University of Sheffield

William Briggs Report

William Briggs

Supervisor: Dr Mark Stevenson

Panel: Dr Andreas Vlachos, Professor Richard Clayton

Department of Computer Science

April 13, 2018

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: ____________________

Signature: ____________________

Date: ____________________

Abstract

Something

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Medical literature poses interesting challenges for Natural Language Processing (NLP) researchers. The sheer volume of medical data makes it difficult for humans to process efficiently. One key task is the creation of systematic reviews. Systematic reviews are transparent reviews that aim to pull together and critically analyse relevant literature to a topical question. The process of creating a systematic review is rigorous and time consuming with varying degrees of complexity in-between steps. This report will look at the existing work done so far on using NLP as part of the systematic review process as well as the novel work by myself.

1.1 Steps of a Systematic Review

It is useful for us to break down the steps involved in creating a systematic review into subtasks. This way we can observe what techniques can be applied during the relevant sub tasks to improve the efficiency of the process. The following definitions are derived task simplifactions from the cochrane tutorial on systematic reviews: [9].

1. Question definition.
2. Relevant literature search.
3. Data Filtering.
4. Data Extraction.
5. Analysis and Data combination.

1.1.1 Question Definition

One of the best known techniques for formulating a systematic review question is known as the PICO strategy [10]. This technique focuses on exposing 4 pieces of information in the systematic review question: patient population, intervention or exposure, comparison or control and outcome.

Example: (credit goes to [10])

”Is animal-assisted therapy more effective than music therapy in managing aggressive behaviour in elderly people with dementia?”

P	elderly patients with dementia
I	animal-assisted therapy
C	music therapy
O	aggressive behaviour

A potential point of interest would be attempting to generate these questions automatically given some literature context.

1.1.2 Relevant literature search

After formulating a question, systematic reviews need to search for the relevant literature that surrounds this question.

Large medical database-such as pubmed contain relevant studies that can be used to create the review. These databases are typically very large and require concise queries to efficiently retrieve data.

Naturally this can be modelled as an information retrieval problem. We have a large number of documents and we wish to retrieve the most relevant ones. One task for the 2017 CLEF conference was to produce a ranking of the most releavent documents for topics [7]. Many techniques have been proposed for ranking of relevant documents, with varying degrees of performance [3] [5] [8].

An important aspect of the relevant literature search step is the construction of the query. Query creators often apply filters (also known as hedges) to increase the effectiveness or/and the efficiency of the searching. Two key attributes for the query are the precision and the sensitivity (aka recall). By including synonymous phrases e.g: quality adjusted life or quality of well-being or disability adjusted life the sensitivity can be increased, but as expense of the precision. The creation of this query is a task that could potentially have some aspects of NLP applied to it.

1.1.3 Data Filtering

The data filter stage involves reducing the amount of documents returned by the initial query down to a smaller subset of relevant document. This is can also be referred to as the abstract screening phrase [7].

The length of this stage is highly dependant on how many documents were returned by the initial query, often in the excess of 5000 studies for a single query. In response to this, stopping criteria methods have been proposed that aim to optimize two key parameters; the effort and the recall. That is to say we want to get as many relevant documents as possible, whilst looking at the fewest. Examples of approaches include the knee method [12] and the target method [4]. Other techniques could be applied and evaluated such as curve fitting.

1.1.4 Data Extraction

The data extraction phase involves pulling the relevant information from the filtered subset of studies. Examples of important information includes how many people took part in the study and what the results were.

Being able to extract the relevant information from studies presents itself as an information extraction problem. The task to automate the process of extracting relevant information would reduce time and complexities of manually reviewing studies [6].

Chapter 2

Literature Survey

Systematic reviews have many different stages that propose themselves as a candidate for automation. This section is going to look at the techniques that have been applied for some of these stages in previous literature.

2.1 Indexing and Querying Medline and Automated Query Generation

Medline is a large collection of medical literature and data from around the world. Typically each entity will contain a title and an abstract containing some information on the study. Whilest Medline as a whole is very easy to access [1], the large size and complexity of the data makes it difficult to retrieve the relevant information.

