Medical Information Retrieval System

Patrick Mc Connell

Final Year Project – 2017/18
B.Sc. Computer Science and Software Engineering

Department of Computer Science
Maynooth University
Maynooth, Co. Kildare
Ireland

A thesis submitted in partial fulfilment of the requirements for the B.Sc. Computer Science And Software Engineering.

Supervisor: Liadh Kelly

Contents

Declaration:

I hereby certify that this material, which I now submit for assessment on the program of study as part my final year project report, module code CS440, is *entirely* my own work and has not been taken from the work of others. Such work that has been previously done and used as resources in conducting this report has been cited and acknowledged within the text of my work. I acknowledge that my work is in line with the Maynooth Universities Plagiarism guide as stated and available at: https://www.maynoothuniversity.ie/sites/default/files/assets/document/Plagiarism%20Sept%202015_0.pdf.

Signed: Patrick Mc Connell **Date**: 28th March 2018

Acknowledgements:

I would like to thank my main supervisor, Liadh Kelly who provided all the knowledge that I needed to conduct the relevant research in this domain. All the material and relevant information that was acquired to conduct this project and the learning outcome that was achieved is predominantly achieved because of my supervisor. I would also like to thank the university for all the supplementary material and resources that were available to me, as well as my peers for allowing me to solidify my approach to this project.

Abstract:

Often laypeople posing queries to find out medical information causes a problem when considering the methodologies of the retrieval and the base language that the query is posed in. This Cross-Lingual Medical Information Retrieval Investigation of weak baseline algorithms used over four different metrics, Normal Discounted Cumulative Gain (NDCG) and Precision at the top five and 10 documents and three choice algorithms, Vector Space Model (VSM), Okapi's Best Match 25 (BM25) and Dirichlet Smoothing describes an in-depth analysis on currently used retrieval algorithms to decide on strong and weak retrieval method over 8 different languages. The use of the 2015 CLEF eHealth Information Retrieval test collection is used to test laypeople's medical queries against a document collection of over one million medical documents. Medical Queries in each language are first translated into English to allow searching on the English document collection. Runs are carried out against an Inverted Index of medical documents with the use of a handmade relevance scoring for each query to compute the relevance of each of the sixty-six different medical queries posed. The standard Information Retrieval tool, Trec_Eval is used to compute the different relevance scoring regarding the purposed metrics and statistical significance testing is then done with the Wilcox Signed Rank test to illustrate that the Dirichlet Smoothing language-based model is the strongest retrieval model to use overall when considering different languages.

Chapter 1: Introduction

1.1 The Topic:

Medical information retrieval refers to the methodologies and technologies that we use to access medical information archives such as medical journals and medical content found on the web. Health-related content is one of the most searched-for topics on the internet and is ever increasing as such the need for accurate retrieval is growing. As outlined by a survey done by the pew research center, conveying “that nearly 70% of search engine users in the U.S. have performed health-related searches” [1]. The CLEF eHealth evaluation challenges were created to support the development and improvement of medical information retrieval systems over a multilingual domain. This then is exemplified by projects such as the Khresmoi[1] project and its follow-on project, kconnect[2] that is ongoing. The Khresmoi project is defined as “The Khresmoi project is developing a multilingual multimodal search and access system for medical and health information and documents.” [2]. The key areas that these are concerned with is in the research and development of medical search support solutions. In the current age of big data, with the national library of medicine[3] harvesting over 90,000 medical journals, and with nearly 75% of these documents being published in English, medical information retrieval is of utmost importance especially over a multilingual domain. Shared challenge test collections have been created to support medical information retrieval technique development. These test collections typically consist of a large sample of medical documents such as medical web pages, [referred to as a *document collection*], a set of queries the target users, for example, a layperson with a medical query might pose to the retrieval system [referred to as *topics* or *queries*] and a list of documents which answer the information need expressed by each query referred to as a *result file*.

This report uses the CLEF eHealth 2015 [3] information retrieval (IR) Task 2’s test collection[4] to investigate medical information retrieval systems over several choice query languages and for different retrieval algorithms. These test collections that are used adhere to HONCode Principles [approx. 60–70% of the collection], meaning they are predominantly health and medical sites [4]. The CLEF eHealth task consists of around one million medical documents, given in their html markup with hyperlinks and basic text included. CLEF eHealth has been running as an activity within the benchmark labs of the CLEF Conference (Conference and Labs of the Evaluation Forum) since 2013. Each year CLEF eHealth offers IR, information extraction (IE) and information management tasks to volunteer task participants which aim to evaluate systems that support laypeople in searching for and understanding health information [5,6,7,8]. In 2015 the task was built around a user centered health information retrieval use case by trying to mimic laypeople’s queries that are confronted with a symptom or condition to help find out more information [9]. The goal of this task is to evaluate the effectiveness of information retrieval systems when searching for medical information on the web. Strong baselines are necessary when evaluating developed information retrieval techniques as well as developing ones. This report uses 67 different queries that have been used in the 2015 CLEF eHealth information retrieval test collection over 8 distinct languages, namely English, French, German, Italian, Portuguese, Arabic, Czech and Persian. It aims to evaluate the different benchmark

[1] http://www.khresmoi.eu/
[2] http://kconnect.eu/
[3] https://www.nlm.nih.gov/
[4] https://sites.google.com/site/clefehealth2015/task-2

algorithms, such as BM25 and Vector Space Model that are used for medical information retrieval to identify what benchmarks retrieval algorithms should be used in different languages.

1.2: Motivation

Users often look to the web for information and diagnosis regarding their medical query. Consider the elderly and individuals living with long term illnesses that require more specific and complex information regarding their condition, where medical advancements and new treatments mean that their current knowledge about their illness may not be sufficient. Often varying medical knowledge and diction arises as a problem for these individuals querying the web, making it difficult to return the most relevant information which in turn restricts their ability to access the information they seek. Much of the medical documentation available online is written in English, focusing on monolingual retrieval. However, there is a growing population of non-English speaking individuals accessing the web to search for medical information. These user queries can be translated from the source language to English to search for English document collections. All these users require relevant medical information and with there being copious amounts of documents online referencing their illness or condition, not all are advisable or admissible to the general user. Many medical documents uncovered by these users are often written by medical practitioners for other medical practitioner and as such, laypeople are obfuscated when trying to find the medical information they seek, and some of the retrieved information is unreliable. It is evident that the use of strong benchmark baselines is needed to compare against newly hypothesized retrieval methods as very often, comparison against weak baselines are used and shows no statistically significant improvements over stronger baseline choices[leveling]. As shown in this paper, “Additional experiments employed the ranking scheme BM25, which increased the number of relevant and retrieved documents and the mean average precision significantly (0.3878 MAP)” [10] this shows that there are different results depending on the retrieval method used, such as BM25, and the languages that are queried. This shows that there is not currently a strong enough baseline for comparing developed retrieval techniques for query translated from non-English languages to English.

1.3: Problem Statement

The difficulty with medical information retrieval is one that has been addressed and progress has been made within this sphere. As outlined above, the collection of people querying for medical information is diverse and unique, and with medical diction being thinly skewed to medical practitioners, this aspect of information retrieval is of foremost importance. This difficulty with sufficiently accurate medical information retrieval is compounded when we work with different languages that may not have direct translations or when language sentence structure makes it difficult to create the request. An example of this across many languages is the English word ‘have’. This is difficult in many languages as there is no direct translation, such as that in Finnish where saying “I have X” translates to “at me is X”. This creates ambiguity of the meaning of the request. This project then speculates that it is evident that the use of strong benchmark baselines for different languages. This can be different for different languages and as such, strong baselines are needed to compare against newly hypothesized retrieval methods as very often, comparisons to weak baselines are used and this shows no improvement on medical information retrieval when a known algorithm performs better and are not used for comparison.

1.4: Approach

In this project the 2015 CLEF eHealth test collection was used. An inverted index of about one million medical documents was created. Queries were translated from their source language to English. These were queried on the inverted index using multiple *standard* retrieval approaches. Retrieval performance of each approach is evaluated using *standard* retrieval evaluation metrics to determine the best approach to use as a strong *baseline* for each language. An analysis of the retrieved results was then conducted.

