THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

Enhanced Search, Interaction and Visualisation of Free-Text Electronic Health Records

by

Ivan Litvinov

School of Information Technology and Electrical

Engineering University of Queensland

Submitted for the degree of Master of Information Technology

June 2019

10 June 2019

Prof Michael Brünig
Head of School
School of Information Technology and Electrical Engineering
The University of Queensland
St Lucia QLD 4072

Dear Professor Brünig,

In accordance with the requirements of the Degree of Master of Information Technology in the School of Information Technology and Electrical Engineering, I submit the following thesis entitled

“Enhanced Search, Interaction and Visualisation of Free-Text Electronic Health Records”

The thesis was performed under the supervision of Dr Anthony Nguyen. I declare that the work submitted in the thesis is my own, except as acknowledged in the text and footnotes, and that it has not previously been submitted for a degree at the University of Queensland or any other institution.

Yours sincerely,

____

Ivan Litvinov

Acknowledgments

I am grateful to my supervisor Dr Anthony Ngoc Nguyen for accepting me to work on this thesis. His no-nonsense approach, prompt feedback and supportive guidance were greatly appreciated as they made this a very manageable experience. Also, assistance by David Conlan and Dr Bevan Koopman at CSIRO helped me in overcoming some technical hiccups along the way.

Abstract

With the abundance of medical records and journals being stored electronically, clinicians and medical researchers often need to extract information from that vast sea of data to help their patients or to advance their research. However, a lot of these resources are stored as plain text and searching for useful information in medical plain text by regular keyword searching is hampered by a semantic gap: there is a difference between the raw data contained in a medical document and the way a medical professional may interpret that data [1]. The search results are either missing the relevant records due to vocabulary and granularity mismatch or contain many irrelevant records with false positives and partial matches of the search query.

Researches and natural language processing and information retrieval professionals have been developing systems to overcome these challenges for many years with increasing success. One of the methods employed by the systems in overcoming the semantic gap is to use medical ontologies and match medical terms in plain text to concept identifiers (IDs) and to conduct searches for concept IDs instead of text. The problem with a lot of the systems developed, however, is that they are purpose-built for information retrieval competitions or research and do not have a user interface that a regular medical practitioner can understand and use on a daily basis.

What clinicians require is a sophisticated search system that overcomes the semantic gap challenges but is still simple to use, like the familiar keyword-based search interfaces. In addition to dealing with semantic gap issues it may be beneficial for a search system to provide a level of visual representation and interactivity that allows clinicians to easily gauge their search results and refine them further using interactive elements [3].

This thesis presents the Enhanced Search System (ESS). It is designed to make searching plain-text EHRs effective by attempting to overcome the vocabulary and granularity mismatch challenges of the semantic gap while providing a simple user interface, useful interactive visualisations of the search results, and intelligent highlighting of searched terms within the results. ESS's core method of overcoming

the semantic gap challenges is to extract medical concepts from EHRs and user search queries, and to search medical concept identifiers instead of text.

Contents

Acknowledgments **iii**

Abstract **iv**

Contents **vi**

List of Figures **vii**

List of Tables **viii**

1 Introduction **1**

2 Background **4**

3 Methods **7**

3.1 Concept extraction 7

3.2 Creation and indexing of annotated documents 11

3.3 Searching for concept IDs instead of text to tackle the semantic gap challenges 12

3.4 Intelligent highlighting of searched terms 14

3.5 Interactive visualisation of search results 15

3.6 The search interface - flexible search options 18

3.7 Summary of the resulting system architecture 20

3.7.1 Stage 1 - concept extraction and indexing 20

3.7.2 Stage 2 - the search interface 20

4 Results and discussion **21**

5 Conclusion **25**

Bibliography **26**

List of Figures

Figure 3.1: Example of marked-up text with embeddings of concept IDs 11

Figure 3.2: Example of how Elasticsearch annotated text highlighter indicates a search hit 14

Figure 3.3: Screenshot of ESS highlighting spans of text that match the query 15

Figure 3.4: Screenshot of the user interface of ESS.. 17

Figure 3.5: Screenshot of the Ontoserver concept look-up and autocomplete feature 19

List of Tables

Table 3.1: Comparison of concept extraction systems QuickUMLS and Bioportal 9

Table 3.2: Semantic types used in ESS during concept extraction........................... 10

Table 3.3: Example of concept extraction from text performed by QuickUMLS........ 10

Table 4.1: Comparison of sample search results using three search methods......... 23

Chapter 1

Introduction

With the advances in information technology and ubiquitous use of electronic devices in all areas of our lives, more and more information is recorded and stored electronically. Such is the case in the medical field. Patient notes, test and examination results, discharge summaries and so on are recorded as electronic health records (EHRs). A lot of these records are created in an unstructured plain-text form. These EHRs are a useful source of information for medical professionals to help with clinical decision-making and to advance research in the medical field. Naturally, a search system is needed to find the relevant information. But a regular keyword-based free text search is not very effective. The reason is that the health search is complicated by a semantic gap problem: there is a difference between the raw data contained in a medical document and the way a medical professional may interpret that data [1].

