University of Sheffield

Analysing Evidence Quality In Medical Research Papers

Nitheesh Dharmapalan

Supervisor: Dr.Mark Stevenson

This report is submitted in partial fulfilment of the requirement for the degree of Bsc Artificial Intelligence and Computer Science by Nitheesh Dharmapalan

in the

Department of Computer Science

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Nitheesh Dharmapalan

Signature: Nitheesh

Date: 11/05/2022

Abstract

Systematic reviews analyse evidence quality for specific clinical topics. As a result, these reviews have become extremely significant when it comes to making evidence based decisions about healthcare. Due to the boom in the expansion of biomedical research papers, these become very difficult to produce. This project aims to mitigate this issue by proposing a technique to automate the risk of bias assessment portion of the reviews as this is an important part of the reviews and also time-consuming to produce. This will be achieved by using distant supervision learning by using data from the Cochrane Database of Systematic Reviews as a pseudo corpus.

Acknowledgements

I would like to thank my supervisor Dr Mark Stevenson for his help and guidance. I would also like to thank my friends and family for their support.

Contents

List of Figures

Chapter 1

Introduction

Randomised controlled trials(RCT) form the majority of literature in evidence based medicine because they are one of the most effective ways of assessing treatment effects and interventions. Systematic reviews, which aim to summarize the evidence of randomised controlled trials in an unbiased way, are thought to be the strongest form of evidence in the medical field [1]. Assessing risk of bias in randomised control trials is one of the most important tasks of these reviews. This is because introduction of bias can result in over or under estimating the results of treatments [1]. For example, inadequate allocation concealment,which is a process to hide the allocation sequence from researchers when they assign intervention groups, can exaggerate the effect of medical intervention by 40% [2]. Risk of bias is the likelihood that features of a study design will give misleading results. Due to the exponential growth of of evidence based literature, conducting risk of bias analysis becomes very challenging and therefore warrants automation.

1.1 Aims and Objectives

The main aim of the project is to create a system that automates risk of bias analysis and to test the performance of such a system. The idea is to make use of distant supervision, which involves using an already existing collection of data to create labelled data for training. In this case, the Cochrane database of systematic reviews(CDSR) will be used. This contains several systematic reviews which can be used to extract risk of bias information about clinical trials. The machine learning technique to be used is called support vector machine(SVM), which is a supervised learning algorithm that can be used for classification. Different data representation techniques will be used and compared against each other. These techniques include bag of words and bigrams. Bigrams are a two word sequence of features that give context about the surroundings of the words. There will be six different risk of bias factors considered (see Figure 1.1).

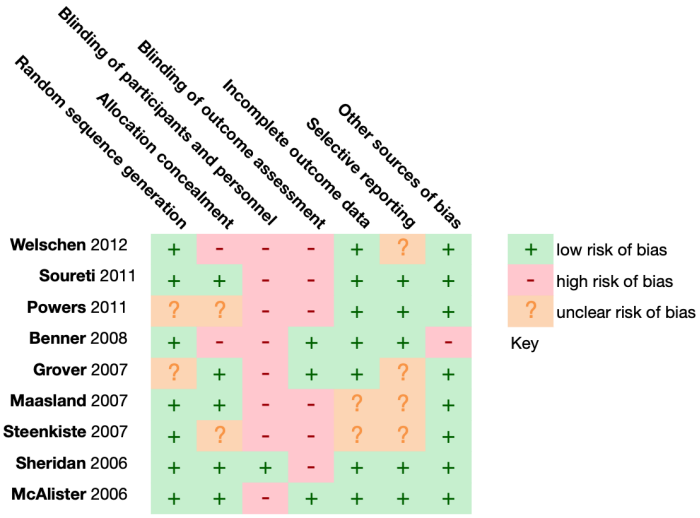

Figure 1.1: Risk of bias output from Cochrane reviews
Risk of bias output from Cochrane reviews. The risk factor labelled 'Other sources of bias' will be ignored because it is too broad and changes for almost every clinical topic [1]

1.2 Relevance to Degree

This project has significant relevance to the degree. The main module it overlaps with is Text Processing, in which different preprocessing techniques and data representation, such as, bag of words are taught. In addition, text classification algorithms, including Naive Bayes, were taught as a part of this module. Learning this module while doing this project proved very usefull as they complemented each other well. Moreover, there is also some relevance to the Adaptive Intelligence module, in which more advanced classification techiques are taught, such as, using neural networks.

1.3 Overview of the Report

The report will contain a literature review in Chapter 2, which will contain relevant research pertaining to the area. It will then be followed by requirements and analysis, which contains the main objectives of the project in chapter 3. Chapter 4 will contain detailed information about the methodologies, including the design, implementation and the changes made from the analysis section. Chapter 5 will contain the results and performance metrics of the classifiers. Finally, Chapter 6 will contain the concluding remarks.

Chapter 2

Literature Survey

This chapter is split into three sections. The first consists of an overview and description of risk of bias criteria. The second section contains a description of methodologies of relevant work and their effectiveness. The thrid section will explain how SVM works.

2.1 Description Of Risk Of Bias

There are six risk of bias criteria that are commonly considered in risk of bias analysis which will be used to create the classifiers. These are outlined in detail in the Cochrane handbook [3]. One of the types of bias analysed is called random sequence generation in which the authors are checking if an appropriate method of randomisation was used. There would be a low risk of bias for this domain if the authors of the trial describe a random aspect in the process of sequence generation. This could include referring to a random number table, using a computer random number, shuffling cards and throwing cards. If there is non random component in the sequence generation process, such as sequence generation by odd or even date of birth, hospital record and preference of the participant. If this risk of bias is high, then the trial is not randomised and other biological factors like age can affect the result. This can result in over or under estimation of treatment effect. Usually, the effect of the treatment is exaggerated by this risk of bias [4].Another risk of bias that is being investigated includes allocation concealment, which determines whether researchers are able to influence which group participants are allocated to. There is low risk of bias for this if the methods were used to conceal the allocation of the participants so that neither the participants nor the investigators know which group they are in. This could include using sequentially opaque sealed envelopes or pharmacy controlled randomisation, in which pharmacists split up participants randomly [5]. There would be a high risk of bias for this if participants or investigators could know what groups the participants were going to be in. This could happen if unsealed envelopes were used. Furthermore, blinding of participants and personnel is also an important risk of bias to consider. The treatment group the participants were in must be concealed from them and the investigators. If blinding is not carried out, then participants might be more heightened to the effects of the treatment to be studied.

Studies that were not blinded exaggerated the treatment effects by 17% [6]. Additionally, blinding of outcome assessment is also a risk of bias to consider. The person assessing the outcomes must be blinded to the intervention group of the participants. There is low risk of bias for patient reported outcomes if the participants were blinded to the treatment. If they were not, then their own personal bias of the treatment can come into play. If the outcomes are reported by care providers, then there is low risk of bias. If forms are being used, then there is low risk of bias if the treatment or the effects of the treatment could not be noticed in the data in the form. Having a high risk of this once again exaggerated the treatment effect by about 13 % [7]. Incomplete outcome data is also a risk of bias criteria. It is used to check if there is no missing outcome data. If participants withdraw from the trial, then there is a low risk of bias for incomplete outcome data if there is no bias in the result due to the withdrawals. The data for these trials must be handled properly in order to avoid this risk and there is significant room for this in the literature [8]. Selective reporting bias is the final risk of bias to consider. This checks whether any health outcome has not been published [4] [9]. There is a low risk of bias for this if the study protocol is published. The study protocol contains the objectives, methodology and the organisation of the research being carried out. This can be checked against the results to confirm no data is missing.