To approach the problem of people comparing their information retrieval techniques to weak baselines, and hence then to establish strong baselines for the different languages, it was necessary to evaluate this problem by conducting the relevant research into the area of medical information retrieval. Firstly, the identification of the different obstacles that would be presented in this project and the technologies that would be used to find suitable solutions. Through extensive research, it was found that the information retrieval library that would be most suitable to use was Lucene. Lucene[5] is a java-based information retrieval library primarily used for full text retrieval. Alternatively, there were other considerations such as the terrier[6] and lemur[7] libraries. Both terrier and lemur were strong candidates for use however the ease of use of Lucene, accompanied with the vast online resources and it being able to easily retrieve information ultimately made it the choice for this project. Other mentionable technologies that were used in this project were SQLite[8], used for storing the medical queries of the different languages, Lukeall[9], a tool that allowed the user to view and search the index once it was created for validation, and google translates API for translating the queries into English.

1.5: Metrics

To evaluate the findings Trec_Eval[10] was used. This is the standard tool used in the TREC community for evaluating ad-hoc retrieval runs given the retrieval runs and the judgement files, i.e. *Result files*. Retrieval runs refer to the process of retrieving the results using a given algorithm and using this with Trec_Eval to evaluate the results over the given metrics. This was used to evaluate the retrievals runs over different metrics, namely precision at 5 [P@5], precision at 10 [P@10], and normal discounted cumulative gain (NDCG) at these respectively. These were used as they are the same metrics used in the CLEF eHealth task outlined above [3]. These are the standard metrics used in the CLEF eHealth information retrieval tests, and as such use of these are imperative as a control variable to consider this research applicable and comparable to other research done in this domain.

---

5 https://lucene.apache.org/
6 http://terrier.org/
7 https://www.lemurproject.org/
8 https://www.sqlite.org
9 https://github.com/DmitryKey/luke/releases

10 https://github.com/usnistgov/trec_eval

1.6: Significant Achievements

Some of the noteworthy achievements that were accomplished during this research project were successfully creating an inverted index with Lucene that is searchable. It was also necessary to strip out the xml content of text documents that contained the queries and clean them as to process without error. It was also vital to the project that successfully using the google API to translate the queries and store these in a database with SQLite was achieved. It was then necessary to investigate the results and discover significant discrepancies in different languages in retrieval that prompts issues with medical information retrieval low threshold algorithms. From the research conducted, a proposal of using the language-based model, Dirichlet Smoothing is suggested as it has performed better over seven of the eight languages test and has shown considerable statistical significance in evaluation. The other language retrieval model that should be considered for the other language is BM25 for Portuguese retrieval, however it only performed marginally better in this case. This is being suggested as content retrieved is only relevant in the first 5 documents and BM25 performed marginally better in this metric [Graph. 7, see appendices] [pg. 564, 11].

Chapter 2: Technical Background

2.1: Topic Material

Information retrieval is the acquisition of information that is stored and is then presented to the user. The book that was used to gain information on this is 'An Introduction to Information Retrieval' [12] and Although it is exhaustive with content that is not necessarily applicable to this project, it achieves in giving a strong insight into the meaning behind information retrieval and the strong mathematical background that it is built on. It strongly described the ideas of "*test collections*", which consists of a document collection, queries, and relevant judgements for this collection. Some of the different collections used for adhoc information retrieval are the *Cranfield*[11] collection and the CLEF collection, which is being used in this project.

The most applicable information for this project that was found within this book was a description and an explanation of two retrieval methods, namely BM25 and Vector Space model. BM25 known also as Okapi BM_25, is a best match information retrieval algorithm. It is based on a probabilistic retrieval framework which works off weighted terms to decide what relevance weighting the document will get. The figure below (1) describes how the algorithm works as follows:" The first part of the expression reflects relevance feedback (or just idf weighting if no relevance information is available), the second implements document term frequency and document length scaling, and the third considers term frequency in the query" [pg233-234,12]. It has been used widely in different information retrieval test collections and is considered a standard method across the TREC[12] community.

$$.\mathbf{RSV}_{d} = \sum_{t \in \mathbf{q}} \boldsymbol{log}\left[\left[\frac{(|VR_t| + \frac{1}{2})/(V\,N\,R_t + \frac{1}{2})}{(df_t - |VR_t| + \frac{1}{2})/(N - df_t - |VR| + |VR_t| + \frac{1}{2})}\right] \times \frac{(k_1 + 1)tf_{td}}{k_1((1-b) + b(L_d/L_{ave})) + tf_{td}} \times \frac{(k_3 + 1)tf_{tq}}{k_3 + tf_{tq}}\right] \quad (1)$$

Best Match 25

[11] http://ir.dcs.gla.ac.uk

[12] http://trec.nist.gov/

Another method of information retrieval is known as Vector Space Model(VSM). This is an algebraic model that represents the documents as vectors to rank the documents weight further than a Boolean retrieval method of being either relevant or not (0 or 1). Its basis is in linear algebra, namely the use of cosine Similarity to compute the weights and relevance scores for the different queries. In the figure below (2), ||n|| is denotes the normal vector of *n,* and $A_2 \times n$ denotes the intersection value of A_2 and n. it gains its strength when looking for like terms in a document that you might also be interested in, as stated "Such a search is useful in a system where a user may identify a document and seek others like it"[pg. 112, 12].

$$\cos\theta = \frac{d_2 \times q}{||d_2|| \times ||q||} \quad (2)$$

Vector Space Model

Another method of information retrieval that was used in the project was Dirichlet Smoothing (with μ = 1000). This is a language-based model, which are described as "The language modeling approach to information retrieval represents documents as generative probabilistic models" [13]. Figure below depicts the mathematical model that is used for Dirichlet smoothing information retrieval. The terms are described as follows: $F_{t,d}$ is the number of times t appears in d, L_d is the length of the document, $M_c(t)$ as the maximum likelihood model based on term frequencies in the collection as a whole, L_t is the number of times t occurs in collection C, L_c is the total number to tokens in the collection, t is a term, μ is the prior and q is a query [14]. Documents here are modeled with a hidden Dirichlet random variable that specifies a probability distribution on a low-dimensional topic space to obtain a generative model.

$$\sum_{t \in q \cap d} q_t \cdot \log \frac{M_d^{\mu}(t)}{\alpha_d M_c(t)} + n \cdot \log \alpha_d = \sum_{t \in q} q_t \cdot \log\left(1 + \frac{f_{t,d}}{\mu} \cdot \frac{l_C}{l_t}\right) - n \cdot \log(1 + \frac{l_d}{\mu}) \quad (3)$$

Dirichlet Smoothing model

There are many different shared evaluation challenges out there within the Information Retrieval (IR) community. There are multiple established different shared challenged series such as TREC[13], NTCIR[14], FIRE[15], CLEF[16] and most recently MediaEval[17] [15]. All these different campaigns within the Information Retrieval (IR) space allow for cross comparable evaluation. CLEF is just one forum that is looking at these evaluations and CLEF eHealth is one campaign within this forum. With all these different campaigns within the Information Retrieval (IR) space the need for the use of strong baselines when evaluation of these different tasks is evident. As discussed by Leveling Et Al. from the years of 2003 to 2005, with consideration of the Mean Average Precision (MAP) metric that was used in these tasks, when employed the use of the Information Retrieval model BM25, the results were significantly better than in previous years [9]. This exemplifies that the models that are currently used are not strong enough and that over different languages, different models perform better or worse and that there is currently no standard strong baseline for Information Retrieval over different languages.

---

13 https://trec.nist.gov/
14 http://research.nii.ac.jp/ntcir/index-en.html
15 http://fire.irsi.res.in/fire/2018/home
16 http://clef2017.clef-initiative.eu
17 http://www.multimediaeval.org/

Some of the other material that was used in this project comes from the book 'Information Retrieval' [16]. This was of significant use in this project as it helped solidify the idea of Information retrieval. It simply stated what was of use and defined the mathematics separately. It provided excellent insight to the different retrieval methods used. It gave a great understanding into the standard metrics that are used to evaluate information retrieval tasks, such as Precision and Recall.

Some of material done in the same domain such as that done by Koopman expressed the same concern with medical information retrieval. In this paper they state, "We require IR systems capable of bridging the 'semantic gap' – overcoming the mismatch between the terms found in documents and the terms used in queries" [17]. This again illustrates the demand for effective retrieval systems in the medical domain and conveys strongly the need for strong benchmarks for algorithms over a cross-lingual space to be evaluated.