Semantic gap can be represented by several challenges [2]. The main challenge is vocabulary mismatch. A keyword search looking for records with “heart attack” will not bring up records containing synonyms like myocardial infarction or cardiac infarction. All the synonyms will need to be entered into the search box to find all the relevant records, which is cumbersome, time-consuming, and inconvenient.

The next challenge, very relevant to the medical domain, is granularity (generalisation/specialisation) mismatch. A keyword search for a type of medication will not bring up records containing specific brand names of that medication if the overarching type-of-medication keyword was not specifically mentioned in the record. Another example of granularity mismatch is when a disorder has many sub-concepts that specify different variations of that disorder based on which body part or type of cells is affected by it. In the case of cancer, there are more than 100 sub-types of cancer and a lot of them don’t have the word “cancer” as part of their name (e.g. lymphoma, carcinoma, melanoma). The total number of sub-concepts of cancer is more than 5000! Manually typing in all these terms into the search field to get all the relevant records is simply impractical.

Another challenge, specific to medical search, is the challenge of negation recorded in medical records. The record may say “no fever”, but a search for “fever” will bring up that record as a match. There are a few other semantic gap challenges and all of them make searching plain-text EHRs using a regular term-based search system quite ineffective. The search results are either missing the relevant records due to vocabulary and granularity mismatch or contain many irrelevant records with false positives and partial matches of the search query.

Researches, natural language processing and information retrieval professionals have been developing systems to overcome these challenges for many years with increasing success. One of the main development and testing grounds for these systems is the Text REtrieval Conference (TREC), that’s been running different medical search challenges (MedTrack) since 2011 [2, 4, 5]. One of the methods employed by the systems in overcoming the semantic gap is to use medical ontologies and match medical terms in plain text to concept identifiers (IDs) and to conduct searches for concept IDs instead of text. The problem with the systems developed at TREC, however, is that they are purpose-built for the challenges and do not have a user interface that a regular medical practitioner can understand and use on a daily basis.

What clinicians require is a sophisticated search system that overcomes the semantic gap challenges but is still simple to use, like the familiar keyword-based search interfaces. In addition to dealing with semantic gap issues it may be beneficial for a search system to provide a level of visual representation and interactivity that allows clinicians to easily gauge their search results and refine them further using interactive elements [3].

This thesis presents the Enhanced Search System (ESS). It is designed to make searching plain-text EHRs effective by attempting to overcome vocabulary mismatch and granularity mismatch challenges of the semantic gap while providing a simple user interface, useful interactive visualisations of the search results, and intelligent highlighting of searched terms within the results. ESS’s core method of overcoming the semantic gap challenges is to extract medical concepts from EHRs and user search queries, and to search medical concept identifiers instead of text. The following chapters will provide background material on medical search, go into the details of the methods employed by the ESS and provide an overview of its user interface and system architecture. The results of performing searches and how the semantic gap challenges are tackled are presented in the Results and discussion chapter. Finally, the contribution of the system and possible future work is discussed in the Conclusion chapter.

Chapter 2
Background

With the abundance of medical records and journals being stored electronically, clinicians and medical researchers often need to extract information from that vast sea of data to help their patients or to advance their research. Clinicians may be trying to determine the diagnosis for a patient's symptoms, the best treatment for the illness or what test a patient should do. The researchers may need to find suitable cohort for a clinical trial, match a clinical trial to a patient, or identify suitable literature for a systematic review [4]. All these tasks require conducting a medical search, which is hampered by the semantic gap.

Koopman, in his thesis [2], explores the many aspects of semantic gap in medical search. He identifies that searching EHRs and medical literature is inefficient with standard keyword-matching search engines due to vocabulary mismatch, granularity, conceptual implication, inferences of similarity, and challenges specific to medical domain - negation and family history, temporality, age and gender, and levels of evidence. A certain level of inference is required to overcome them, which the author explores with his semantic search systems.

The first common step in dealing with semantic gap is using medical concept ontologies or thesauri to match terms in queries and searched documents to medical concepts [2]. This handles the vocabulary mismatch problem by treating all the synonyms of a medical term as one concept. The most commonly used ontologies are Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT) [6] and Unified Medical Language System (UMLS) [7].

The use of medical thesauri and ontologies like SNOMED CT is increasing in the medical field in order to, among other reasons, put more meaning into EHRs and to minimise the semantic gap. This led to development of ontology tools like Ontoserver by CSIRO [REF] and Bioportal by NCBO [REF] that provide medical ontology-related services like concept look-up and in case of Bioportal, extraction of medical concepts from plain text.

Ontologies are also utilised to deal with the challenge of granularity, where higher level concepts are mapped to its subclass variants or child concepts. The original search query is then expanded, with child concepts also being searched in the documents [2].