2.2 Relevant methods used and their Reliability

There has not been much research done in the area of information retrieval for medical reviews. However, the work done by Marshall et al [1] is particularly relevant to this project. In fact, this project is inspired by the work of the authors mentioned above. Marshall and colleagues used distant supervision to create a data set from the CDSR and trained an algorithm that automates risk of bias assessment using SVM. In addition, they also trained a joint model that predicts risk of bias of a clinical trial and provides reasoning for it by using quotes from the trial. The same techniques will be leveraged in this project to train a basic system that predicts risk of bias. Marshall et al only produced models using bag of words as their only data representation technique, neglecting the potential of improvement if other techniques are used. This project will produce the models with bag of words and bigrams. Another study by Millard et al [10] also automated risk of bias analysis using logistic regression, which is a classification algorithm that separates discrete classes. Three risk of bias criteria were used by them, which included sequence generation, allocation concealment and blinding. Two models were trained in this investigation. One was for predicting whether a sentence in a trial is contains relevant information and the other predicts the risk of bias level, which wad wither low or not low for each risk of bias. The performance metric used to test the results was AUC which represents the degree of separability of the classes. So, higher the AUC the better the classifier. An AUC score of greater 0.98 was achieved for the sentence extraction part and 0.72 for the risk level identification task. It is estimated in this study that 33% of articles can be assessed by one viewer whereas they would usually require two. Additionally, Marshall et al conducted [11] another study where they presented

a web application called RobotReviewer that conducts risk of bias analysis of a trial that has been uploaded to the app. This uses NLP and recurrent neural networks to work, which is a type of neural network that allows prior outputs to be used as inputs. There has been a review done assessing the reliability of this application by Gates et al [12], who used the chi-squared test to compare the reliability of the app with human data. The chi-squared test is a test that can be used to determine if there is a statistically significant different between actual and expected outcomes. This review found that the reliability of RobotReviewer is similar to human beings for most domains and better for allocation concealment, blinding of participants and personnel and overall risk of bias. Another study of RobotReviewer done by Soboczenki et al [13] found that using this app along with manual analysis resulted in faster times of risk of bias analysis. Reviewers in this study rated the app in between "good" and "excellent" on average. These reviews suggest that automating risk of bias analysis could be very useful. There have also been studies that use automated data extraction for gathering cancer related and genetic literature. A specific example of this would be Ling et al [14] [15]. In this study, the authors developed a method to generate gene summaries from literature using the idea of distant supervision. They leverage an existing corpus called FlyBase, which contains gene summaries, to create a training data set. In this project, distant supervision is being used directly for each domain of interest, whereas Ling et al had to infer text for each facet they were looking for [14]. Another study that has a similar use risk of bias analysis attempted to automate citation classification [16], which involves deciding whether a trial is relevant or not to the topic, for systematic reviews written on drugs. This study used a perceptron based algorithm to achieve results. It was found that using this classifier resulted in at least 50% less time for 11 out of 15 drug review topics. Finally, another study by Marafino et al [17] found that using SVM with n-grams can be useful in disease diagnosis in ICU patients. They found that using n-gram and SVM on ICU notes provided very high performance metrics for several diseases including jaundice.

2.3 Support Vector Machine explanation

SVM stands for support vector machine and is a supervised binary classification machine learning algorithm. Supervised learning involves creating a data set and adjusting the weights of the model as the input data is fed in until the weights are appropriately configured. The idea behind the SVM algorithm is to find a hyperplane in N dimensions that distinctly classifies the data points. A hyperplane is a subspace that has a dimension of one less than the space around it. So for N dimensions the hyperplane will have a dimension N-1. So for a two dimensional vector space, the hyperplane will be a one dimensional line. There could be many such hyperplanes that divides the data points but the objective of SVM is to find the one with that has the maximum margin. This means that the hyperplane must divide the classes such that there is maximum distance between the data points of both the classes. The margin can be maximised by using support vectors.

These are data points that are closest to the hyperplane and influence its position and

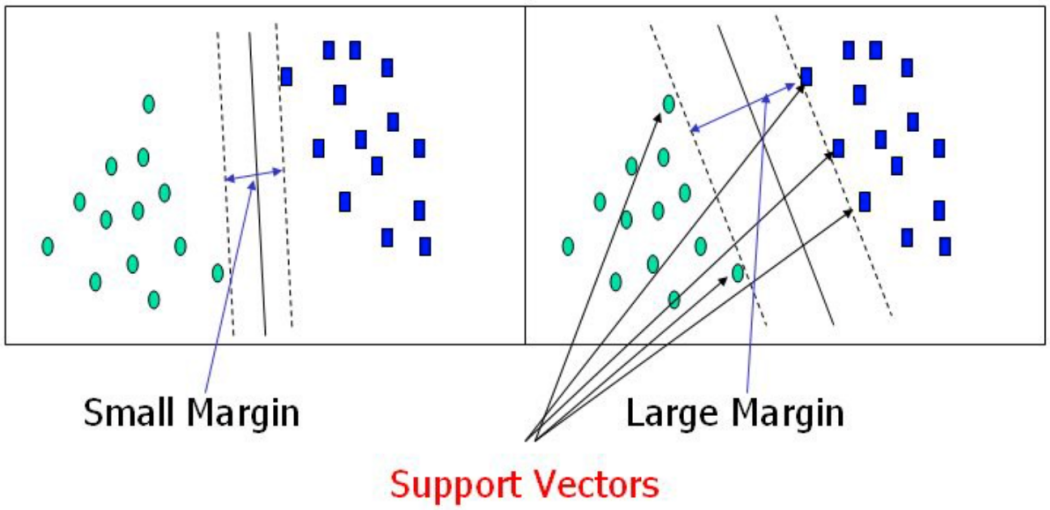

Figure 2.1: Support vector example

orientation the most (see figure 2.1). Deleting these vectors will change the position of the hyperplane and hence they are used to build the SVM classification system. The output is taken by passing inputs into this hyperplane function and depending on its value, which could be either 1 or -1, the classes are assigned. In order to maximise the margin between the data points and the hyperplane, a loss function is required. The loss function calculates the distance between the current output and the expected output of the algorithm. The loss function used is the hinge loss function. The hinge loss function will return zero if the predicted value and the actual value are the same. If not it will find calculate the distance between the predicted value and the actual value, that is, it calculates the loss. The function is given below:

$$c(x, y, f(x)) = 1 - y \times f(x)$$

Where c is the cost function, x is the input, f(x) is the predicted output and y is the actual output.

A regularisation parameter is then added to the loss function. This is done so that the margin maximisation and loss is balanced. The partial derivative is then taken to get the gradient function, which is then updated after each iteration. If the model correctly predicts the output, the gradient is only updated using the regularisation parameter. If there is misclassification, that is the predicted output is different from the actual output, then the loss function is also calculated when updating the gradient.

Chapter 3

Requirements And Analysis

3.1 Requirements

The main mandatory requirement is to build an SVM classifier that can perform risk of bias analysis and test this classifier using the precision, recall and F1 Score using bag of words. This can optionally be extended to use bigrams as an alternative data representation technique. To cover the basics, the metrics can only be measured for low risk of bias versus non low risk of bias but can optionally be extended to include all levels of risk.

3.2 Analysis

The program can be split into three main parts. The first part involves collecting data using distant supervision. The next involves training the model or potentially models and then finally evaluating the models.

3.2.1 Collecting data

The CDSR contains an abundance of semi structured data about systematic reviews in the form of XML. Usually these reviews contain information about several clinical trials. The goal is to extract specific information from these clinical trials, including, risk of bias assessment and the reference to the studies. In order to accomplish this, the XML file is first parsed. The risk of bias information for the six different metrics is then extracted for each study by accessing the relevant XML tags. The citation data for the studies are also extracted using the same method. This data includes the title of the trail and the authors' names, which is then used to form a search query that can be used to find the full PDF of the trial. The search query is then entered into Unpaywall, which is an online database that has access to over 30 million scholarly articles. These PDFs are converted to full text using Xpdf [1]. The data will be stored as a JSON file with the risk criteria mapped to the study name, the risk level and the path to the full text.

