return | contains | Travel-related | informative |
| dysuria | contains | urogenital | informative |

Note: *This is just an example and does not represent an actual output.*

Table. 2

Example of Algorithm output

|  |  |  |  |
| --- | --- | --- | --- |
| ClinHx | immunosuppression | Travel related | No Clinical Details |
| ‘Transplant patient’ | 1 | 0 | 0 |
| ‘cancer, return’ | 1 | 1 | 0 |
| ‘Nil’ | 0 | 0 | 1 |
| ‘transplant temp Nil’ | 1 | 0 | 0 |

Note: *This is just an example and does not represent an actual output.*

**Chart

Description automatically generated**Figure. 1

**Figure .1 –** Bar Graph Showing the amount of document in each category from the algorithm output

# R Code

Note: *R version 4.0.2 (2020-06-22).*

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101 | ```{r}  library(readr)  library(quanteda)  library(tm)  ```  ```{r}  BloodCult<- read\_csv("BloodCultureClinHxSpeciDesc2015-202012.csv")  CleanedClinHx <- trimws(removeNumbers(removePunctuation(BloodCult$ClinHx)))  ```  ```{r}  ClinHxDictionary <- read\_csv("ClinHxDictionary.csv")  ClinHxDictionary$Group <- tolower(trimws(ClinHxDictionary$Group))  ClinHxDictionary$ContainsorMatch <- tolower((trimws(ClinHxDictionary$ContainsorMatch)))  NumGroups <- as.integer(length(unique(tolower(trimws(ClinHxDictionary$Group)))))  Groups <- unique(ClinHxDictionary$Group)  Dict\_Mat <- matrix(nrow = nrow(BloodCult), ncol = NumGroups)  ```  ```{r}  # Contains Dictionary  Dic\_List <- list()  for (i in 1:NumGroups){  temp <- (ClinHxDictionary[which(ClinHxDictionary$Group == Groups[i] & ClinHxDictionary$ContainsorMatch == 'contains'),]$Term)    #Adding \* for GLOB Style matching  temp <- lapply(temp, function(x) str\_c(x,'\*'))  Dic\_List[[length(Dic\_List)+1]] <- temp  }  names(Dic\_List) <- Groups  Dict <- quanteda::dictionary(Dic\_List)  ```  ```{r}  BloodCultTokens <- tokens(CleanedClinHx)  ```  ```{r}  Dict\_lookup <- tokens\_lookup(BloodCultTokens, dictionary = Dict, case\_insensitive = TRUE, valuetype = 'glob')  mat\_lookup <- as.matrix(dfm(Dict\_lookup))  colnames(mat\_lookup) <- names(Dict)  ```  ```{r}  colSums(mat\_lookup)  ```  ```{r}  # Exact Match Dictionary  Dic\_List\_Match <- list()  for (i in 1:NumGroups){  temp <- (ClinHxDictionary[which(ClinHxDictionary$Group == Groups[i] & ClinHxDictionary$ContainsorMatch == 'match'),]$Term)    #Adding \* for REGEX Style matching  temp <- lapply(temp, function(x) str\_c('^',x,'$'))  Dic\_List\_Match[[length(Dic\_List\_Match)+1]] <- temp  }  names(Dic\_List\_Match) <- Groups  Dict\_Match <- quanteda::dictionary(Dic\_List\_Match)  ```  ```{r}  #Since Token Seperation removes the sentence structure new tokens must be constructed  BloodCultTokens <-as.tokens(as.list(CleanedClinHx))  ```  ```{r}  Dict\_Match\_lookup <- tokens\_lookup(BloodCultTokens, dictionary = Dict\_Match, case\_insensitive = TRUE, valuetype = 'regex')  mat\_lookup\_match <- as.matrix(dfm(Dict\_Match\_lookup))  colnames(mat\_lookup\_match) <- names(Dict)  ```  ```{r}  mat\_lookup <- mat\_lookup+mat\_lookup\_match  (mat\_lookup)  ```  ```{r}  #NONE PROVIDED  None\_Provided <- rep(0, nrow(BloodCult))  None\_Provided[which(is.na(CleanedClinHx) & rowSums(mat\_lookup) == 0)] <- 1  #PUNCTUATIONS/SingleCharacter  SingleCharacter.or.Punctuation <- rep(0, nrow(BloodCult))  SingleCharacter.or.Punctuation[which(!grepl("\\w", CleanedClinHx) & !is.na(CleanedClinHx )& rowSums(mat\_lookup) == 0)] <- 1\SingleCharacter.or.Punctuation[which(nchar(gsub(' ' ,'', CleanedClinHx)) == 1 & rowSums(mat\_lookup) == 0)] <- 1  Final\_Mat <- cbind(SingleCharacter.or.Punctuation, mat\_lookup)  Final\_Mat <- cbind(None\_Provided, Final\_Mat)  ```  ```{r}  write.csv(Final\_Mat, 'output.csv')  ``` |