Being able to create a reliant index of Medline would help with the effectiveness of the queries. As such existing medline indexes and IR systems have been created [2].

2.1.1 Automated Query Generation

Being able to automate query generation for literature searching would save systematic reviewers a significant amount of time. However, medical literature queries are typically complex and contain multiple levels of logical operators and synonymous term look ups. This makes the task of creating a query manually in-its-self a challenging piece of work.

Rapid Automatic Keyword Extraction Algorithm

Rapid automatic keyword extraction (RAKE) is a keyword extraction algorithm was proposed by Rose, Engel and Cramer in 2010 [11]. This algorithm is used for taking the key pieces of information from text and is useful the domain of information extraction. This algorithm is of interest to us as it has potential usage within the field of query generation.

RAKE is heavily relies on stop-words and punctuation separators as an indicator for the importance of a phrases and words. RAKE will iterator over sequences of words until a stop-word or separator is found, this phrase/word is then split and extracted. Frequency of occurrence (tf) and word co-occurrence matrices can then be used to reduce the key-word set down further.

RAKE can be further optimized by specifying minimum term frequency rates to capture more prominent terms.

2.2 Stopping Criteria

Stopping criteria a topic of being able to know when to stop looking at a set of documents. This could be useful in a decision making process. Consider having 100 relevant documents, where each document contains a binary value. If we looked at 1/3 of these documents and saw a trend of positive values, we could use this to infer the reliability of the remaining documents.

Another use of stopping criteria is when filtering through potentially relevant documents. Consider a query that returns 10000 documents, of which only a small sub-set of these are relevant. Reviewers would need to filter through each of these 10000 documents to pull out the relevant ones. Or it could be that the reviewers are happy to hit a 90% recall of relevant documents, and are happy to miss the remaining 10% in exchange for time-saved.

Two key methods have been proposed for finding stopping points so far, the target method [12] and the knee method. [4]. Both these methods are discussed below 2.2.2

2.2.1 Evaluation Metrics for Finding Stopping Points

In order to evaluate the suitability of our stopping method, we can use two evaluation metrics. The recall, which is simply the number of documents returned for a topic. The effort which is the number of documents we had to look at for a topic.

$$Recall = \frac{R}{|D|} \tag{2.1}$$

Where R is the number of returned documents and D is the set of relevant documents.

$$Effort = \frac{L}{|D|} \tag{2.2}$$

Where L is the number of returned documents looked at.

Naturally we could exclusively optimized each of the parameters by either returning everything in the document collection ($R = |D|$) or by just looking at a single document. ($L = 1$)

Therefore it becomes difficult for us to evaluate our stopping criteria as we need to consider both of these parameters adjacently.

In response to this we can make use of two more evaluation metrics that tell use more about the performance of our stopping method [4]

$$reliability = P[acceptable(S) == 1] \tag{2.3}$$

reliability is computed over all searches and is read as the probability of the acceptability being 1. Where acceptability is calculated as:

$$acceptability(S) = \begin{cases} 1, & recall(S) >= 0.7. \\ 0, & recall(S) < 0.7. \end{cases} \tag{2.4}$$

A stopping point is deemed to be acceptable if 70% of the relevant documents have been found. As such, the reliability is an average over a search method.

2.2.2 Existing Stopping Methods

Target

The target method is a fairly straight forward approach to establishing a stopping point. It was first proposed by Cormack and Grossman [4].

The target T denotes how many documents we should randomly select from our initial query. A larger value of T will increase the effort required as we are more likely to select a document towards the end of the query set. Documents are looked at until the target point T has been reached.

While this approach is shown to acquire 95% reliability, the effort needed is often significantly highly, often requiring us to look at huge volume of documents.

We can evaluate how well this method does against an existing set of relevance rankings. We will use the Sheffield run data from the CLEF 2017 task [7]. We found that a T parameter of 5 was the lowest we could go to still acquire a reliability of over 70%. Overall we were able to obtain an average recall of 90% by making an average effort of 58%.