The semantic gap issues of negation and family history were investigated further by Koopman and Zuccon [8]. They built on previous research and developed a system to deal with both and evaluated their findings. Negated concepts simply used to be removed from documents in medical search systems, but the authors found that such removal could actually worsen search performance. Assigning different weights to negated concepts proved to be more effective. Family history on the other hand had an insignificant effect. Kuhn and Eickhoff [9] developed their own system of dealing with negated concepts, which also outperformed the negation-removal method.

Because searching medical plan text is a non-trivial task and has significant importance, the Text REtrieval Conference (TREC) has been running different medical search challenges (MedTrack). Teams from different research organisations and institutions attempt to create a system that delivers the best search results for the challenge. Some of the tasks the systems had to perform in the challenges are as follows:

- Extract concepts from EHRs
- Find patients with a certain condition
- Determine diagnosis, test or treatment from the EHR
- Match a patient to a suitable clinical trial

Nguyen et al. [5] provide an overview of the systems and summary of common system components used by teams competing at the 2016 MedTrack challenge. Along with the use of UMLS and systems matching text to medical concepts (annotators or concept extraction systems), competitors used other approaches and tools like Medical Subject Headings (MeSH, a thesaurus used for indexing medical articles), pseudo relevance feedback (PRF) and learning to rank (LTR). The search engines included Terrier, Indri, Solr, Elasticsearch and Lucene. The authors note, that many of teams approaches or systems are hard to reproduce and compare due to not enough details being released by the teams on the specifics of their systems. Information retrieval systems are commonly evaluated based on precision and recall. Precision shows what percentage of documents in the results is relevant to the query. Recall shows the percentage of all relevant documents shown in the search results. Medical searches may require either high precision or high recall or both [2].

Some well-performing purpose-built medical search systems, like the ones competing at MedTrack, rely on annotation components that extract medical concepts from the searched documents (and record them as annotations). One such system, QuickUMLS, developed by Soldaini and Goharian [11] is worth noting. It showed comparable or better effectiveness than its counterparts [12] while being significantly faster, which makes it well suited to searching document collections of significant size [11].

Koopman et al. used QuickUMLS in their system designed for task-based medical search concentrating on diagnoses, tests and treatments [3]. The system extracted concepts from 733,138 medical articles (from title and abstract). They found that providing aggregation and interactive elements for the user improved the effectiveness of the search. The aggregation elements (column charts) allowed the user to quickly see the most common diagnoses, tests and treatments in the search results, and by being interactive they in turn allowed them to filter the search results further by one of those diagnosis, tests or treatments. The search engine component of the system was implemented using Elasticsearch [13]. QuickUMLS and Elasticsearch are also used in the practical part of the SIGIR Health Search tutorial [14].

Another example of medical search that uses Elasticsearch is the Openscan.io website. It provides a good example of a user-friendly web interface designed to help the user with their clinical trial search. The data is structured and has a set of tags like Status, Country of Recruitment, Phase and so on. This allows for an easy filtering of required parameters. In addition to filters, the site utilises sophisticated use of the search entry field. By default, it applies query expansion which can be switched off, allows use of Boolean expressions and search of specific fields. The site also offers an interactive visualisation of the same content using Kibana. Instead of tick boxes, this page offers pie charts, geographical map and histograms that the user can interact with to find the trial they are after.

Chapter 3
Methods

The goal of this thesis is to create a search system that can be used by medical professionals to search information in plain-text EHRs that overcomes vocabulary and granularity mismatch challenges of semantic gap by way of extracting concept IDs from plain text and user queries and searching for concept IDs instead of text. The system needs to be simple to use and provide useful visualisation of search results by way of highlighting the searched queries in text and present useful aggregations of data about the search results. It would also be useful to provide interactivity to such visualisations to help users in their searching.

The methods employed by ESS can be summarised as follows:

- Use concept extraction system and a medical ontology to extract medical concepts and their identifiers (IDs) from plain-text EHRs and search queries.
- Annotate EHRs with concept IDs and index them into a search engine.
- Search for concept IDs instead of text to overcome some of the semantic gap challenges.
- Highlight text that matches searched concept IDs.
- Provide clickable charts showing useful aggregations on data contained in the search results that also work as filters.

This chapter will discuss these points in detail, as well as present the system architecture and describe the search interface.

3.1 Concept extraction

The main component of ESS is the concept extraction system (the annotator). Plain text of medical documents is passed through the annotator line by line. The annotator identifies medical concepts within text and returns their names, concept IDs, which semantic type(s) the concepts belong to and where in the line of text they were found (start and end indices). The same process of concept extraction is applied to user search queries by the ESS, with concept IDs being the essential piece of information needed to perform the searching.