Another study done in the medical information domain which focus' mainly on the semantic similarity and relatedness states that "The issue is particularly acute in the medical domain due to stringent completeness requirements on such IR tasks as patient cohort identification" [18] when talking about the use of thesauri of synonymous or nearly synonymous terms for information retrieval. This strengthens the need for research to be done in this domain and epitomizes the reasons for investigating different retrieval algorithms, especially over the cross lingual territory which has the added complication of translation errors and language semantics.

2.2: Technical material

This tutorial [19] on the Lucene Tutorial page was very useful at the beginning of the project as it provided a good starting off point in getting to grips with coding with the library. Lucene is a high-performance search engine used for information retrieval and is completely built in java. It built the foundation of the work that would later be altered, providing a good example of an" simple indexer which indexes text and HTML" [18]. It provided an intuitive understanding of the practical use of Lucene in the context of information retrieval, it provided more in-depth knowledge of the process of working with the Lucene Library, namely the order in which things need to be done. It also provided a great proof of concept at the beginning stage for the newly developed code's behavior.

This blog by 'Uwe Schindler' [20] talks about the use of the virtual memory in java's Lucene library, focusing on the use of the *MMapDirectory* method rather than using the standard *simpleFSDirectory* or *RAMDirectory*. This was helpful in this project as it allowed optimization when handling the memory management of a massive index of documents. This was useful as it used a kernel function called "MMap". It helped map data to the physical memory from the disk and cache. It allows us to not have to use an unnecessary system calls as the Memory Management Unit (MMU)/ Transition Lookaside Buffer (TLB) handled all mapping for us. This allows no concurrency issues and paging in and out of buffers as it is handled by the O/S kernel. This was important as the version of code that I was working off at the beginning of the project to get to grips with Lucene was Lucene 3.1, however some of the known methods used in this sample were no longer used or accessed in the same way. Upgrading to the newest version of Lucene, 7.0 was necessary and with use of the online documentation[18] , refactoring the code to be compatible with this version and the methods that it was using was much simpler. This was an elegant solution to the memory

[18] https://lucene.apache.org/core/7_0_0/

problem as Creating the inverted Index for the information retrieval was the center piece to completing this project.

With this comprehensive rundown of the trec_eval[19] tool, and with a small amount of configuration to create an executable for Cygwin on windows, this page provided all the knowledge that was needed to evaluate my runs under the different metrics that were used. It provided the basis that was needed to execute the evaluation and to properly understand the metrics that were being used.

Chapter 3: The Problem

The problem presented in this report is that there is a difficulty in retrieving medical information, especially in the cross lingual domain as the cross lingual retrieval systems (CLIR) are not efficient enough [10]. Although there is research being done, weak baselines are always being used for retrieval across different languages. This is exemplified by the fact that there is no know benchmark retrieval algorithm that is best use for each individual language. This presents the reasoning for investigation of different retrieval algorithms to identify which algorithms work as strong baselines in different languages. Some of the recent papers on cross-lingual information retrieval (CLIR), such as that by D Zhou states that “out-of-vocabulary terms can severely degrade the retrieval effectiveness of a CLIR engine, especially when the source queries being translated are very short” [20]. As of present, there is no strong enough baseline for information retrieval in different languages as expressed by Leveling Et Al when researching into German Information retrieval TestDatabase (GIRT) [9]. This reflects on the necessity for strong baselines in cross lingual information retrieval, especially when we consider laypeople searching for medical information.

Figure 1 shows the workflow of how information systems operate in this regard and how the end user’s queries are processed. It is evident from this model that there are many different variables that need to be considered such as the method of stemming and translation which inevitably influence the retrieval process. From the diagram, the user queries in their native language. This query is then automatically translated to English for searching on the English document collection. The query is them stemmed and reduced to its root derivation[20] as can be seen in the official website. This is done In the *Query Operations* component of the diagram. This query is then ran against the index of medical documents. This is done by using one of the information retrieval algorithms such as BM25 or Vector space model which determines what documents it deems as relevant for this query. These results are then returned to the user.

[19] https://githuzb.com/usnistgov/trec_eval
[20] http://snowball.tartarus.org/algorithms/porter/stemmer.html

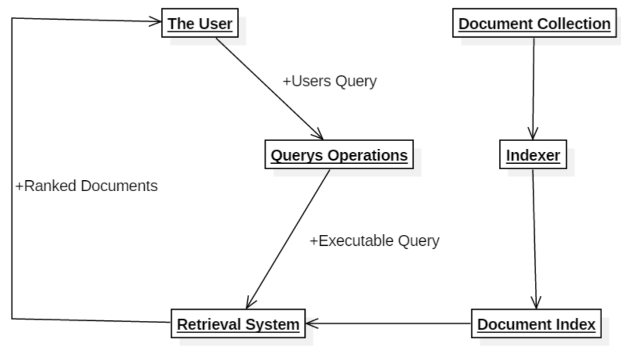

Figure 1: System Architecture

In general, the CLEF eHealth tasks have identified different methods and subgroups that medical information retrieval effects and has made efforts to improve this. As outlined by Leveling Et Al "Results are still inconclusive in which cases NLP methods provide a better performance and even seem to depend on the ranking scheme employed" [10] which again illuminates the problem at hand of weak baselines used in Cross Lingual Information Retrieval (CLIR). It is also known that a lot of the research and techniques used in Information Retrieval Tasks such as that set by CLEF eHealth each year perform well on ad-hoc runs against relatively low baselines which in turn, provide somewhat misleading improvements in retrieval.

Considering this, the problem of strong and weak baselines for medical information retrieval needs to be addressed. This project focuses on the different retrieval methods used such as BM25, Vector Space Model (VSM) and Dirichlet Smoothing to investigates these with the goal of identifying a strong baseline for medical information retrieval for the different chosen languages. It highlights the performances of the different languages and retrieval algorithms and shows that in a cross lingual domain, there is a need for strong baselines for the different languages and that there is not necessarily an algorithm that fits all purposes when translating the queries into English to search for English medical documents.

Chapter 4: The Solution

4.1 Query Topic Translation and Processing:

4.1.1 Preparing the data:

In designing the solution to this research project, there were several various aspects that needed to be addressed. The first matter that needed to be addressed was cleansing the data. The queries needed to be in text format to pass to the translation process.

```
<topics>
<top>
<num>clef2015.test.3</num>
<query>suchá červená a šupinatá chodidla u dětí</query>
</top>
</topics>
```

Figure 2: Czech XML Query

Figure 2 shows the data contained with the XML tags when first presented. It was necessary to parse out the XML content first, meaning removing the tags and only having the test number and the query left. Then the lines needed to be delimited to show the query and it reference number on different line for the translation process. This was done by adding a coma to the end of each query and test number. This was needed as with specific languages, words depend on the preceding and post letter[21]. This is especially relevant with the Arabic language, as its characters can change depending on what comes before and after. Following queries of the same language could distort the translation process if not read in on a different line.

```
qtest.1 0 agesp1837_12_000048 0
qtest.1 0 agesp1837_12_000062 0
qtest.1 0 agesp1837_12_000075 0
qtest.1 0 agesp1837_12_000095 0
qtest.1 0 agesp1837_12_000205 0
qtest.1 0 agesp1837_12_000434 0
```

Figure 3: Relevance of Queries

Figure 3 depicts the standard query relevance test file. This is used as a ground truth for how relevant different queries are to different documents. It has been generated by hand by an accessor who manually determines what documents are relevant to which query. This Is used against our retrieval to gather the different scorings for the different retrieval methods.

4.1.2 Google Translate:

To translate the queries, Google translates API was used. This was used as it is one of the best free choices available. It was necessary to create an account with google cloud applications to gain access to the API so that a key could be generated for connection when passing the queries from the code to the API. A new instance of a translator needed to be created to pass the private key for authorisation and from that, a file writer was created to pass the entire file and retrieve the translation. The delimitation used above allowed each query to be easily exported on new lines making it easier to view the translations.

[21] https://www.madinaharabic.com/Arabic_Language_Course/Lessons/L004_003.html

4.1.3 Character Encodings:

It was also necessary to ensure that the encoding of the files was in UTF-8 as with different languages, there was characters that were not in the character set, meaning they were not displayed. This was necessary as with the integrated development environment that I was using, Eclipse, the default is CP-1252 on windows machines. This was of utmost importance to the translation process as with languages such as Arabic, there are characters not in this character set and this would not allow for a legitimate translation. Another concern in the translation process was that with Arabic characters are read for right to left when reading. Google takes in a bit stream from left to right which meant that if the characters were not correct, the translation would completely distort the process and different words would be returned.