Chapter 3

Novel Work

In this section the work completed so far will be presented. Two main areas of the systematic review process has been focused on. Stopping criteria and indexing/querying pubmed.

3.1 Random Sample Method to Stopping

As approach to determining when to stop looking at document abstracts returned by the query we are proposing a new sampling method. This approach assumes we have optimum ranking algorithm for returning documents for a query.

The first step of this method is to randomly sample a returned set of documents into a subsets.

$$U = \frac{|D|}{S} \tag{3.1}$$

Where U is the computed randomised subset, D is the document collection and S is the sample size.

We then use this subset U to create a model / baseline for our topic as a way of predicting how many documents one would need to look at to hit a threshold. The intuition behind this approach is that the rate of which relevant documents occur should be relatively similar when the number of returned documents in the same.

A limitation of this sampling method is that for topics with very few documents it is easy for a sample to miss many of them. This creates a subset set bias, where one set contains a larger percentage of relevant documents. Consider a query that returns 10000 candidate documents of which only 10 have been pre-determined to be relevant. Its not too unlikely that a randomly sampled subset would contain 0 relevant documents. We can use the following equation to tell us how much information we can take from a pre-evaluated topic:

$$I = \frac{rel(T_i)}{|T_i|} \tag{3.2}$$

Where T_i is a given topic and rel computes the number of relevant documents for that topic. Therefore I is telling us how useful the topic is at fitting a curve. We can take the average simply by taking the mean of I across all topics:

$$Usefullness = \frac{\sum I}{|T|} \tag{3.3}$$

3.1.1 Curve Fitting

Our first approach uses a simple curve fit against a sample set along with a simple non-linear function. $f(x)$

$$F(x) = n - a\exp^{-kx} \tag{3.4}$$

Where a, k and n are learnt weights and x is an associated return rate for a document.

We can visualise the curve along with the confidence intervals. The Y axis is the predicted number of relevant documents for the topic. X axis shows the true number of documents returned for the query.

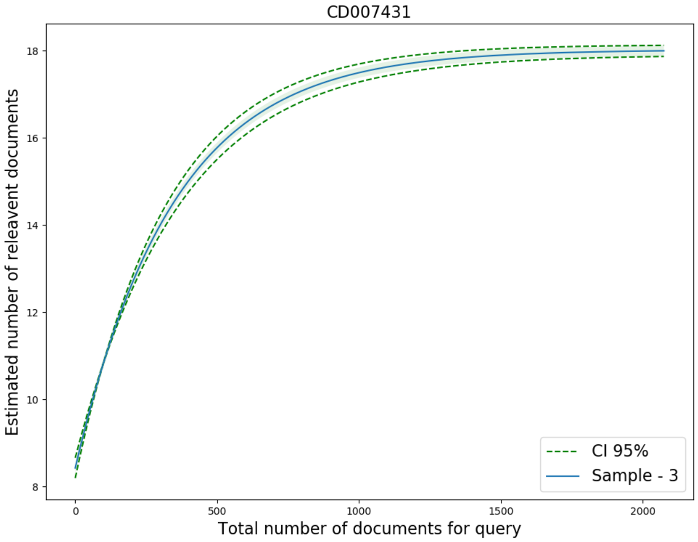

Figure 3.1: Example of fitting a curve for a topic using sampling

sample size	recall	reliability	effort
1	0.91	0.96	1
3	0.66	0.5	0.48
5	0.481	0.33	0.315

Table 3.1: Comparison of different sample sizes against recall and effort. Ranking Method: Test_Data_Sheffield-run-2 [3]

The first sample size of 1 is included to show how the effort metric is effected. For us to use sample everything, we would need to look at everything, as such the effort averaged out at 1. In this example we were only concerned in achieving 70% recall, as such even when sampling everything we would really obtain 100% recall at the expense of 100% effort.

Looking at every 3rd document and then producing a prediction curve will reduce our effort. We are still required to look at at 1/3 of documents, as such the effort will always be above 0.33.