3.2.2 Training

For the basic model with SVM, the two classes to be classified are low risk level and not low risk level, which includes both high and unknown risk. Since, SVM is a binary classifier, only two classes can be used. So, the high and unknown risk are grouped into one class. Before the training is done, useful features that determine low risk of bias are extracted using bag of words. Bag of words involves counting the occurrences of the words that appear in low risk studies and using the most common ones. A stop list will be used to remove the most common words before extracting the features. Once these features have been extracted they are fed into the SVM training algorithm. Similarly, the same approach could be extended to create more classifiers such as high risk and non high risk and unclear risk and non unclear risk. The bag of words model can be replaced with a bigram using NLTK.

3.2.3 Evaluation

The models will be evaluated using precision, recall and F1 value. Each of these will be calculated for every risk of bias factor. There will be results that are true positive false positive, true negatives, false negatives, The results that fall into these categories are used to calculate precision, recall and F1 value. Precision is the number of true positive divided by the total predicted positives. Total predicted positives is given by the sum of true positives and false positives. The F1 value is the harmonic mean of precision and recall.

In general, higher recall means lower precision and vice versa. This is because a maximum recall value of one can be obtained if all the trials in the set are retrieved but this would make precision very low. Therefore, by using the harmonic mean, one can ensures that the final score isn't too high if one measure is improved at the expense of the other and uneven class distributions can be accounted for.

Chapter 4

Methodology

The development of the system can be broken up into three components. First the training data needs to be created using the CDSR. A model that predicts risk of bias will be trained using bag of words and another one with bigram data representation. Then the systems needs to be evaluated and compared. The data in the CDSR will be divided into two sets, a train set to train the data and a test set to check if the system is performing well when given unseen data. The coding will be done in python.

4.1 Design

The CDSR contains an abundance of semi structured data about systematic reviews in the form of XML. Usually these reviews contain information about several clinical trials. The goal is to extract specific information from these clinical trials, including, risk of bias assessment and the reference to the studies. In order to accomplish this, the XML file is first parsed. The risk of bias information for the six different metrics is then extracted for each study by accessing the relevant XML tags. The citation data for the studies are also extracted using the tags. Once the test and train data is collected this way, the classifier is trained and then tested. Since SVM is a linear classifier, a separate classifier is trained for each risk level, including low, unclear and high. This done for each of the six risk criteria. So, a total of 18 classifiers are trained and cross validated. The main program of the project, responsible for training and testing the model, will consist of six functions. This includes a function called convert_to_text_and_get_features, which will take the JSON data for testing or training, the risk criteria and risk level as parameters and then return a dataframe with all the filenames that fall under the risk criteria, the full text version of the PDF and the labels. The label would be 1 if the risk level matches the parameter or it would be zero otherwise. Then, a function, called make_representaiton is created to make a bag of word or bigram model. The type of model to be created should be passed as a parameter to the function and the full text of all the trials needed to make these models should also be passed. Then a training function is created to train each classifier for both bag of word and bigram models and also perform cross validation on the training data. It will take the type of representation to be

used, bigram or bag of words, as a parameter to decide which one to train the models with. Models will be trained for each possible configuration of risk level and criteria. Then, the testing function will test the classifiers on the test data for bag of words and bigram models The type of representation, bag of words or bigram, to be used needs to be passed in as a parameter. This function will also perform evaluation. The evaluation will also be done for each configuration of criteria and risk level. Evaluation will involve calculating precision, recall and F1 score using scikit learn. The test results will be printed out to the terminal and gathered to make tables and graphs displaying the results. There will be a main method that will call the test and training functions.

4.1.1 Evaluating the models

The models will be evaluated using precision, recall and F1 value. Each of these will be calculated for every risk of bias factor for every risk level. For example, for the low risk of bias level, there will be results that are true positive, consisting of the studies that are low risk and are labelled low risk by the system, false positive, which represent studies that have been labelled positive for low risk but are not low risk, true negatives, which represents studies that are not low risk and correctly labelled as high or medium risk and false negatives, which are studies that are labelled as high or medium risk when they are actually low risk. The results that fall into these categories are used to calculate precision, recall and F1 value. Precision is the number of true positive divided by the total predicted positives. Total predicted positives is given by the sum of true positives and false positives. Therefore, precision provides the proportion of studies that were labelled low risk are actually low risk. Recall is calculated by dividing the true positives by the total number of actual positives, that is, it provides the proportion of low risk studies that were actually retrieved by the system. The F1 value is the harmonic mean of precision and recall. It is given by:

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

In general, higher recall means lower precision and vice versa. This is because a maximum recall value of one can be obtained if all the trials in the set are retrieved but this would make precision very low. Therefore, by using the harmonic mean, it can be ensured that the final score isn't too high if one measure is improved at the expense of the other and uneven class distributions can be accounted for. The same logic of these metrics applies to unclear risk level and high risk level.

4.2 Justification for methods used

For the collection of the training and the testing data, distant supervision has to be used because a corpus of trials with risk of bias data is hard to find and very time consuming to make manually. For the data representation, a bag of word model is used because it is a standard model that is commonly used and a bigram is used because n-gram processing along

with SVM can result in high performance when it comes to automation tasks for example automating disease diagnosis [17]. Bigrams are chosen because they are generally superior to bag of words and are often tough to beat [18]. Sometimes using higher n-grams like trigrams can result in overfitting of data, meaning the model ends up to similar to the training data and cannot adapt easily to the test data. The SVM model is used because the paper that inspired this dissertation also used SVM [1]. Although, logistic regression produced good results, it was only experimented on three of the six criteria being used [10]. So, it is a better idea to use the SVM model. Cross validation is performed on the training set in order to avoid over-fitting and splitting the collected training data into a separate validation set. The training and testing data were stored in a JSON file because JSON is an efficient and easy to understand storage system.

4.3 Implementation

The implementation is split into four parts detailed below.

4.3.1 Creating the training and testing data from the Cochrane Database of Systematic Reviews

The short hand name of the study and the title of the study is extracted. The short hand name of the study is found in the NAME attribute (See 4.1) of the STUDY tag. It consists of the main author's last name followed by the year. The title is extracted from the TI tag (See 4.1). The title is used as a search query. This is then entered into Unpaywall, which is an online database that has access to over 30 million scholarly articles. Unpaywall has an API that can be used with python. This API is interfaced with python by the unpywall package. Once the query is sent through the API, a response is sent by the API containing the main author's last name. This is cross checked with the author's name contained in the short hand name of the study in order to ensure the trial collected corresponds to the one used in the review. Unfortunately, Unpaywall not provide enough access to open source clinical trials. Therefore, SciHub, which is another database containing several trials, was used to download PDFs. A package that interfaces SciHub with python called scidownl was used to download PDFs from SciHub. The short hand names of the trials for which there are PDFs are stored in a set. For these, the risk of bias levels for the criteria available is retrieved. This is once again done using the relevant tags and attributes. The criteria is found within the $\langle p \rangle$ tag (see Figure 4.2). This criteria does not always correspond to the exact name of the risk of bias domain that it represents. So, it needs to be matched to the exact criteria being searched for using the key phrases that occur when specific criteria are present. The word 'sequence 'is always present in random number sequence and allocation is always present in allocation concealment, blinding and outcome are always in the blinding of outcome assessment criteria. For blinding of participants and personnel, only the word blinding is always present. Outcome is always present for incomplete outcome data and the word selective is always mentioned when discussing selective reporting bias. These key words

were found by examining the reviews. The short hand study name is then extracted from the STUDY_ID (see Figure 4.2) attribute. This is checked against the short hand name for all the trials that have access to the pdf. If it matches, the risk of bias is extracted from the RESULT (see Figure 4.2) attribute and the data is stored in a dictionary that has the short hand name of the study mapped to the title, the path to the full pdf and the risk of bias data for the study. This data is then written to a JSON file (see Figure 4.3).

```
<STUDY DATA_SOURCE="PUB" ID="STD-Biernacki-1998" NAME="Biernacki 1998" YEAR="1998">
<REFERENCE PRIMARY="NO" TYPE="JOURNAL_ARTICLE">
<AU>Biernacki W, Peake MD</AU>
<TI>Acupuncture in the treatment of stable asthma</TI>
<SO>Respiratory Medicine</SO>
<YR>1998</YR>
<VL>92</VL>
<PG>1142-5</PG>
```