4.2 Document Collection Indexing*:*

4.2.1 Creating the index:

One of the main difficulties that was presented in this project was the use of Lucene. The main task that was presented was the version of the Lucene library that was being used versus the version that was used as reference material. In the latest version of Lucene, version 7.0, there was numerous changes to how the index was created and how the stemming process was done. This involved taking a method that took numerous variables and refactoring it so that it now took the expected variables as outlined in the method signature.

As seen above, there has been occasions that has led to refactoring the code so that it behaves as needed. With consultation of the official Lucene documentation, refactoring the code was much more intuitive and led to a greater understanding of the underlying application of the different built in methods and their uses. In figure 5, you can see that to create the index writer it was needed to define what the index and config were before passing them as arguments within the writer. In the previous versions of Lucene, like that in figure 4, it was possible to pass these all in as parameters directly to the writer without any adjustments. As for the current version of Lucene, the index writer only takes 2 arguments and the use of *AnalyzerUtils* highlighted above is no longer used and is now deprecated.

```
EnglishAnalyzer engAnalyse = new EnglishAnalyzer ();
// EnglishAnalyzer already has stemming built in:
Index = FSDirectory.open (INDEX_DIR2);
IndexWriterConfig config = new IndexWriterConfig(engAnalyse);

config.setOpenMode(IndexWriterConfig.OpenMode.CREATE);
Writer = new IndexWriter (index, config);
// set the analyser and config for the writer
```

Figure 5: Lucene 7.0

```
IndexWriter Writer = new IndexWriter (INDEX_DIR2, AnalyzerUtil.getPorterStemmerAnalyzer (new
StandardAnalyzer ()), writeToLuceneIndex.createNewIndex);
```

Figure 4: Lucene 3.1

4.2.2 Stemming:

Another matter that needed to be considered in the index creation was the method of stemming that was to be used. In the end the snowball stemming, an improved porter stemming method was used as it is the "de facto" stemming algorithm for stripping suffixes from words within information retrieval. Stemming refers to removing the suffixes from word that have the same root, such as "running" and "ran" to "run" as they all have the same meaning. There are some derivation rules used to generate the different root cases as seen on the porter stemming algorithm page[22] as mentioned earlier. Another consideration for classical stemming was the Lovins stemming approach. This is the first published stemmer written by Julie Beth Lovins in 1968 [23]. Other considerations included in this project were to use lemmatization or a stochastic approach to determine the right root for the given words. The stochastic approach to this problem was considered, however given the strict time scale for this project, it was ultimately omitted for consideration as to implement this effectively would have taken up an indeterminate amount of time and deviated too far from the primary objective of analyzing the different retrieval algorithms. The use of lemmatization was also strongly considered, as it would correctly reduce words to their root, known as a lemma. This however, involved constructing a corpus and a morphological analysis of the words and contextual analysis of the word, such as returning "saw" depending on the use of the word being as a verb or noun. This then was no longer a viable option within the scope of this project. For this project, the use of porter stemming came down to it having repeatedly shown to be empirically effective and its strong interrelationship with Lucene and the English language**.**

4.2.3 MMapDirectory Usage:

In creating the inverted index, there were many obstacles that presented themselves. One of them was the memory management of the index. As it stands, the current methods for handling how the virtual memory and the disk space is done by methods in the Lucene library called *FSDirectory* and *RAMDirectory*. These both are native to 32-bit machines and for creating the index of over one million documents, this did not suffice. There were errors thrown up mid creation of the index which indicated that these methods were not able to handle the memory management efficiently enough to bypass the overflow errors. This is where *MMapDirectory* was utilized to help overcome these errors [20]. With this method, managing the access of the index was much simpler. With the previous two methods, both had downfalls such as using the too much disk space being used, and the java garbage collector was then finding it difficult to release space from the heap. This in turn leads to latency in searching the enormous collection, sometimes leading to minutes for a single search. *MMapDirectory* access' the file system cache to avoid this problem. It also allows us to not allocate too much heap space to the program allowing for improved performance and avoid problems with the size of the index**.**

4.3 Querying and Results Set Generation:

4.3.1 Existing Query Approach Selection:

There were three algorithms that were used for querying the index. These were BM25, Vector Space Model (VSM) and Dirichlet Smoothing. These were chosen as they are different retrieval models and were an appropriate choice for this investigation. They were also used in CLEF eHealth information retrieval tasks.

[22] http://snowball.tartarus.org/algorithms/porter/stemmer.html

Vector Space Model is an algebraic model that is derived from linear algebra, determining each weight through a cosine similarity. BM25 is based off the bag of words retrieval function that ranks documents on the query terms appearing in each document. This model acts not on the ordering of the terms but the number of occurrences as shown here" the exact ordering of the terms in a document is ignored but the number of occurrences of each term is material" [pg.107, 12]. It is considered one of the better information retrieval models to use. Dirichlet Smoothing is a language-based model. It's a Bayesian Smoothing with special prior Dirichlet Distribution. These were used as they are very different retrieval algorithms. Vector Space Model (VSM) was used as a control algorithm and as such was expected to perform worse compared to the other two models. BM25 is a commonly used retrieval algorithm and has been shown to be a strong baseline algorithm in most cases. Lastly, Dirichlet Smoothing was used as it performs well when considering different languages as it is based off probability rather than the similarity used in the likes of Vector Space Model.

4.3.2 Querying and Results Generation:

To generate the results for this project, different runs were needed to be done for the different languages over the different models of retrieval as outlined above. Querying was done by taking the translated queries from the database, where they were stored and querying them against the inverted index that was created

```
    Directory Dir = new MMapDirectory(Point_To_Index);
    // the Lucene Index: Point_To_Index
    IndexReader reader = DirectoryReader.open(Dir);
    // open the index as readable
    IndexSearcher searcher = new IndexSearcher(reader);
    // searcher for the index.
    Analyzer analyzer = new EnglishAnalyzer();

    LMDirichletSimilarity similarity = new LMDirichletSimilarity();
    searcher.setSimilarity(similarity);// set the algorithm to use.

    QueryParser parser = new QueryParser(query.getFirst().getField(), analyzer);
    parser.setDefaultOperator(QueryParser.OR_OPERATOR);
    BooleanQuery b = new
    BooleanQuery.Builder().add(parser.parse(query.getFirst().getText()),
    BooleanClause.Occur.SHOULD).build();

TopScoreDocCollector collector = TopScoreDocCollector.create(100);
```

Figure 6: Querying the Index

from the million documents. Code files such as *SearchEventsContent_DirichletSmoothing* was used to open the index with the use of *MMapDirectory and* use *Indexsearcher* to allow it to be searchable. The Similarity comparison, which is used to decide on the different type of retrieval method you want to use, such as BM25 or Vector Space Model (VSM) was them specified. This was used with a *TopScoreDocCollector* to collect the relevant documents for this method, then on retrieval, it was stored in its own table that indicated the different method and language that was used, as well as normalized.

The results that were generated were held in a database. To do this, SQLite was used. This is a relational database management system (RDMS). This involved using SQLite and the JSoup[23] jar file within my java environment which allowed a connection between the database and the written source code that was writing the queries and the relevance scores to the database. This database held all the different queries for the different languages and the retrieval models used, accompanied with the relevance scores of each query. This database was then used to fetch the relevance scores of each query and with these, it was then possible to compute the different scores for Precision and Normal Discounted Cumulative Gain (NDCG) at 5 and 10 top documents respectively. Trec_Eval was used to do this as seen in the next section.

4.4 Results Analysis:

4.4.1 Trec_Eval Usage:

To generate the results sets for the different retrieval runs over the 8 distinct languages, a tool called Trec_Eval was used. Trec_Eval is the standard tool used in the Information retrieval space to generate the different results. Once this was successful, it was then possible to begin accessing the retrievals against the standard metrics, namely, precision at 5, 10, Normal discounted cumulative gain at 5, and 10. To do this, it was fundamental that the use of Trec_Eval was employed. To begin using this tool, there were a small amount of configuration that needed to be done. Firstly, this software is used and built for Unix systems, and as such is not window compatible out of the box. This meant that the use of Cygwin, a command line interface for Linux to use this software was needed. Using this, it was then possible to create an executable for windows that could be used on the windows operation system through Cygwin. Figure 7 below depicts a usual run to generate the NDCG results for Arabic using the BM25 algorithm. This is used with the *qrel.txt* file which contains the relevance scores for the different terms.