Relevance Ranking

Our results so far have been based on Test_Data_Sheffield-run-2 [3] of CLEF 2017. Naturally, the reliability of our curve is heavily based on how good the initial rankings are for each topic. We can compare different ranking methods for generating our stopping curve. By looking at the CLEF 2017 technology assisted review task [7] we can determine the best candidates to use.

We also introduce a new field. Topics sampled is the number of topics that were evaluated using the curve. This is included as some of the ranking methods do not produce enough relevant documents to generate a suitable curve.

Submission	recall	reliability	effort	topics sampled
Test_Data_Sheffield-run-2	0.66	0.5	0.48	30
Waterloo A-rank-cost	0.65	0.41	0.39	29
Waterloo B-rank-cost	0.70	0.46	0.40	30
auth run-1	0.71	0.5	0.41	30
auth run-2	0.67	0.46	0.40	30
ntu run-1	0.56	0.18	0.54	22
ucl full-text	0.55	0.36	0.70	11

Table 3.2: Comparison of different of sample method using curve fitting for different CLEF 2017 runs. Sample size = 3. Results are taken as averages over all topics for search method. No topic drop-out

This second set of results will use a mandatory drop out parameter for topics with less than

0.5% of relevant documents. The maximum number of topics for this dataset remains at 30.

We also include a confidence interval evaluation for lower bounded range of a 95% confidence interval. The key advantages of using a curve as method of evaluating stopping criteria is being able to make use of this confidence interval in a real systematic review. In the context of a systematic reviewers at the data filtering stage, we could specify that the system is 95% certain that 70% of relevant documents have been found. At which point the reviewer can decide if its worth continuing to look at documents.

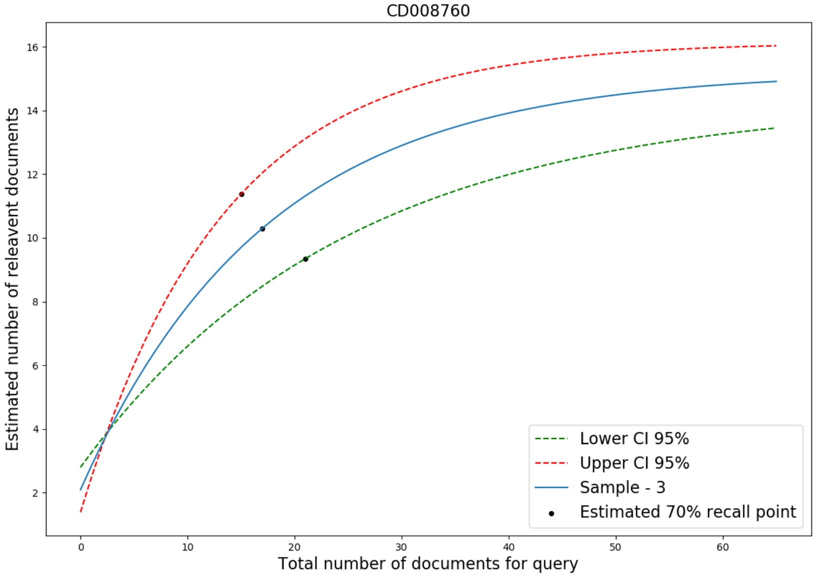

Figure 3.2: Visualisation of using a confidence interval for predicting a stopping point.

Submission	recall-upper-lower	reliability-upper-lower	effort-upper-lower	topics sampled
Test_Data_Sheffield-run-2	0.71 0.67 0.71,	0.57 0.53 0.57	0.50 0.48 0.50	26
Waterloo A-rank-cost	0.68 0.67 0.68,	0.41 0.33 0.41	0.40 0.40 0.41	24
Waterloo B-rank-cost	0.72 0.71 0.73,	0.53 0.53 0.61	0.41 0.41 0.41	26
auth run-1	0.73 0.73 0.74,	0.52 0.52 0.60	0.42 0.42 0.43	23
auth run-2	0.72 0.72 0.72,	0.54 0.52 0.54	0.42 0.42 0.42	22
ntu run-1	0.61 0.55 0.60,	0.26 0.25 0.26	0.54 0.52 0.54	15
ucl full-text	0.61 0.29 0.58,	0.5 0.125 0.375	0.75 0.52 0.68	8