Figure 4.1: Illustration of citation relevance tags

```
<QUALITY_ITEMS MODIFIED="2009-04-27 13:36:28 +0100" MODIFIED_BY="Toby J Lasserson">
<QUALITY_ITEM CORE_ITEM="YES" ID="QIT-01" LEVEL="STUDY" MODIFIED="2009-04-27 13:36:28 +0100" MODIFIED_BY="Toby J Lasserson" NO="1">
<NAME>Adequate sequence generation?</NAME>
<DESCRIPTION>
<P>Was the allocation sequence adequately generated?</P>
</DESCRIPTION>
<QUALITY_ITEM_DATA>
<QUALITY_ITEM_DATA_ENTRY MODIFIED="2008-04-26 08:30:07 +0100" MODIFIED_BY="Toby J  Lasserson" RESULT="UNKNOWN" STUDY_ID="STD-Biernacki
<DESCRIPTION>
<P>No information available </P>
</DESCRIPTION>
```

Figure 4.2: Illustration of risk relevant tags

```
▼ Biernacki 1998:
    0:                                          "Acupuncture in the treatment of stable asthma"
  ▼ 1:                                          "Papers/Train/Acupuncture in the treatment of stable asthma.pdf"
  ▼ 2:
      Random sequence generation:               "Unclear risk"
      Allocation concealment:                   "Unclear risk"
      Blinding of participants and personnel:   "High risk"
      Incomplete outcome data:                  "Low risk"
```

Figure 4.3: Example of JSON data entry

4.3.2 Preprocessing and Making representations

The convert_to_text_and_get_features function uses the PyPDF2, which is a python library that converts PDFs to text to make full text articles. It also uses NLTK to stem the data, remove common words and tokenise the data. The make_represntaion function is responsible for converting the full text to bag of words or bigrams. It does this using CountVectorizer method from scikit learn. This method can perform these functionality when it is supplied

the appropriate parameters. the ngram_range parameter for this function can compute any n-gram processing. So, to create the bag of word representation the tuple, (1,1), is passed as the argument for the ngram_range parameter and for bigrams, the (2,2) tuple is passed.

4.3.3 Training the models

The models will be trained using SVM. The model will be trained using the representation specified in the parameter passed to the function. In order to train models for each possible configuration of risk criteria and risk level, a nested for loop is used. The outer loop iterates over all the criteria which is stored in a list. The inner loop iterated over the risk of bias level which is also stored in a list. Then for each of these configurations, a model is trained. The training can be done using the scikit learn library, which has a function to train a SVM model. This function requires the features and the labels of the data. The features will be the bag of word or bigram model. To create this, the training data loaded from the JSON file is passed to the function that returns the dataframe containing the name of the studies, the full text version of the studies and the label. From this dataframe, the full text articles are extracted from the data frame and passed to the function that creates these representations. After these parameters are extracted they are passed to the SVM function. While training is taking place, five fold cross validation is run and the F1 score is tracked. The parameters of the SVM model is then tuned until the F1 score reaches a level comparable, within 5%, to what was achieved by Marshall et al [1]. The parameters are tuned during training using grid search. The grid search is also implemented using scikit learn. The grid search is the standard way of tuning parameters. It exhaustively searches over a range of values of tuning parameters provided to it. The parameters to tune for SVM include the kernel, which transforms the input data into the required form. Another parameter to be tuned is the regularisation parameter, also called C, which determines the penalty for misclassification of points. The higher the value of C, the larger the margin of the hyperplane. The smaller the value of C, the smaller the hyperplane. Gamma determines how closely the model should fit the training set. So, for grid search all the possible kernels are included in the search including, linear, RBF, polynomial and sigmoid. These kernels transform the data using different functions (see Figure 4.4). For example, the RBF kernel uses a Gaussian distribution. For C, a range of 10 values ranging from 1 to 100000 was needed to get the desired F1 score and for gamma a range of 10 small values from 0.1 to 0.0000001 was required to get the accepted F1 score. Once all the models are trained, they are stored in a dictionary with the risk of bias level concatenated with the criteria as keys the and then written to a stored in a pickle file.

4.3.4 Testing the model

For the testing, a nested loop is once again used to test for every model. Then the relevant model is extracted from the pickle file using the risk of bias level concatenated with the criteria as the key. The testing data is loaded from the JSON file and passed to the convert_to_text_and_get_features to retrieve the testing labels and the full text of the PDFs.

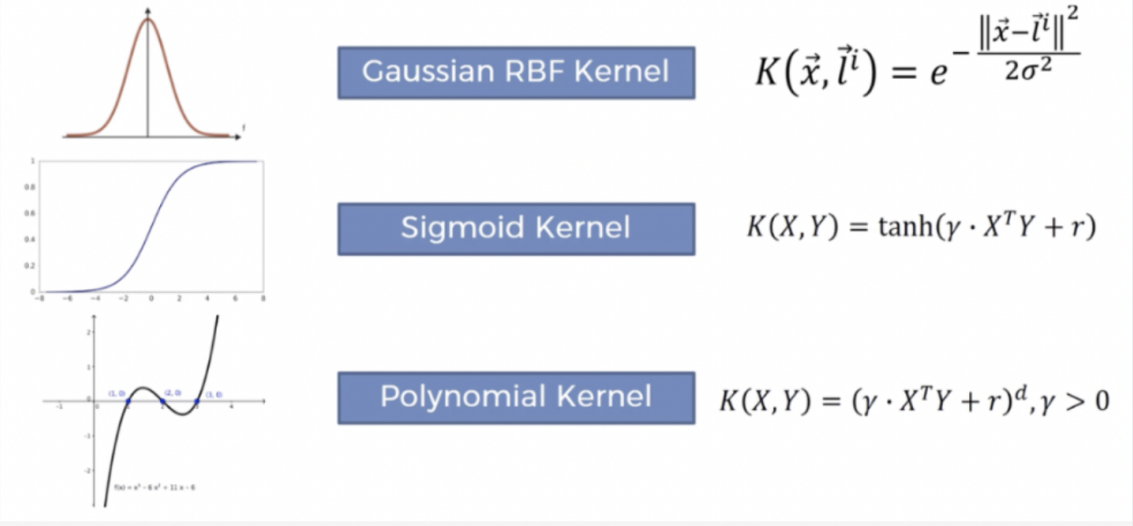

Figure 4.4: Different functions used by the kernels

Then, the model is tested using the predict function in scikit-learn. Then, the predictions are tested using the metrics package in scikit-learn. This package has functions that calculate precision, recall and F1 score when the test result and the predicted results are passed as parameters to these functions. These results are then printed out to the terminal.

4.4 Changes from analysis section

The main changes included using scikit learn to create the data representation models. This is because scikit learn has a more simple method for creating these. The other change is that the data is stored differently in the JSON file. Instead of the risk criteria being the key, the short hand filename is the key. This means that there is less data mapped on to one key making the parsing easier.