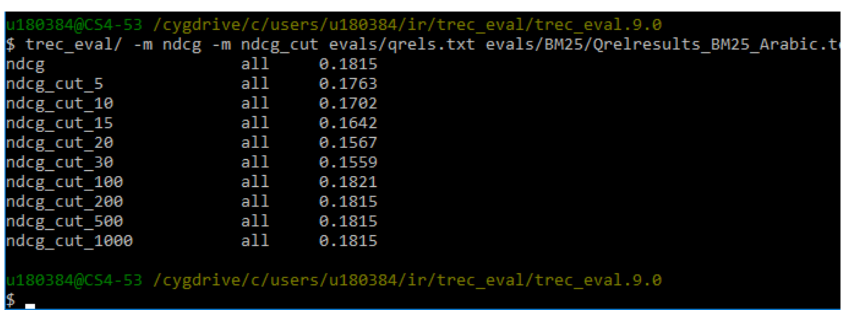

Figure 7: Trec_Eval Output

4.4.2 Metrics:

The metrics that were used to conduct the evaluation for the generated results were precision and normal discounted cumulative gain (NDCG) at top 5 and 10 documents respectively. Precision is the true positive over the true positive added to the true negative values. True positive in this respect is the number of documents that are deemed relevant to our retrieval and true negative is the documents that are relevant that were not recovered in the retrieval.

[23] https://jsoup.org/download

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} \quad (8)$$

Precision Formula

Normal Discounted Cumulative Gain (NDCG) is a measure of ranking quality. The premise of DCG is that highly relevant documents that appear low in a search result list are penalized as the graded relevance value is reduced logarithmically proportional to the position of the result. A logarithmic reduction factor is used to produce a smooth reduction. normalization across all queries is done by sorting all relevant documents in the collection by their relevance, producing the maximum possible DCG through position p. Normal Discounted Cumulative Gain is computed as seen in figure below.

$$\text{nDCG}_\text{p} = \frac{\text{DCG}_\text{p}}{\text{IDCG}_\text{p}} \quad (9)$$

Normal Discounted Cumulative Gain Formula

4.5 Validation:

4.5.1 Viewing the Index:

Once this was done, another standard Information retrieval tool was used, called Lukeall[24]. This is a tool used to view and query an index. This was used to validate that the index had been create correctly without corruption or error. This was also used with a smaller index as a proof of concept and code validation to ensure that the index and retrieval was correct**.** With this tool, showing that the index had been created and that the stemming process had been a success was achieved.

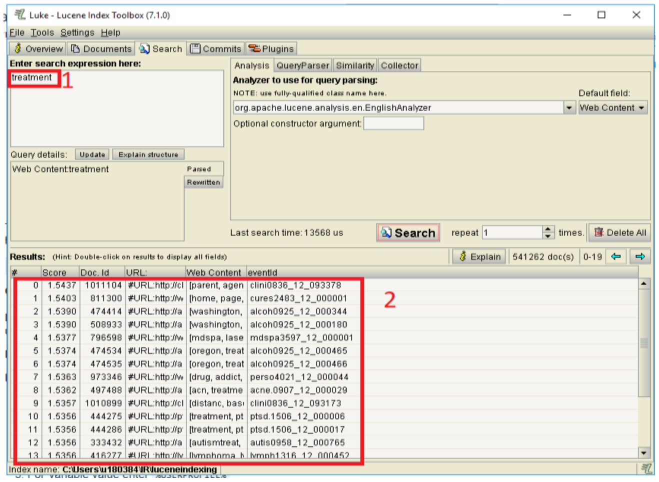

Figure 10: Lukeall Index Viewing

Figure 10, indicated by the number 1 shows that with this tool it is possible to view an index that has been created and search it for different query terms. This tool allowed to search for any term in this way. Figure 10, indicated by the number 2 shows the documents that have been retrieved when searching for the term "*treatment*". This screen then also allows the validation that the stemming process has been successful, and

[24] https://github.com/DmitryKey/luke/releases

that the creation of the index has been successful. This can be seen in the figure in the search results with the word "Distance" which has dropped the "e" as part of this process.

Chapter 5: Evaluation

For evaluation of the investigation of cross lingual medical information retrieval systems' low benchmarks, the Wilcoxon signed-rank test was used. This is a non-parametric statistical hypothesis test used to compare related samples. "It isn't possible to make such a distinction for an individual query, but when evaluation measures are averaged over a number of queries, one can obtain an estimate of the error associated with that measure and significance tests become applicable." [24, pg332]. To run these different tests, The Programming Language R[25] was used. This was used to perform the different Wilcox tests on the different sets of queries, to find if there is any statistical significance to the results. The use of the null hypothesis is used in this measure. The null Hypothesis states that there is no relationship between two measured association between two group. If the hypothesis is rejected, that is the p-value is less than 0.05, there is grounds to believe that the two measures are related in some way. From the figures in table 2, the Asterix shows the entries that have a P-Value of less than 0.05, and thus these are believed to be statistically significant and there is a relationship between these two measures. This was done for comparison against Vector Space Model (VSM), Best Match 25 (BM25) and Best Match 25 with Dirichlet Smoothing.

Table 1: VSM BM25 Significance Testing

Language:	NDCG@5	NDCG@10	P@5	P@10
English	0.00002596*	0.0001051*	0.00009201*	0.0006419*
German	0.00004859*	0.00007538*	0.00009608*	0.0006228*
French	0.0001087*	0.00003883*	0.0001254*	0.00033*
Italian	0.004072*	0.0009673*	0.001466*	0.002273*
Portuguese	0.0000876*	0.00003539*	0.000581*	0.0001982*
Persian	0.002788*	0.0003151*	0.02707*	0.001626*
Arabic	0.0005222*	0.0003409*	0.0002715*	0.0006107*
Czech	0.0003174*	0.0003174*	0.000546*	0.005044*

Figure 11 depicts Vector Space Model (VSM), Best Match 25 (BM25) and Dirichlet Smoothing ran against each metric. Visually, the Dirichlet Smoothing approach is far greater than both VSM and BM25 in all metrics, this is true for seven out of the eight languages used, except for the Portuguese language where BM25 out performs Dirichlet Smoothing in one metric, Precision at 5 (P@5). This however, can be neglected as with Precision at 10 (P@10), Dirichlet Smoothing out performs BM25.

[25] https://www.r-project.org/

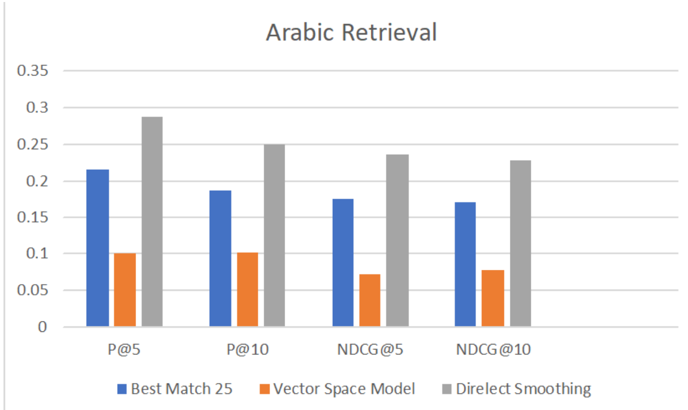

Figure 11: Arabic Retrieval Results

Table 2: BM25 Dirichlet Smoothing Significance Testing

Language:	NDCG@5	NDCG@10	P@5	P@10
English	0.1352	0.009701*	0.03268*	0.006253*
French	0.3654	0.2651	0.5865	0.3492
German	0.2642	0.02866*	0.1932	0.007999*
Italian	0.1039	0.05271	0.141	0.09404
Persian	0.02992*	0.04502*	0.02688*	0.2144
Portuguese	0.9624	0.8346	0.8879	0.8034
Arabic	0.02384*	0.003595*	0.01381*	0.003324*
Czech	0.3142	0.003856*	0.05724	0.0006458*

Most notable in Table 2, which depicts the testing done for BM25 and Dirichlet Smoothing, is that the Arabic language queries were rejected by the Null Hypothesis on all metrics, indicating that these two algorithms are not the same. A two-tailed test was conducted on this sample and found that Dirichlet Smoothing was the stronger of the two algorithms used. It is also notable that Italian, Portuguese and French have proved to not be statistically significant in this test. Overall, employing the use of the language-based model of Dirichlet Smoothing has proved to be significantly stronger than that of Vector Space Model (VSM) and marginally stronger than the commonly used probabilistic model of Okapi's Best Match 25 (BM25).