Two candidate annotators were identified for this task: NCBO's Bioportal and QuickUMLS. Bioportal has the advantage of being a lightweight annotator in terms of installation and use. It is a publicly available web-based service that can be used via an Application Programming Interface (API) and hence it does not require installation. QuickUMLS on the other hand is a public software that needs to be installed on a local computer before use and has several prerequisite components that also require installation. One of the components is a collection of UMLS Metathesaurus files, downloading which requires registration and approval on the US National Library of Medicine (NLM) website [REF]. Installation of these files is also fairly involved. Bioportal can be easily configured to extract concepts belonging to the SNOMED CT ontology and specific semantic types (categories of concepts, e.g. disorders, symptoms, organisms) [REF]. While QuickUMLS also allows easy specification of required semantic types, it is designed to work with UMLS concept IDs (cuis) instead of SNOMED CT. UMLS concepts, however, are not as specific as SNOMED CT concepts [REF]. Therefore, a further processing step would be needed to map UMLS cuis to SNOMED CT concept IDs. A table comparing these features of QuickUMLS and Bioportal are presented in Table 3.1.

In view of the above, the first choice of the annotator was the Bioportal. However, after some testing it was discovered that the annotating performance of Bioportal was significantly inferior to QuickUMLS. The service was not recognising a lot of seemingly obvious medical terms contained in the supplied text. Hence, QuickUMLS became the chosen annotator to be used in the ESS. Some modification was applied to the set up and inner workings of the QuickUMLS software so that it would return active SNOMED CT concept IDs instead of UMLS concept unique identifiers, hence avoiding the extra mapping step.

Table 3.1: *Comparison of concept extraction systems QuickUMLS and Bioportal*

	QuickUMLS	NCBO's Bioportal
Complexity of installation and use	Complex. Python program with many dependencies and UMLS installation	Medium. Web-based service with REST API
Can specify semantic types	Yes	Yes
Provides span information (start, end)	Yes	Yes
Designed to work with SNOMED CT concept IDs	No, but possible with extra mapping/modification	Yes

Four groups of semantic types were selected for extraction: disorders, symptoms, tests and treatments (presented in Table 3.2). Clinicians are most likely to search for which disorders correspond to certain symptoms, or what tests and treatments to prescribe to patients with certain symptoms or disorders [14]. The four selected semantic groups should contain all the medical concepts needed to answer these queries.

Once QuickUMLS is installed and configured to work with SNOMDED CT IDs and specific semantic types it is able to provide all the needed information for the ESS. An example of input text and the resulting concept extraction performed by QuickUMLS is shown in Table 3.3.

Table 3.2: *Semantic types used in ESS during concept extraction*

Category	Type Unique Identifier	Type name
Disorders	T020 T190 T049 T019 T047 T050 T037 T048 T191 T046	Acquired Abnormality Anatomical Abnormality Cell or Molecular Dysfunction Congenital Abnormality Disease or Syndrome Experimental Model of Disease Injury or Poisoning Mental or Behavioural Dysfunction Neoplastic Process Pathologic Function
Symptoms	T184 T033	Sign or Symptom Finding
Treatments	T058 T061 T121	Health Care Activity Therapeutic or Preventive Procedure Pharmacologic Substance
Tests	T059 T060	Laboratory Procedure Diagnostic Procedure

Table 3.3: *Example of concept extraction from text performed by QuickUMLS*

Input text	"Melanoma is a malignant tumor of melanocytes which are found predominantly in skin but also in the bowel and the eye."
Output	“'start': 0, 'end': 8, 'term': 'melanoma', 'code': '2092003', 'semtypes': {'T191'} 'start': 9, 'end': 29, 'term': 'malignant tumor', 'code': '363346000', 'semtypes': {'T191'}”

3.2 Creation and indexing of annotated documents

Once concept extraction is configured, the next step is to annotate medical documents and index them into a search engine.

The 477 documents used for annotation and searching are part of a dataset used in the 2010 i2b2/VA Challenge on Concepts, Assertions, and Relations in Clinical Text [REF]. The dataset is a mix of plain-text discharge summaries and progress reports, supplied by three medical centres in the USA.

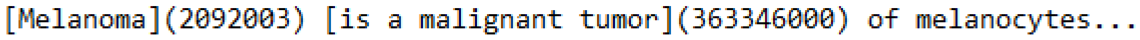
The component of the ESS that does the indexing and searching is the Elasticsearch software [REF]. Elasticsearch is one of the most popular search engines in the world. It is fast, well-documented and free to use. To use Elasticsearch, the annotated documents need to be in JSON format. For each original document a JSON file is constructed having the following seven sections:

- original plain text
- marked-up text containing extracted concepts
- disorders
- symptoms
- tests
- treatments
- filename

Original plain text is used for regular full-text searching.

Marked-up text is the original text augmented with special mark-up that contains extracted concept IDs. An example of marked-up text is presented in Figure 3.1.