Chapter 5

Results And Discussion

The training data was analysed to determine the spread of data across the three different risk criteria. This is done to find out how much training each classifier receives, which can be used to predict the evaluation metric for each risk of bias criteria and level. The total number of trials each risk of bias criteria was present in is collected along with the proportion of studies that fall into the three risk levels, that is, percentage of studies that are low risk, unclear risk and high risk are found for each risk criteria(see Table 5.1). From the table, it is clear that not all risk of bias criteria occur in all the studies. 1499 trials were used for training, extracted from 3500 reviews. 501 trails were used for testing. These were extracted from 1000 reviews.

Criteria	Trials with low risk of bias(%)	Trials with unclear risk of bias(%)	Trials with high risk of bias(%)	Total
Random sequence generation	56.5	24.4	19.1	1491
Allocation concealment	42.7	32.1	20.7	1476
Blinding of participants and personnel	47.2	28.3	24.5	1469
Blinding of outcome assessment	51.2	30.2	18.6	1367
Incomplete outcome data	63.1	20.4	16.5	750
Selective reporting	59.7	25.8	14.5	1384

Table 5.1: Spread of training data
The total column refers to the number of trials the criteria appears in

5.1 Results for test data with bag of words

According to the results gathered, the precision, recall and F1 scores are generally high across all the risk criteria for the low risk of bias level (see Table 5.2). This is because for all risk criteria the low level of risk had the largest percentage of studies. The criteria with the highest F1 score was incomplete outcome of data, which has the largest proportion of studies that are low risk of bias compared to the other criteria. However, allocation concealment has the lowest proportion of studies that are low risk of bias but has the third lowest F1 score. This

suggests that the amount of data is not the only factor that influences performance metrics. The main other factor that could influence this is the level of agreement between researchers about what features of a trial affects a particular risk criteria [19]. Often there is disagreement on this topic between researchers. Also, it has been found that 22% of judgements made by researchers for blinding of outcome assessment did not follow the Cochrane handbook, which outlines how to conduct risk of bias [20]. Another study found that 12% of reviews did not follow the guidelines for random sequence generation bias [21]. This suggests that researchers can make mistakes when assessing risk of bias which can lead to unreliable data. For the other risk levels, it appears that larger amounts of data results in higher F1 scores(see Figure **??**), suggesting that when data levels are low enough, it will outweigh the certainty of the features of a study that will affect risk level. For example, the incomplete outcome data criteria has the lowest proportion of unclear risk and high risk of bias and has the lowest F1 score for both these categories. Overall, the unclear and high risk of bias levels had much lower performance metric scores because of lower amounts of data (see Table 5.3 and Table 5.4). The average F1 scores of these the unclear risk level was about half of that of the low risk level. For high risk, the average F1 score was one-third of that of the low risk. So, the unclear risk level performs better than the high risk level. This is due to there being more training data for the unclear risk level (see Figure 5.2).

Criteria	Precision	Recall	F1 Score
Random sequence generation	0.71	0.67	0.68
Allocation concealment	0.66	0.72	0.68
Blinding of participants and personnel	0.65	0.54	0.58
Blinding of outcome assessment	0.52	0.82	0.63
Incomplete outcome data	0.65	0.94	0.76
Selective reporting	0.62	0.84	0.71

Table 5.2: Table showing the results of bag of words model for low risk of bias

Criteria	Precision	Recall	F1 Score
Random sequence generation	0.3	0.28	0.28
Allocation concealment	0.49	0.54	0.51
Blinding of participants and personnel	0.38	0.32	0.34
Blinding of outcome assessment	0.3	0.48	0.36
Incomplete outcome data	0.21	0.3	0.24
Selective reporting	0.26	0.36	0.3

Table 5.3: Table showing the results of bag of words model for unclear risk of bias

5.2 Results for test data with bigram

With the bigram model, similar trends for the different risk criteria are observed. Once again incomplete outcome data and selective reporting were the criteria with the two highest

Criteria	Precision	Recall	F1 Score
Random sequence generation	0.24	0.22	0.22
Allocation concealment	0.31	0.34	0.32
Blinding of participants and personnel	0.33	0.28	0.3
Blinding of outcome assessment	0.18	0.29	0.22
Incomplete outcome data	0.16	0.24	0.19
Selective reporting	0.15	0.2	0.17

Table 5.4: Table showing the results of bag of words model for high risk of bias

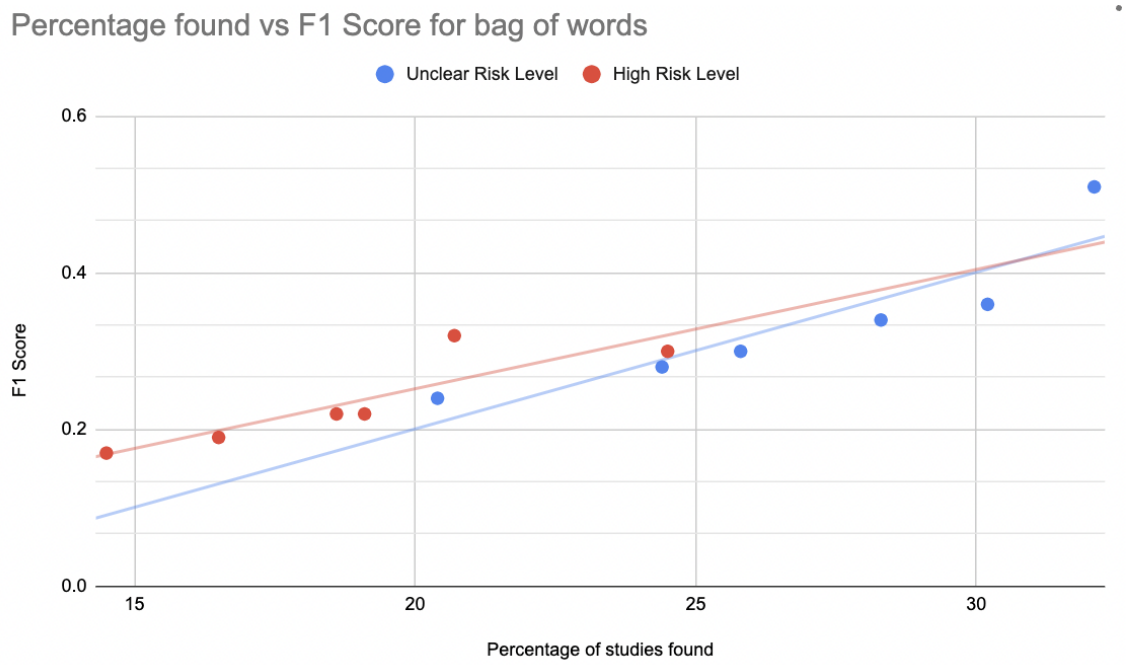

Figure 5.1: Effect of training data size on F1 score for bag of words

highest F1 scores respectively. As with the bag of word model, allocation concealment did not have the lowest F1 score for low risk of bias even though it has the smallest proportion of low risk of bias studies (see Table 5.5). For the other bias levels, larger amounts of data resulted in higher F1 scores (see Table 5.6, Table 5.7 and Figure 5.6). The F1 scores of unclear risk was also about half as much as that of low risk. For high risk, the average score was about one-third of the low risk level (see Figure 5.7). Overall, for all risk of bias levels and criteria, the bigram model performed better. This is because bigrams capture more context around the words, providing more insight into what words are more likely to follow each other. Therefore, it is likely to be much more superior to bag of words. In this case, it performs about 17% better than bag of words because the average F1 score of bigrams is 17% better than that of bag of words. This is true for all risk levels (see Figure **??**, Figure 5.4 and Figure 5.5).