Table 3: Dirichlet Smoothing Retrieval Score

Language:	P@5	P@10	NDCG@5	NDCG@10
Arabic	0.2879	0.2500	0.2362	0.2287
Czech	0.3515	0.3121	0.2688	0.2665
English	0.3879	0.3470	0.2936	0.2933
French	0.3333	0.2985	0.2612	0.2564
German	0.2909	0.2591	0.2412	0.2342
Italian	0.2879	0.2621	0.2276	0.2275
Persian	0.2697	0.2348	0.2313	0.2173
Portuguese	0.2909	0.2682	0.2358	0.2344

Table 3 depicts the retrieval scoring for the Dirichlet Smoothing model and table 4 Vector Space Model (VSM). Clearly, the queries that did not need to be translated into English performed best. This may be due to translation issues when using the Google Translate API. These scores could be improved by using other translation software that are not freely available, such as PROMT[26]. There is a considerable difference in the retrieval scoring between Vector Space Model (VSM) and Dirichlet Smoothing, exemplifying that Vector Space Model is not a strong baseline for information retrieval runs, as with the English Language Dirichlet Smoothing is over two times better in Precision over the top 5 documents when translation is not a threat to the retrieval. Over all the other languages and metrics, Dirichlet Smoothing Provides a strong case for a strong baseline algorithm and evidence that Vector Space Model (VSM) is not an Information Retrieval Model that should be implemented when retrieving cross-lingual medical information.

Table 4: Vector Space Model Retrieval Score

Language:	P@5	P@10	NDCG@5	NDCG@10
Arabic	0.1000	0.1015	0.0723	0.0782
Czech	0.1723	0.1554	0.1243	0.1227
English	0.1758	0.1773	0.1228	0.1352
French	0.1662	0.1677	0.1145	0.1238
German	0.1242	0.1182	0.0938	0.0976
Italian	0.1385	0.1385	0.1033	0.1109
Persian	0.1292	0.1231	0.0953	0.0980
Portuguese	0.1877	0.1554	0.1236	0.1210

As shown in table 2, there are only a choice few case where the results are statistically significant. This highlights that Best Match 25 (BM25) is a relatively strong baseline algorithm to use over most languages, with the exception of the Persian and Arabic Language where most metrics were proven to be statistically significant, resulting in evidence of Dirichlet Smoothing being the better choice of baseline algorithm when testing known and newly hypothesized algorithms in the Information Retrieval Space.

Table 5: BM25 Retrieval Score

Language:	P@5	P@10	NDCG@5	NDCG@10
Arabic	0.2152	0.1864	0.1763	0.1702
Czech	0.2954	0.2385	0.2237	0.2076
English	0.3273	0.2909	0.2534	0.2523
French	0.3046	0.2738	0.2334	0.2302
German	0.2646	0.2045	0.2174	0.1957
Italian	0.2431	0.2185	0.1928	0.1884
Persian	0.2092	0.2046	0.1718	0.1788
Portuguese	0.2985	0.2631	0.2349	0.2280

[26] http://www.promt.com/

Chapter 5: Conclusion

5.1: Summary

In this project, the aim was to investigate the different benchmarks for medical information retrieval over a cross lingual domain. Investigation was done on 8 different languages such as Persian and Czech. Three different algorithms were used for the retrieval process and 4 different metrics were used to gather the results. Statistical analysis was done on the result sets that were generated and an evaluation on cross lingual information retrieval benchmarks was then produced. This Project has provided a state of the art contribution to the domain of Medical Information Retrieval as it provides analysis and details on strong baselines for Cross Lingual Information Retrieval (CILR) systems that did not exist before this, to the best of my knowledge.

5.2: Critical Analysis

5.2.1: Google Translate

While Google translate might be considered a strong baseline translation system, in this project, short layperson medical queries were used. This could potentially cause a problem for the translation process since queries that were used are specific to the medical domain. Short keywords make it difficult for translation systems to get the right context for the translation.

queryNum:

```
clef2015.test.61
```

queryTerms:

```
Muddy blood under the nail
```

Figure 12: Persian Query

Figure 12 shows that the google translation process is potentially liable to different errors. This query in English is “Hematoma” and as such does not replicate the original request. Further work could explore the use of different translation technologies such as Moses[27] which is used in this paper [25] to looks at this issue. Despite this, google translate was used as it is a good standard translation engine which is freely available to use. Other machine translation systems that were looked at but omitted for this project because they were not easily available were PROMT[28], SYSTRAN[29] and Omniscien Technologies[30].

---

[27] http://www.statmt.org/moses/
[28] http://www.promt.com/
[29] http://www.systransoft.com/lp/machine-translation/
[30] https://omniscien.com/

5.2: Results case

These results are based on medical information retrieval run using the 2025 CLEF eHealth task 2. The results show that the use of relevant baselines for Cross Lingual Information Retrieval (CLIR) is necessary as the use of weak baselines inhibits the performance of the retrieval. From the results gathered, in most cases the Dirichlet smoothing approach provides an improvement in retrieval as mentioned in the tables above, except for the Portuguese language retrieval run. This has provided statistical significance in retrieval as outlined with the Wilcox Signed Rank Test, indicating that there is reason to believe that the use of Dirichlet Smoothing is a better retrieval model. as this is a language-based model we can conclude that how it ranks the corpus of words in queries works better for Cross-Lingual Medical Information Retrieval (CLIR) than the use of a probabilistic model such as Okapi's Best Match 25 (BM25) in the laypeople's medical queries.

5.3: Threats to Validation

Some of the treats to the validation of this report have been mentioned. This includes the use of Google Translate while it may not be considered the best technology for Machine Translation (MT). This in turn could have influenced the retrieval for scores of the different queries, which in turn could produce worse scores than typically expected when using a strong retrieval algorithm, such as BM25.

5.4: Future Works

Reflecting on the work that has been done in this project, there are some observations and considerations if given a lengthier time scale for this project. One of the considerations that was found that could improve the project was the use of statistical machine translation (SMT). Documents are translated according to a probability distribution from a target language from a source language. The benefits of this is that it creates more fluent translations which would improve the retrievals for language-based models such as Direct Smoothing and Witten Bell Smoothing.

Another aspect that could have been addressed in this project given more time would be the use of more complex retrieval algorithms. There are different methods such interpolation and back-off to adjust the probability of seen and unseen events, such as that used in the Dirichlet Smoothing approach to Information retrieval. This however, was not extensively used as the time estimation for the project scope was too small to include such complexity. As stated by Chen, "However, this approach requires a large number of interpolation weight parameters to be robustly estimated and therefore leads to a severe data sparsity problem when given limited data" [23, pg183] highlighting that knowledge of the data used is important for these complex algorithms.

One more aspect that would have been addressed would be using more statistical significance testing methods to ensure that the results are within a strong confidence threshold. Other tests that should have been used to give more confidence in the statistical significance of the results were the Paired T-test and the sign test. Use of these would allow for a stronger confidence in the results and with these, a greater insight to the results and discrepancies in the data would become more obvious. As stated by Hull, "What if the test is not significant? This does not necessarily mean that there is no difference between the methods, merely that the test was unable to detect one." [pg.24, 26] showing that the use of more than one test is necessary to be confident in the findings.

From completing this project, considerations in researching within the domain of medical information retrieval and information theory have been significantly strengthened and has led to a more meaningful knowledge of the extent of the computer science spectrum. Further work on this project is a very strong possibility as the document index and architecture is understood and already created. Consideration of different models of information retrieval to research what in natural language processing are the defining features of a strong and weak retrievals and how to use these to improve existing algorithms in information retrieval. Research into the corpus of words that different people of different languages would be studied to try to refine existing retrieval algorithms term weights to querying and consider these behaviours. Then use to more accurately retrieve information in different languages. Zips law[31] would come into play and investigation of why given a large sample of words used, the frequency of any word is inversely proportional to its rank in the frequency table and how this could be used to improve how we approach these problems in information retrieval.