Table 3.3: Comparison of different of sample method using curve fitting for different CLEF 2017 runs. Sample size = 3. Results are taken as averages over all topics for search method. with 0.5% drop-out

We have deliberately compared two of the better participant rankings (Waterloo and auth) and two of the lower performers (ntu and ucl). We can see the quality of the initial rankings significantly influences the performance of our stopping criteria. This suggests there is a important relationship between using a curve to predict a stopping point and how good the initial ranking of documents is. Some of the ranking methods struggle to produce curves and when combined with a drop-out parameter produce become not worth considering in our evaluation.

3.1.2 Gaussian Process fitting

As an alternate approach to fitting a simple curve, we can apply a GP.

We will apply a constant kernel plus a squared-exponential kernel. One of the key challenges when applying a Gaussian Process was the amount of over-fitting.

Submission	recall	reliability	effort	topics sampled [Max 30]
Test_Data_Sheffield-run-2	0.694	0.66	0.49	30

Submission	recall-upper-lower	reliability-upper-lower	effort-upper-lower	topics sampled
Test_Data_Sheffield-run-2	0.71 0.67 0.71,	0.57 0.53 0.57	0.50 0.48 0.50	26
Waterloo A-rank-cost	0.68 0.67 0.68,	0.41 0.33 0.41	0.40 0.40 0.41	24
Waterloo B-rank-cost	0.73 0.73 0.73,	0.71 0.71 0.71	0.41 0.41 0.41	27
auth run-1	0.74 0.60 0.75,	0.56 0.52 0.60	0.42 0.41 0.39	23
auth run-2	0.72 0.70 0.73,	0.52 0.56 0.52	0.42 0.42 0.42	23
ntu run-1	0.67 0.67 0.67,	0.40 0.36 0.40	0.64 0.64 0.64	22
ucl full-text	0.72 0.72 0.72,	0.52 0.56 0.56	0.82 0.82 0.82	25

Table 3.5: Comparison of different of sample method using gp for different CLEF 2017 runs. Sample size = 3. Results are taken as averages over all topics for search method. with 0.5% drop-out

We can see the gp does a better job at fitting to the topics than the curve method. However, for most of the lower ranking runs the effort is increases significantly, suggesting the gp is not fitting very well to the data.

3.2 Indexing and Querying Medline using Systematic Review Protocols

Being able to produce an efficient index of Medline is something highly desirable in the field of medicial research. A more efficient index will produce more concise results when given a query. This could potentially save time in filtering through many non-relevant studies and providing a more concise data-set for reviewers to observe.

3.2.1 Acquiring Key Information from A Systematic Review Protocol

A systematic review protocol is created before the systematic review process is started. A systematic review protocol describes the rationale, hypothesis, and planned methods of the review. The Medline query is created manually with the help of the protocol. Here we are looking to generate a suitable query/relevant information from the protocol to then automatically query Medline.

We will use RAKE [11] to extraction key-words from a protocol. The minimum word occurrence count is set to 1, as the protocols are typically small. We used a pubmed stop list as the phrase splitting parameter. Example shown below:

Topic: CD008122
Title: Rapid diagnostic tests for diagnosing uncomplicated P. falciparum malaria in endemic countries
Objective: To assess the diagnostic accuracy of RDTs for detecting clinical P. falciparum malaria (symptoms suggestive of malaria plus P. falciparum parasitaemia detectable by microscopy) in persons living in malaria endemic areas who present to ambulatory healthcare facilities with symptoms of malaria, and to identify which types and brands of commercial test best detect clinical P. falciparum malaria.

```
endemic countries objective|ambulatory healthcare facilities|rapid diagnostic tests|falciparum parasitaemia detectable|malaria endemic areas|diagnostic accuracy|falciparum malaria
```

The | symbol represents a separation between a phrase. The protcols are pre-processed as follows: Reference removal, lowercase, words less than N length removed, pubmed stoplist. We decided to not perform any stemming/additional manipulation at this stage, due to uncertainty of query format. The resulting content is stored in a separate file appended with '.kwq' (key-word query).