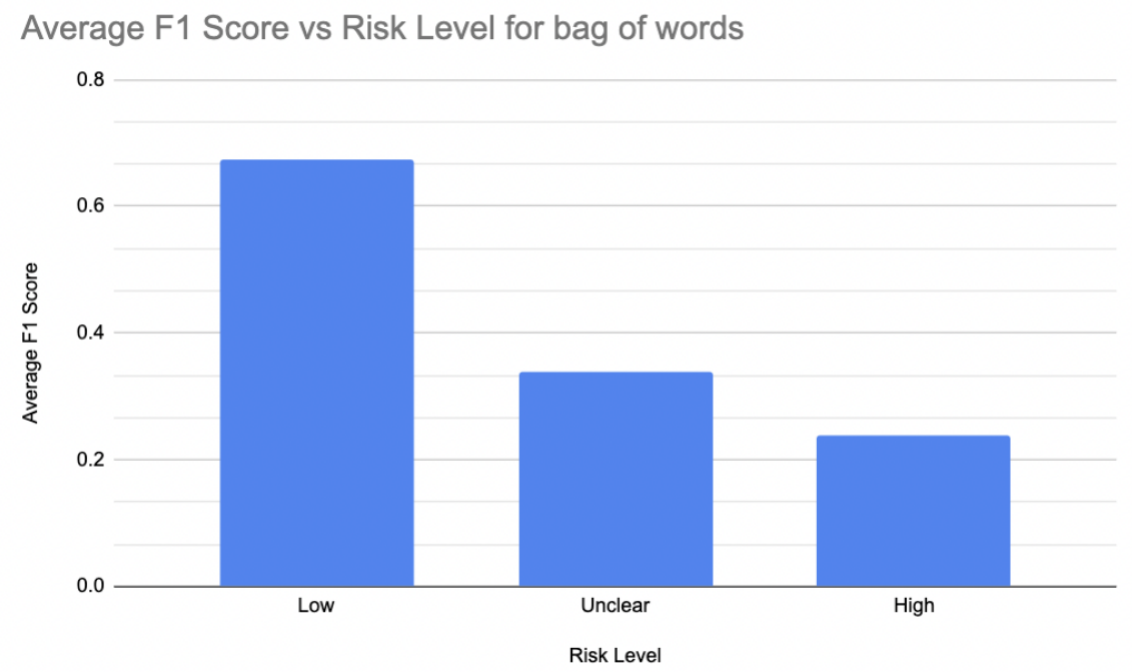

Figure 5.2: Average F1 score for the different risk levels for bag of words The average was calculated for each criteria

Criteria	Precision	Recall	F1 Score
Random sequence generation	0.8	0.71	0.75
Allocation concealment	0.75	0.79	0.76
Blinding of participants and personnel	0.76	0.65	0.7
Blinding of outcome assessment	0.64	0.87	0.73
Incomplete outcome data	0.74	0.97	0.83
Selective reporting	0.73	0.96	0.82

Table 5.5: Table showing the results of bigram model for low risk of bias

Criteria	Precision	Recall	F1 Score
Random sequence generation	0.34	0.3	0.31
Allocation concealment	0.56	0.59	0.57
Blinding of participants and personnel	0.45	0.38	0.41
Blinding of outcome assessment	0.37	0.51	0.42
Incomplete outcome data	0.23	0.31	0.26
Selective reporting	0.31	0.41	0.35

Table 5.6: Table showing the results of bigram model for unclear risk of bias

5.3 Future work

The main improvement that could be done would be developing a joint model that would provide justification along with the risk of bias level for each criteria. This was included in

Criteria	Precision	Recall	F1 Score
Random sequence generation	0.27	0.24	0.25
Allocation concealment	0.36	0.38	0.36
Blinding of participants and personnel	0.39	0.33	0.35
Blinding of outcome assessment	0.23	0.31	0.26
Incomplete outcome data	0.19	0.25	0.21
Selective reporting	0.17	0.23	0.19

Table 5.7: Table showing the results of bigram model for high risk of bias

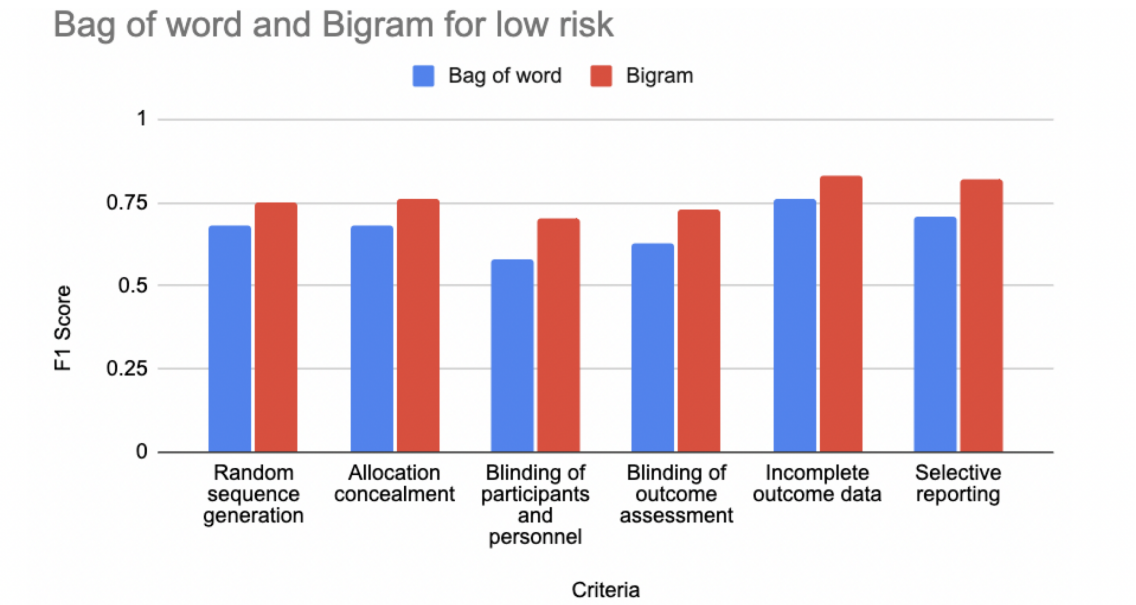

Figure 5.3: F1 scores for bag of words and bigrams for low risk of bias

the initial plan of the project. However, it had to be abandoned because it required a domain expert to label each sentence in a trial as to whether it corresponds to a risk of bias criteria. Sometimes, the authors will use justification from the trial in the systematic review, but this happens very rarely, less than 1 percent of the time, in the data set provided. Another improvement could be using a neural network to train the data, which work similar to a human brain. There are usually many layers in a neural network. Weights are applied to the input of each layer and then passed through an activation function for the final output. This output is then compared with the desired output and the weights are readjusted using backward propagation until the optimum output is achieved. The neural network would be very time consuming to train. Finally, other data representation techniques, such as Word2Vec, could have been used. Although bigrams are hard to beat, the Word2Vec can extract relatedness across trials and might have resulted in better performance metrics.