This special mark-up is required by the experimental Elasticsearch plugin called Mapper Annotated Text released in the second half of 2018 [REF]. With this mark-up, the plugin can index both, the piece of text containing a medical concept, and the extracted concept ID, and “link” them together. This then allows to find that part of the document by searching for either the words contained in that piece of text, or the embedded concept ID(s). The significance of this feature is discussed in section 3.4.



```
[Melanoma](2092003) [is a malignant tumor](363346000) of melanocytes...
```

Figure 3.1*: Example of marked-up text with embeddings of concept IDs*

The four groups of disorders, symptoms, tests, and treatments are the extracted concepts from that particular document. Each group is a list of concept IDs and concept names. When QuickUMLS matches text to a medical concept, the returned concept name can be one of several synonyms for that concept. To make sure that the same concept name is recorded next to all occurrences of a particular concept ID in the constructed JSON files, the ESS utilises the Ontoserver API. An HTTP request containing a concept ID is sent to the Ontoserver and a preferred name for that concept is returned. These preferred concept names are stored in a separate JSON file for all concept IDs extracted from the plain-text documents. This file is utilised later in the interactive visualisation part of the search interface (discussed in section 3.5).

Finally, the original filename is stored to be displayed in the search results. The resulting JSON files are then indexed into Elasticsearch, ready to be explored in the search interface.

3.3 Searching for concept IDs instead of text to tackle the semantic gap challenges

Having the marked-up version of plain text documents indexed in the search engine allows ESS to conduct searches for concept IDs instead of text. All synonyms of one medical concept found in plain text are “tagged” with the same concept ID. Consequently, searching for that single concept ID finds the documents containing any synonym of the concept, thus overcoming the vocabulary mismatch challenge of the semantic gap.

The medical practitioners and researches, however, do not remember hundreds of thousands medical concept IDs. So, naturally, the plain-text search query, entered by a medical professional, needs to undergo the same concept extraction process as the text of the medical documents. The text search query is sent to QuickUMLS for concept extraction, the extracted concept IDs are passed as the search query to the search engine, and finally, the search results are presented to the user.

The user is also shown the concepts that were extracted from their query. This allows the user to verify that the system is searching for what they had in mind, since concept extraction is not yet a 100-percent-accurate process. To help the user in formulating the queries that are well “understood” by QuickUMLS, the ESS, again, takes advantage of the Ontoserver functionality. When the user types in their query, the entered text is sent via an HTTP request to the Ontoserver API and a list of up to 20 medical concept names, that closely match the entered text, is shown to the user. Selecting one of these concepts populates the search query. The user can then add more text to the query, if needed, and run the search. ESS sends the query to QuickUMLS for concept extraction and having that fully specified medical concept name provided by Ontoserver maximises the concept extraction accuracy.

Ontoserver is also utilised to help ESS overcome another semantic gap challenge – the granularity mismatch. ESS gives the user an option to search for not only the concepts extracted from the search query, but also all their descendant concepts. The concept IDs extracted by QuickUMLS are sent to the Ontoserver API to get all their descendant concepts. The returned list of descendant concept IDs is added to the search query. Consequently, the documents containing any of the synonyms of the concepts extracted from the search query, and/or the synonyms of all their sub-concepts, are returned in the search results.

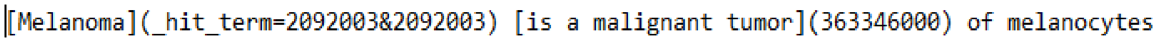
This conversion of text into concept IDs also provides partial relief from the family history challenge of the semantic gap. SNOMED CT ontology has more than 500 concepts to represent family history of some clinical finding. Some examples are: “Family history of malignant lymphoma”, “FH: Gastrointestinal disease”, and “Maternal history of insulin dependent diabetes mellitus”. These family history concepts written in plain text in this form are recognised by the annotator and marked with their relative concept IDs. The search system would then ignore documents containing these terms when searching, for example, for “gastrointestinal disease” or “malignant lymphoma”, unless the non-family-history versions of these disorders were also present in text. The reason why this would provide only a partial relief to the family history aspect of the semantic gap is that family history can also be recorded in clinical text in the following form: “mother had diabetes”, or “father has cancer”, etc. This form of expressing the family history is not accounted for by SNOMED CT ontology and, hence, by QuickUMLS, and such instances would still be tagged with the concept IDs of the mentioned disorders/symptoms.

3.4 Intelligent highlighting of searched terms

A key feature of the ESS is its ability to highlight the relevant spans of text within a document, even though it is searching for concept IDs instead of text.

Word or phrase highlighting has been a standard feature of search interfaces. This allows a user to quickly see where the term or phrase they searched for appears in the search results. Similarly, Elasticsearch can be set up to return snippets of text that contain searched words and phrases (hit terms) for each document in the search results. The hit terms are enclosed in HTML tags that allow system designers to apply desired styling to highlight them. This accomplishes highlighting when searching for words or phrases. The ESS, however, needs to search for concept IDs, not text, to overcome the semantic gap challenges. Tagging medical documents with extracted concept IDs outside of the original text would allow to find the relevant documents when searching for those IDs. Creating a JSON file with a separate section of concept IDs achieves this task. The difficulty, however, is then finding the part of text that contains the medical terms those concept IDs refer to. Finding and highlighting this text would normally be a non-trivial task.