[31] https://simple.wikipedia.org/wiki/Zipf%27s_law

References:

1. . Fox, S.: Health topics: 80% of internet users look for health information online. Pew Internet & American Life Project (2011)

2. Hanbury, Allan and Müller, Henning (2013) Khresmoi – multilingual semantic search of medical text and images, MedInfo 2013, August 2013, Copenhagen, Denmark.

3. C. (n.d.). This website aims at gathering all the information related to past, current and future editions of the CLEF eHealth Evaluation Lab. Retrieved March 09, 2018, from https://sites.google.com/site/clefehealth/

4. Kelly, L., Goeuriot, L., Suominen, H., Névéol, A., Palotti, J., & Zuccon, G. (2016). Overview of the CLEF eHealth Evaluation Lab 2016. Lecture Notes in Computer Science Experimental IR Meets Multilinguality, Multimodality, and Interaction, 255-266. doi:10.1007/978-3-319-44564-9_24

5. Lorraine Goeuriot, Liadh Kelly, Leif Hanlen, Hanna Suominen, Aur elie N ev eol, Joao Palotti, and Guido Zuccon. Overview of the CLEF eHealth Evaluation Lab 2015. In Proceedings of CLEF 2015, 2015.

6. Liadh Kelly, Lorraine Goeuriot, Hanna Suominen, Aur elie N ev eol, Joao Palotti, and Guido Zuccon. Overview of the CLEF eHealth Evaluation Lab 2016. In Proceedings of CLEF 2016, 2016.

7. Liadh Kelly, Lorraine Goeuriot, Hanna Suominen, Tobias Schreck, Gondy Leroy, Danielle L. Mowery, Sumithra Velupillai, Wendy W. Chapman, Guido Zuccon, and Joao Palotti. Overview of the ShARe/CLEF eHealth Evaluation Lab 2014. In Proceedings of CLEF 2014, 2014.

8. Hanna Suominen, Sanna Salanter a, Sumithra Velupillai, Wendy W. Chapman, Guergana Savova, Noemie Elhadad, Danielle Mowery, Johannes Leveling, Lorraine Goeuriot, Liadh Kelly, David Martinez, and Guido Zuccon. ShARe/CLEF eHealth Evaluation Lab 2013: Three shared tasks on natural language processing and machine learning to make clinical reports easier to understand for patients. In CLEF 2013, Lecture Notes in Computer Science (LNCS). Springer, 2013.

9. Task 2: User-Centred Health Information Retrieval - CLEF eHealth 2015. (2015). Retrieved March 09, 2018, from https://sites.google.com/site/clefehealth2015/task-2

10. Leveling, J. (2006). A Baseline for NLP in Domain-Specific IR. Accessing Multilingual Information Repositories Lecture Notes in Computer Science, 222-225. doi:10.1007/11878773_26

11. Koopman, B., & Palotti, J. (n.d.). Diagnose this if you can. In G. Zuccon (Author), Advances in information retrieval (pp. 562-567). 2015. doi:10.1007/978-3-319-16354-3

12. Manning, C. D., Raghavan, P., & Schütze, H. (2017). Introduction to information retrieval. Delhi: Cambridge University Press.

13. D. R. H. Miller, T. Leek, and R. M. Schwartz. BBN at TREC7: Using hidden markov models for information retrieval. In E. M. Voorhees and D. K. Harman, editors, The Seventh Text REtrieval Conference (TREC-7). Department of Commerce, National Institute of Standards and Technology, 1998. http: //trec.nist.gov/pubs/trec7/t7 proceedings.html

14. Büttcher Stefan, et al. Information Retrieval: Implementing and Evaluating Search Engines. Mit Press, 2016.

15. Conlan, O., & Staikopoulos, A. (2017). 2nd International EvalUMAP Workshop (EvalUMAP2017) Chairs Preface & Organization. Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization - UMAP 17. doi:10.1145/3099023.3099041

16. Van Rijsbergen, C. (n.d.). *Information Retrieval*.

17. Koopman, B., Bruza, P., Sitbon, L., & Lawley, M. (2011). Evaluating medical information retrieval. Proceedings of the 34th international ACM SIGIR conference on Research and development in Information - SIGIR 11. doi:10.1145/2009916.2010088

18. Pedersen, T., Pakhomov, S. V., Patwardhan, S., & Chute, C. G. (2007). Measures of semantic similarity and relatedness in the biomedical domain. Journal of Biomedical Informatics, 40(3), 288-299. doi:10.1016/j.jbi.2006.06.004

19. Lucene Text File Indexer. (n.d.). Retrieved March 06, 2018, from http://www.lucenetutorial.com/sample-apps/textfileindexer-java.html

20. Schindler, U. (1970, January 01). The Generics Policeman Blog. Retrieved March 06, 2018, from http://blog.thetaphi.de/2012/07/use-lucenes-mmapdirectory-on-64bit.html

21. Zhou, D., Truran, M., Brailsford, T., Wade, V., & Ashman, H. (2012). Translation techniques in cross-language information retrieval. ACM Computing Surveys, 44(2), 17-18. doi:10.1145/2089125

22. Lovins, Julie Beth (1968). "Development of a Stemming Algorithm". Mechanical Translation and Computational Linguistics. 11: 22–31.

23. Chen, X., Liu, X., Gales, M. J., & Woodland, P. C. (2015). Investigation of back-off based interpolation between recurrent neural network and n-gram language models. 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 181-186. doi:10.1109/asru.2015.7404792

24. Hull, D. (1993). Using statistical testing in the evaluation of retrieval experiments. Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR 93, 329-328. doi:10.1145/160688.160758

25. Pecina, Pavel and Dušek, Ondřej and Goeuriot, Lorraine and Hajič, Jan and Hlaváčová, Jaroslava and Jones, Gareth J.F. and Kelly, Liadh and Leveling, Johannes and Mareček, David and Novák, Michal and Popel, Martin and Rosa, Rudolf and Tamchyna, Aleš and Urešová, Zdeňka (2014) Adaptation of machine translation for multilingual information retrieval in the medical domain. Artificial Intelligence in Medicine, Elsevier, February 2014.

Appendices:

Retrieval Graphs:

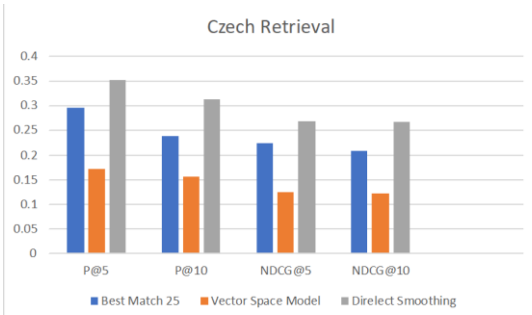

Figure 1: Arabic Retrieval

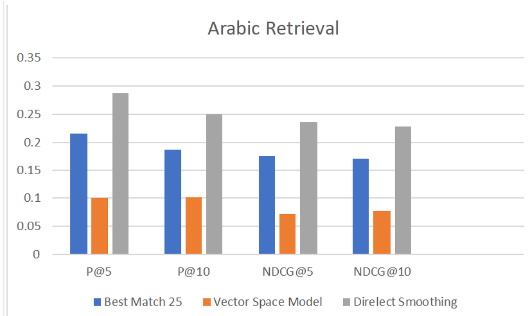

Figure 2: Czech Retrieval

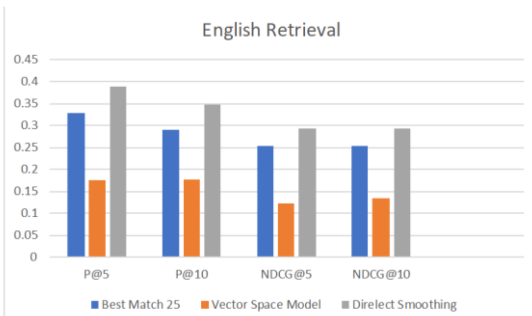

Figure 1: English Retrieval

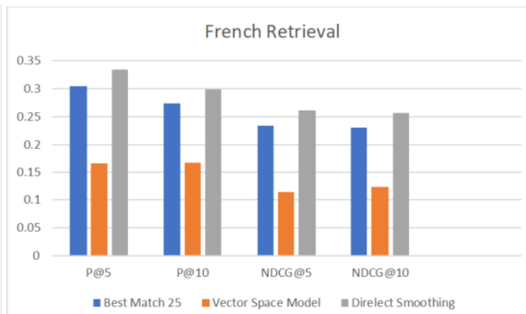

Figure 4: French Retrieval

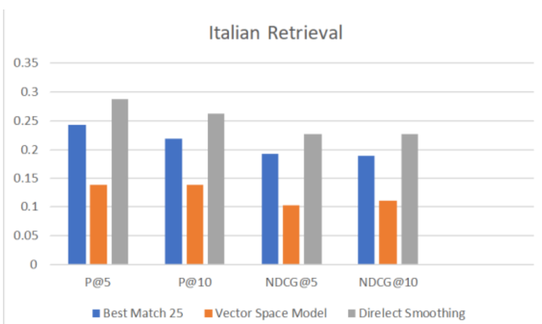

Figure 5: Italian Retrieval

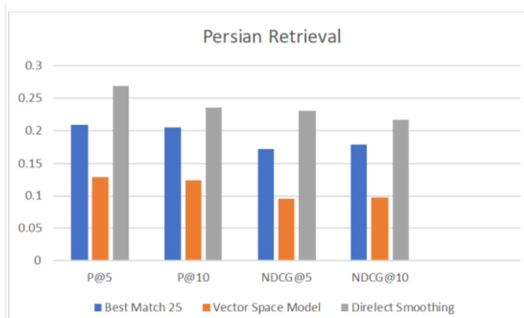

Figure 6: Persian Retrieval

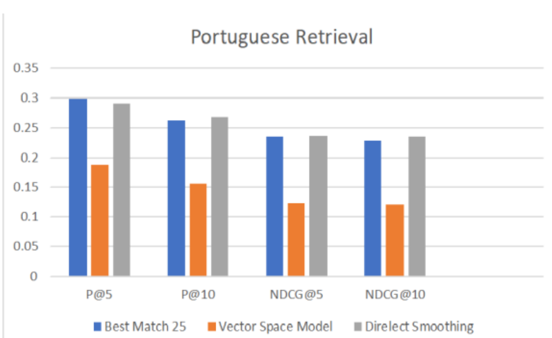

Figure 7: Portuguese Retrieval

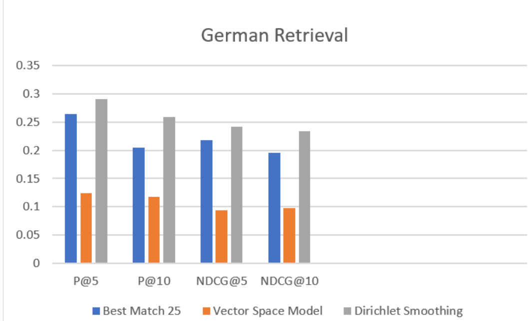

Figure 8: German Retrieval

Translations of Sample Queries for Different Languages:

Arabic:

<top>
<num>clef2015.test.1</num>
<query>علامات حمراء عديدة على الساقين بعد السفر من عندنا</query>
</top>

Translation:

Clef test Number: clef2015.test.1
Query: Many red marks on the legs after our travel

Czech:

<top>
<num>clef2015.test.1</num>
<query>mnohočetné červené skvrny na dolních končetinách po cestě z USA</query>
</top>

Translation:

Clef test Number: clef2015.test.1
Query: multiple red spots on the lower limbs along the way from the USA

German:

<top>
<num>clef2015.test.1</num>
<query>viele rote Flecken an den Beinen nach US Reise</query>
</top>

Translation:

Clef test Number: clef2015.test.1
Query: many red spots on the legs after US Travel

English:

<top>
<num>clef2015.test.1</num>
<query>many red marks on legs after traveling from us</query>
</top>

Translation:

Clef test Number: clef2015.test.1
Query: many red marks on legs after traveling from us

Persian:

<top>
<num>clef2015.test.1</num>
<query> علائم قرمز رنگ بسیار زیاد بر روی پاها بعد از رفتن از پیش ما </query>
</top>

Translation:

Clef test Number: clef2015.test.1
Query: Extremely high red marks on the legs after going past us.

French:

<top>
<num>clef2015.test.1</num>
<query>nombreuses taches rouges sur les jambes après un voyage des États-Unis</query>
</top>

Translation:

Clef test Number: clef2015.test.1
Query: many red spots on the legs after a US trip

Italian:

<top>
<num>clef2015.test.1</num>
<query>numerose macchie rosse sulle gambe dopo un viaggio negli Stati Uniti</query>
</top>

Translation:

Clef test Number: clef2015.test.1
Query: numerous red spots on the legs after a trip to the United States

Portuguese:

<top>
<num>clef2015.test.1</num>
<query>muitas marcas vermelhas nas pernas depois de viajar dos estados unidos</query>
</top>

Translation:

Clef test Number: clef2015.test.1
Query: many red marks on legs after traveling from the United States

Retrieval Score over Choice Language and Metrics:

Table 1: BM25 Retrieval Score

Language:	P@5	P@10	NDCG@5	NDCG@10
Arabic	0.2152	0.1864	0.1763	0.1702
Czech	0.2954	0.2385	0.2237	0.2076
English	0.3273	0.2909	0.2534	0.2523
French	0.3046	0.2738	0.2334	0.2302
German	0.2646	0.2045	0.2174	0.1957
Italian	0.2431	0.2185	0.1928	0.1884
Persian	0.2092	0.2046	0.1718	0.1788
Portuguese	0.2985	0.2631	0.2349	0.2280

Table 2: Vector Space Model Retrieval Score

Language:	P@5	P@10	NDCG@5	NDCG@10
Arabic	0.1000	0.1015	0.0723	0.0782
Czech	0.1723	0.1554	0.1243	0.1227
English	0.1758	0.1773	0.1228	0.1352
French	0.1662	0.1677	0.1145	0.1238
German	0.1242	0.1182	0.0938	0.0976
Italian	0.1385	0.1385	0.1033	0.1109
Persian	0.1292	0.1231	0.0953	0.0980
Portuguese	0.1877	0.1554	0.1236	0.1210

Table 3: Dirichlet Smoothing Retrieval Score

Language:	P@5	P@10	NDCG@5	NDCG@10
Arabic	0.2879	0.2500	0.2362	0.2287
Czech	0.3515	0.3121	0.2688	0.2665
English	0.3879	0.3470	0.2936	0.2933
French	0.3333	0.2985	0.2612	0.2564
German	0.2909	0.2591	0.2412	0.2342
Italian	0.2879	0.2621	0.2276	0.2275
Persian	0.2697	0.2348	0.2313	0.2173
Portuguese	0.2909	0.2682	0.2358	0.2344

Significance testing for retrievals:

Table 4: BM25 Dirichlet Smoothing Significance Testing

Language:	NDCG@5	NDCG@10	P@5	P@10
English	0.1352	0.009701*	0.03268*	0.006253*
French	0.3654	0.2651	0.5865	0.3492
German	0.2642	0.02866*	0.1932	0.007999*
Italian	0.1039	0.05271	0.141	0.09404
Persian	0.02992*	0.04502*	0.02688*	0.2144
Portuguese	0.9624	0.8346	0.8879	0.8034
Arabic	0.02384*	0.003595*	0.01381*	0.003324*
Czech	0.3142	0.003856*	0.05724	0.0006458*

Table 5: VSM BM25 Significance Testing

Language:	NDCG@5	NDCG@10	P@5	P@10
English	0.00002596*	0.0001051*	0.00009201*	0.0006419*
German	0.00004859*	0.00007538*	0.00009608*	0.0006228*
French	0.0001087*	0.00003883*	0.0001254*	0.00033*
Italian	0.004072*	0.0009673*	0.001466*	0.002273*
Portuguese	0.0000876*	0.00003539*	0.000581*	0.0001982*
Persian	0.002788*	0.0003151*	0.02707*	0.001626*
Arabic	0.0005222*	0.0003409*	0.0002715*	0.0006107*
Czech	0.0003174*	0.0003174*	0.000546*	0.005044*

Sample Query:

queryNum:

```
clef2015.test.61
```

queryTerms:

```
Muddy blood under the nail
```

Figure 12: Persian Query