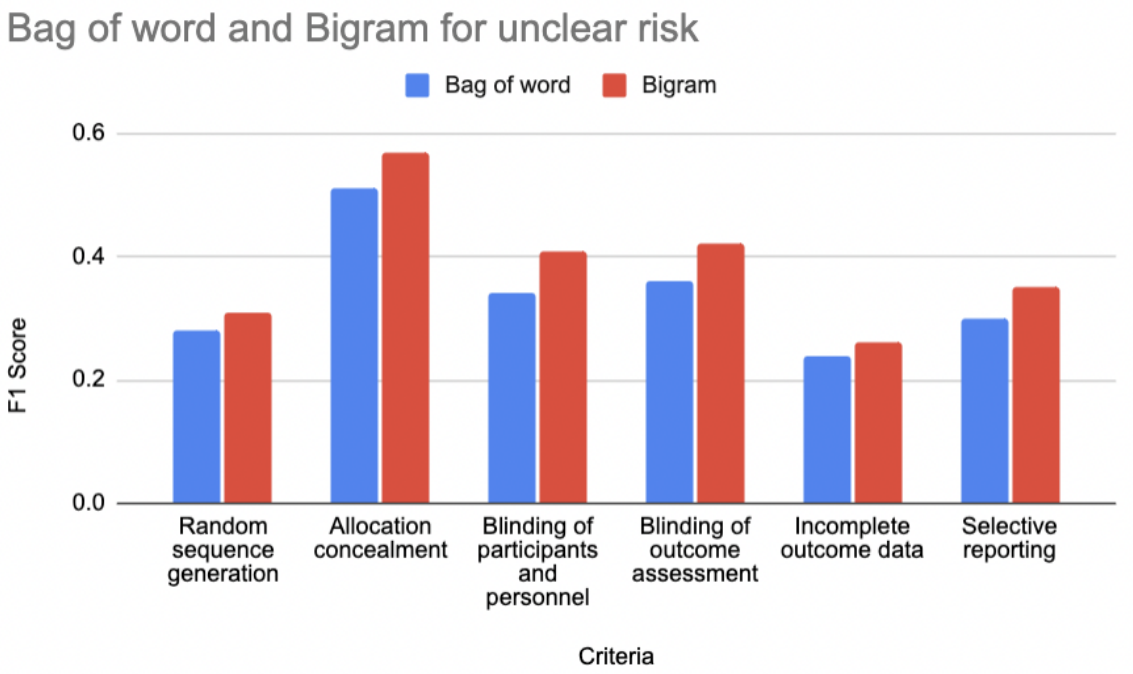

Figure 5.4: F1 scores for bag of words and bigrams for unclear risk of bias

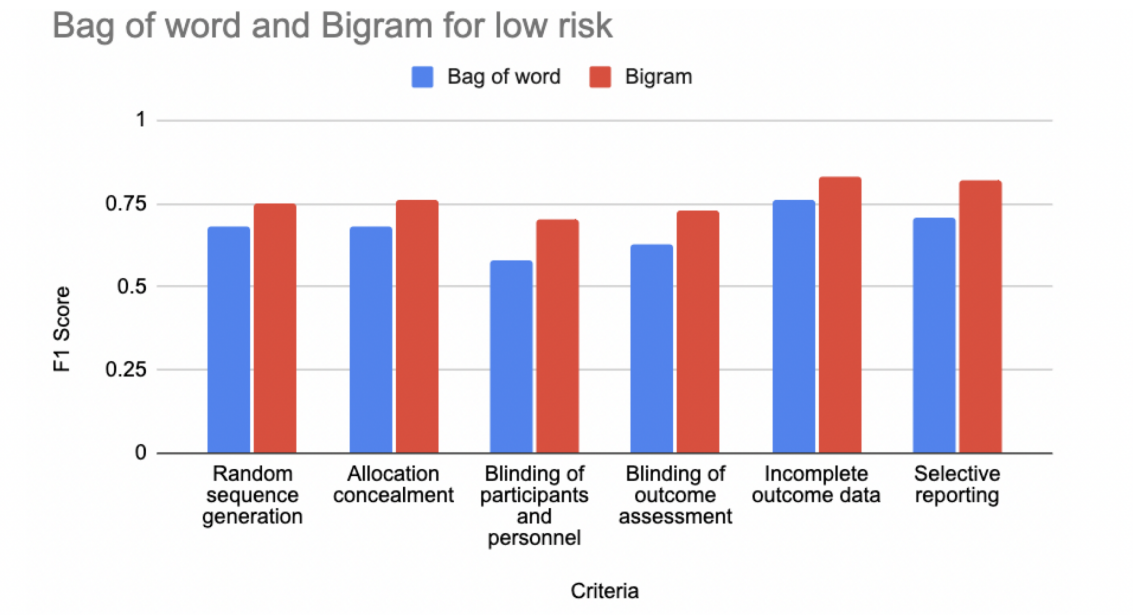

Figure 5.5: F1 scores for bag of words and bigrams for high risk of bias

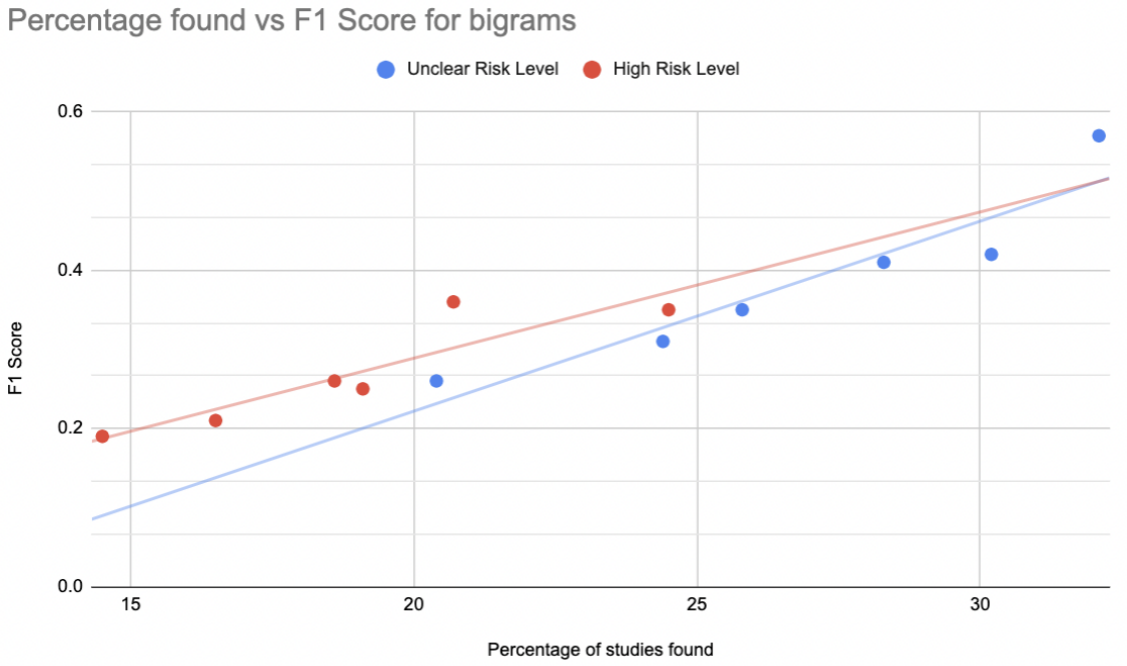

Figure 5.6: Effect of training data size on F1 score for bigrams

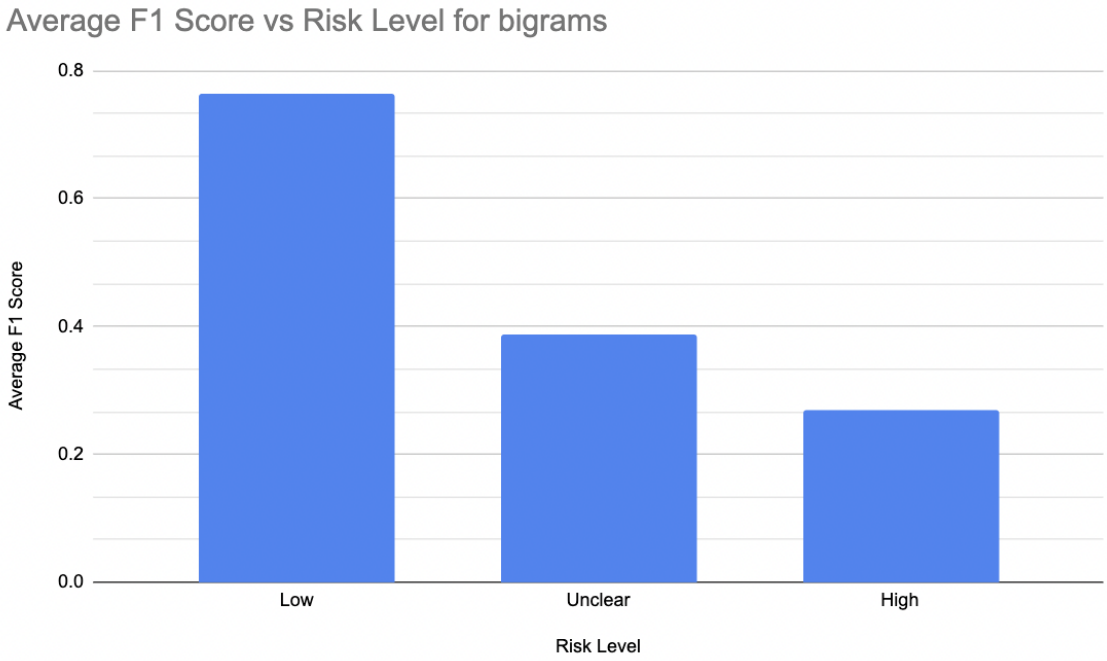

Figure 5.7: Average F1 score for the different risk levels for bigrams The average was calculated for each criteria

Chapter 6

Conclusions

The main takeaways are that it is possible to build a SVM classifier that can be used to determine the risk of bias level for clinical trails. There appears to be many factors that influence the performance metrics of SVM, particularly there seems to be disagreement between authors on how to classify risk of bias. The selective reporting domain seems to show the least agreement [19]. The bigram representation performed much better than bag of words, showing the significance of representation methods. Overall, this project was a success because the results showed that the model with bag of words representation performed equivalent to Marshal et al's basic model for low risk of bias [1]. For the bigram representation, the model outperformed the basic and joint model in Marshal et al[1]. For the other risk levels, the results were significantly worse because of lack of training data. However, no literature is available to compare these results with. The main difficulties in this project was creating the training and testing data using distant supervision. This is because most clinical trials are not open source. In fact, this process took up most of the time for the project. The literature review also proved difficult to conduct because of lack of research on the topic of risk of bias automation. Overall, the project succeeded in what was set out initially and fulfilled the basic requirements and some optional requirements such as training and testing for different data representations and including all risk levels in the models. However, some other interesting optional requirement like creating a joint model was left out because of lack of data and would be facinating to explore as future work.

Bibliography

[1] Iain J Marshall, Joël Kuiper, and Byron C Wallace. Automating risk of bias assessment for clinical trials. *IEEE journal of biomedical and health informatics*, 19(4):1406–1412, 2015.

[2] Kenneth F Schulz, Iain Chalmers, Richard J Hayes, and Douglas G Altman. Empirical evidence of bias: dimensions of methodological quality associated with estimates of treatment effects in controlled trials. *Jama*, 273(5):408–412, 1995.

[3] J.P.T. Higgins and S. Green. *Cochrane Handbook for Systematic Reviews of Interventions*. Wiley, 2008.

[4] H Saltaji, S Armijo-Olivo, GG Cummings, M Amin, BR da Costa, and C Flores-Mir. Impact of selection bias on treatment effect size estimates in randomized trials of oral health interventions: a meta-epidemiological study. *Journal of dental research*, 97(1):5–13, 2018.

[5] Kenneth F Schulz and David A Grimes. Allocation concealment in randomised trials: defending against deciphering. *The Lancet*, 359(9306):614–618, 2002.

[6] Pascal Probst, Kathrin Grummich, Patrick Heger, Steffen Zaschke, Phillip Knebel, Alexis Ulrich, Markus W Büchler, and Markus K Diener. Blinding in randomized controlled trials in general and abdominal surgery: protocol for a systematic review and empirical study. *Systematic reviews*, 5(1):1–6, 2016.