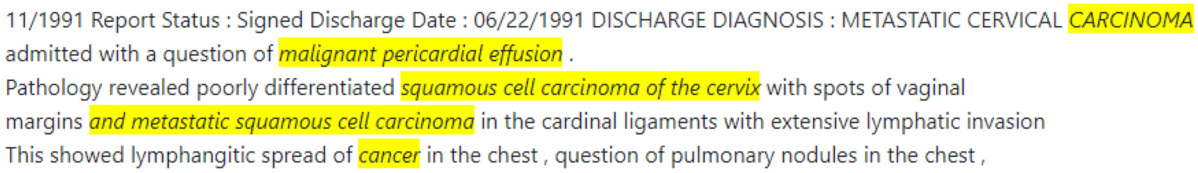
Here is where the experimental Elasticsearch Mapper Annotated Text plugin (discussed in section 3.2) truly shines. Having the original plain text augmented with the special mark-up allows the plugin to index several tokens (words, phrases or concept IDs) as being located in the same spot in the text. Searching for any of these tokens will point to that same spot. This then allows Elasticsearch to apply its highlighting methods, as it does with text search, with some small modifications. Instead of applying HTML tags around the hit terms, the annotated text highlighter, created for this plugin, injects a hit-term indicator to show which of the several tokens anchored to a span of text was a search hit. Figure 3.2 shows an example of this.



```
[Melanoma](_hit_term=2092003&2092003) [is a malignant tumor](363346000) of melanocytes
```

Figure 3.2*: Example of how Elasticsearch annotated text highlighter indicates a search hit*

With simple post-processing, the hit span of text can be surrounded with HTML tags to apply the desired highlighting. During that post-processing, the ESS also removes all the special mark-up and presents the user with just the original text and the



11/1991 Report Status : Signed Discharge Date : 06/22/1991 DISCHARGE DIAGNOSIS : METASTATIC CERVICAL *CARCINOMA*
admitted with a question of *malignant pericardial effusion* .
Pathology revealed poorly differentiated *squamous cell carcinoma of the cervix* with spots of vaginal
margins *and metastatic squamous cell carcinoma* in the cardinal ligaments with extensive lymphatic invasion
This showed lymphangitic spread of *cancer* in the chest , question of pulmonary nodules in the chest ,

Figure 3.3*: Screenshot of ESS highlighting spans of text that match the query*

highlights inside it. Figure 3.3 shows an example of highlighting applied to document snippets in the search results when searching for “Malignant neoplastic disease (disorder)” with descendant concepts included.

3.5 Interactive visualisation of search results

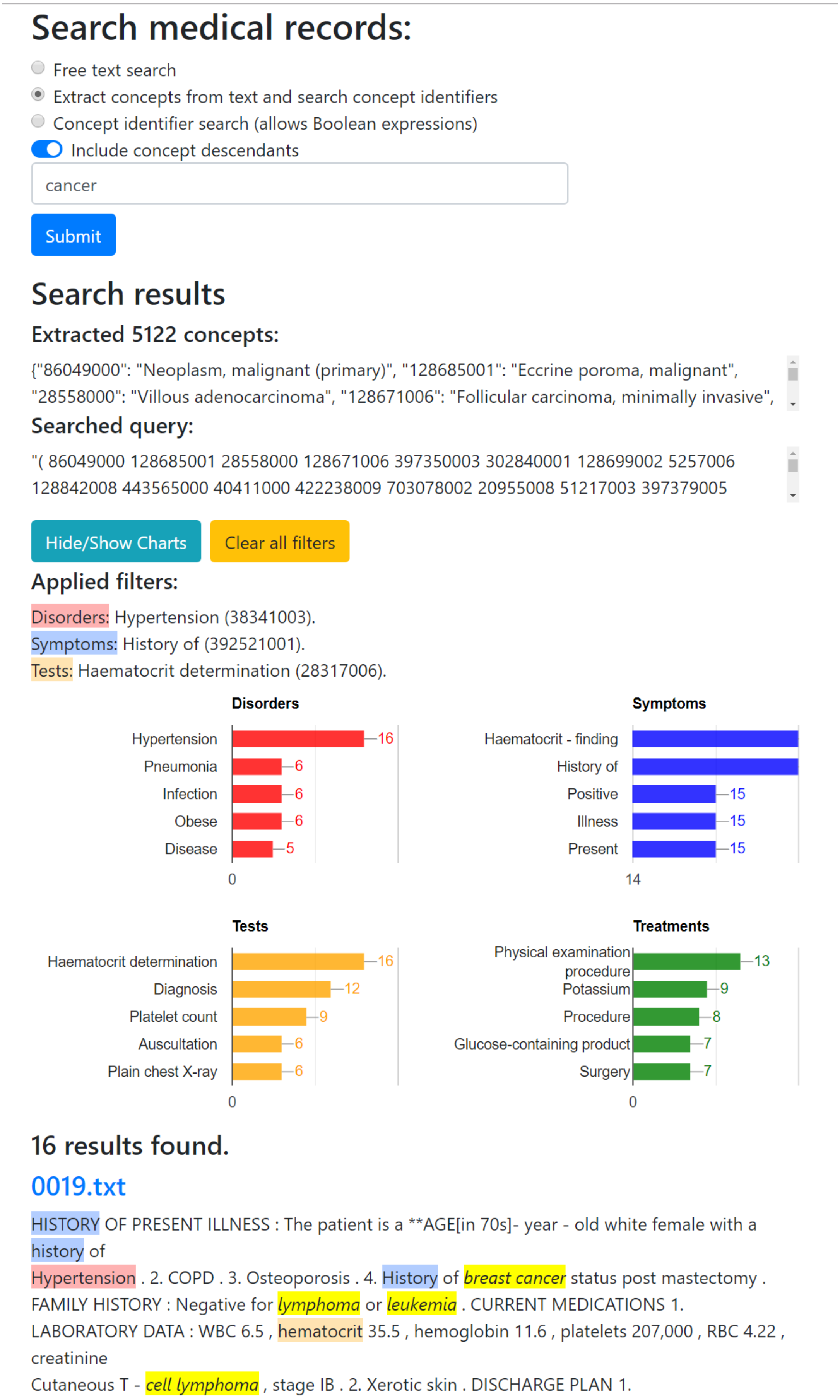
ESS utilises the aggregation capabilities of Elasticsearch in creating useful bar charts based on the search results. Grouping the document’s extracted medical concepts into four semantic categories (disorders, symptoms, treatments and tests) and adding them as separate sections to JSON files for indexing allows to run aggregations on the data. After each query, ESS requests the top 5 concepts for each category (by quantity) that appear in the search results. These top concepts are presented in the four bar charts above the retrieved documents and should help a user to quickly gauge what disorders, tests, symptoms or treatments are most prominent in relation to their query.