The key-word-query receives some final pre-processing prior to being loaded into our IR system. We used a Lancaster stemmer to reduce words down to a base form. The result is as follows:

```
endem country object amb healthc facil rapid diagnost test falcipar parasitaem detect malar endem area diagnost acc falcipar malar
```

3.2.2 Indexing medline

Medline was downloaded from the online resource. We processed the xml files and retrieved the information for each study - title, id, abstract. To reduce the size, we store each record into a local database, containing only relevant.

We used Apache Lucene to generate an index for the medline local database. The abstract and title were concatenated together. Pre-processing was done using the same format as the query: pub-med stoplist, Lancaster stemmer and lower casing.

We will provide comparisons of various ranking methods as well as the evaluation scores for each.

3.2.3 Results

Results were generated using the eval script from the CLEF 2017/2018 task [7]. We calculated the top N results over the CLEF 2017 training set. We include a random baseline as a measure of perspective.

Run	recall	ap	lastrel	wss100	wss95	normarea	N
Random-baseline	0.05	0.02	126.7	0.0	-0.0	0.024	-
Sheffield-bm25	0.233	0.033	697	0.0	0.0	0.214	1000
Sheffield-tfidf	0.127	0.011	609	0.0	0.0	0.112	1000
Sheffield-boolean	0.138	0.011	700	0.0	0.0	0.123	1000

Table 3.6: Results for IR medline system. 1000 Returned documents (N)

Run	recall	ap	lastrel	wss100	wss95	normarea	N
Sheffield-bm25	0.424	0.038	4239	0.0	0.0	0.343	5000
Sheffield-tfidf	0.273	0.015	3568	0.0	0.0	0.205	5000
Sheffield-boolean	0.265	0.013	3770	0.0	0.0	0.209	5000

Table 3.7: Results for IR medline system. 5000 Returned documents (N)

Run	recall	ap	lastrel	wss100	wss95	normarea	N
Sheffield-bm25	0.583	0.038	20333	0.0	0.0	0.495	25000
Sheffield-tfidf	0.487	0.016	20227	0.0	0.0	0.367	25000
Sheffield-boolean	0.437	0.014	20677	0.0	0.0	0.342	25000

Table 3.8: Results for IR medline system. 25000 Returned documents (N)

Improvements could certainly be made to this system:

- MeSH headings would be useful in expanding the range of the query to capture synonymous terms.
- Tokenization could be optimized to capture phrases of different sizes.

Bibliography

[1]

[2]

[3] Alharbi, A., and Stevenson, M. Ranking abstracts to identify relevant evidence for systematic reviews: The university of sheffield's approach to clef ehealth 2017 task 2. *CLEF 2017* (2017).

[4] Cormack, G. V., and Grossman, M. R. Engineering quality and reliability in technology-assisted review.

[5] Cormack, G. V., and Grossman, M. R. Technology-assisted review in empirical medicine: Waterloo participation in clef ehealth 2017. *CLEF 2017* (2017).

[6] Jonnalagadd, S. R., Goyal, P., and Huffman, M. D. Automating data extraction in systematic reviews: a systematic review.

[7] Kanoulas, E., Li1, D., Azzopardi, L., and Spijker, R. Clef 2017 technologically assisted reviews in empirical medicine overview.

[8] Lee, G. E. A study of convolutional neural networks for clinical document classification in systematic reviews: Sysreview at clef ehealth 2017. *CLEF 2017* (2017).

[9] Nunn, J. cochranes. `http://cccrg.cochrane.org/animated-storyboard-what-are-systematic-reviews`.

[10] of Tasmania, U. pico. `https://utas.libguides.com/SystematicReviews/FormulateQuestion`.

[11] Rose, S., Engel, D., and Cramer, N. Automatic keyword extraction from individual documents.

[12] Satopa, V., Albrecht, J., Irwin, D., and Raghavan, B. Finding a "kneedle" in a haystack: Detecting knee points in system behavior.