[7] Rudolf W Poolman, Peter AA Struijs, Rover Krips, Inger N Sierevelt, René K Marti, Forough Farrokhyar, and Mohit Bhandari. Reporting of outcomes in orthopaedic randomized trials: does blinding of outcome assessors matter? *JBJS*, 89(3):550–558, 2007.

[8] Ricarda Lieber, Nikolaos Pandis, and Clovis Mariano Faggion Jr. Reporting and handling of incomplete outcome data in implant dentistry: A survey of randomized clinical trials. *Journal of clinical periodontology*, 47(2):257–266, 2020.

[9] David Collier and James Mahoney. Insights and pitfalls: Selection bias in qualitative research. *World politics*, 49(1):56–91, 1996.

[10] Louise AC Millard, Peter A Flach, and Julian PT Higgins. Machine learning to assist risk-of-bias assessments in systematic reviews. *International journal of epidemiology*, 45(1):266–277, 2016.

[11] Iain J Marshall, Joël Kuiper, and Byron C Wallace. Robotreviewer: evaluation of a system for automatically assessing bias in clinical trials. *Journal of the American Medical Informatics Association*, 23(1):193–201, 2016.

[12] Allison Gates, Ben Vandermeer, and Lisa Hartling. Technology-assisted risk of bias assessment in systematic reviews: a prospective cross-sectional evaluation of the robotreviewer machine learning tool. *Journal of clinical epidemiology*, 96:54–62, 2018.

[13] Frank Soboczenski, Thomas A Trikalinos, Joël Kuiper, Randolph G Bias, Byron C Wallace, and Iain J Marshall. Machine learning to help researchers evaluate biases in clinical trials: a prospective, randomized user study. *BMC medical informatics and decision making*, 19(1):1–12, 2019.

[14] Xu Ling, Jing Jiang, Xin He, Qiaozhu Mei, Chengxiang Zhai, and Bruce Schatz. Generating gene summaries from biomedical literature: A study of semi-structured summarization. *Information Processing & Management*, 43(6):1777–1791, 2007.

[15] Xu Ling, Qiaozhu Mei, ChengXiang Zhai, and Bruce Schatz. Mining multi-faceted overviews of arbitrary topics in a text collection. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–505, 2008.

[16] Aaron M Cohen, William R Hersh, Kim Peterson, and Po-Yin Yen. Reducing workload in systematic review preparation using automated citation classification. *Journal of the American Medical Informatics Association*, 13(2):206–219, 2006.

[17] Ben J Marafino, Jason M Davies, Naomi S Bardach, Mitzi L Dean, and R Adams Dudley. N-gram support vector machines for scalable procedure and diagnosis classification, with applications to clinical free text data from the intensive care unit. *Journal of the American Medical Informatics Association*, 21(5):871–875, 2014.

[18] Yoav Goldberg. Neural network methods for natural language processing. *Synthesis lectures on human language technologies*, 10(1):1–309, 2017.

[19] Lisa Hartling, Maria Ospina, Yuanyuan Liang, Donna M Dryden, Nicola Hooton, Jennifer Krebs Seida, and Terry P Klassen. Risk of bias versus quality assessment of randomised controlled trials: cross sectional study. *Bmj*, 339, 2009.

[20] Ognjen Barcot, Svjetlana Dosenovic, Matija Boric, Tina Poklepovic Pericic, Marija Cavar, Antonia Jelicic Kadic, and Livia Puljak. Assessing risk of bias judgments for blinding of outcome assessors in cochrane reviews. *Journal of comparative effectiveness research*, 9(8):585–593, 2020.

[21] Ognjen Barcot, Matija Boric, Tina Poklepovic Pericic, Marija Cavar, Svjetlana Dosenovic, Ivana Vuka, and Livia Puljak. Risk of bias judgments for random sequence generation in cochrane systematic reviews were frequently not in line with cochrane handbook. *BMC medical research methodology*, 19(1):1–10, 2019.