The charts are not only informative, but also interactive. Clicking on a concept bar reruns the search and filters out all the documents that do not contain that concept. The charts, therefore, can be used as filters to quickly and easily narrow down search results. Several filters can be applied at the same time, and each filter concept is then highlighted in the returned document snippets and the full document using a different colour for each category. This way, the user will know which highlighted text relates to their search query and which is related to the applied filters. For convenience, the concept names and IDs of the applied filters are listed next to the charts. Filters can be cleared and, if required, the charts can be hidden/shown by pressing the relevant buttons in the interface.

The highlighting of the filter concepts in text is achieved in the same way as with the search query, discussed in the previous section of this chapter. When a concept is clicked on in a chart to use as a filter, its concept ID is added to the search query.

During the post-processing to apply the highlighting and remove the annotation mark-up, each hit term is checked against the list of concept IDs used as filters. The ESS then applies slightly different HTML mark-up to these filter concepts resulting in highlighting with different colours. The charts and the highlighting of filter concepts in search results is shown in the screenshot of the search interface in Figure 3.4.

Since the aggregations are run on the concept IDs and not the concept names, ESS needs to match the concept IDs from the aggregations to the relevant concept names to show them in the charts. This is where the JSON file containing the preferred concept names of all the concepts extracted from the dataset comes in handy (mentioned in section 3.2). The file is loaded when the system is started, and a preferred concept name is quickly retrieved for each concept that appears in the interactive charts after each search. Another option to get the concept names during chart creation could be to use the Ontoserver API. However, if an internet connection was not available during the use of the system, this would not work. Getting concept names from a local resource is also faster than processing several HTTP requests. Another feasible alternative to the use of a file would be to create an Elasticsearch index with just the concept IDs and their preferred names and run a few extra searches in the background to get the concept names when the charts are generated. Finally, since the same concept names are stored with each concept ID in the indexed JSON files for the medical documents, it is possible to run extra searches in that main index to get the required information. Using the file is a lot simpler, however, with as fast, if not better, performance and hence this was the selected choice for this task.



Search medical records:

○ Free text search
● Extract concepts from text and search concept identifiers
○ Concept identifier search (allows Boolean expressions)
● Include concept descendants

cancer

Submit

Search results

Extracted 5122 concepts:

{"86049000": "Neoplasm, malignant (primary)", "128685001": "Eccrine poroma, malignant", "28558000": "Villous adenocarcinoma", "128671006": "Follicular carcinoma, minimally invasive",

Searched query:

"(86049000 128685001 28558000 128671006 397350003 302840001 128699002 5257006 128842008 443565000 40411000 422238009 703078002 20955008 51217003 397379005

Hide/Show Charts

Clear all filters

Applied filters:

Disorders: Hypertension (38341003).
Symptoms: History of (392521001).
Tests: Haematocrit determination (28317006).

16 results found.

0019.txt

HISTORY OF PRESENT ILLNESS : The patient is a **AGE[in 70s]- year - old white female with a history of
Hypertension . 2. COPD . 3. Osteoporosis . 4. History of *breast cancer* status post mastectomy .
FAMILY HISTORY : Negative for *lymphoma* or *leukemia* . CURRENT MEDICATIONS 1.
LABORATORY DATA : WBC 6.5 , hematocrit 35.5 , hemoglobin 11.6 , platelets 207,000 , RBC 4.22 , creatinine
Cutaneous T - *cell lymphoma* , stage IB . 2. Xerotic skin . DISCHARGE PLAN 1.

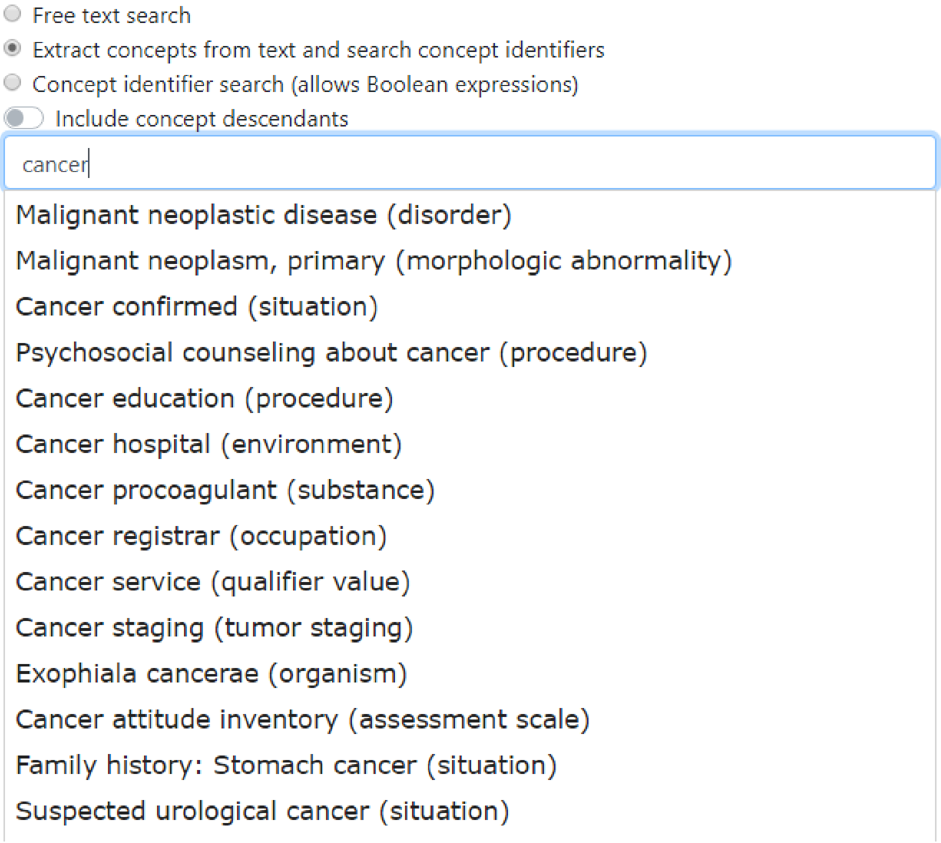
Figure 3.4: *Screenshot of the user interface of ESS*

3.6 The search interface - flexible search options

As can be seen in Figure 3.4, the search interface of the ESS has radio buttons to choose one of three search types a user may wish to employ, the first being “Free text search”. Free text search is a standard feature of any search interface that allows searching for one or several words or phrases. This option allows use of wildcards as well as Boolean operators like AND, OR, and NOT for better precision. Even though free-text search suffers from the semantic gap issues of medical search, this option is provided for quick searching the standard way, if desired, and can be used for comparing search effectiveness with the other search options in the interface. When this option is selected, the search is performed on the original plain-text section of the indexed JSON files.

The second option, “Extract concepts from text and search concept identifiers”, as the name implies, takes advantage of extracting concepts from the search query and runs the searches for concept IDs instead of text to overcome the vocabulary mismatch challenge of the semantic gap. The toggle switch “Include concept descendants” is applicable to this search option and allows to expand the query with descendant concept IDs to overcome the granularity mismatch of the semantic gap. The searches are run on the marked-up-text section of the indexed JSON files.

Finally, “Concept identifier search” is the experimental third search option in the interface that has all the features of free text search (word, phrase, and Boolean searching) combined with the ability to search for concept IDs. The searches are run on the marked-up-text section of the indexed JSON files. With this option, a user can build Boolean queries with several concept IDs and even mix them with words or phrases if desired. Of course, being able to build a meaningful search query with several concept IDs would require the user to know what concept IDs refer to which concepts. There are some features in the user interface that may provide help with this. The main one is the Ontoserver concept look-up and autocomplete functionality mentioned in section 3.2 of this chapter. When either of the first two search options is selected in the interface (Free text search or Extract concepts), the autocomplete feature inserts the fully specified concept name into the search box after the user selects one of the suggested concepts based on what they typed. With this third search option selected (Concept ID search) the autocomplete feature inserts the concept ID instead of the concept name. This, in effect, is equivalent to a quick


○ Free text search
● Extract concepts from text and search concept identifiers
○ Concept identifier search (allows Boolean expressions)
Include concept descendants

cancer

Malignant neoplastic disease (disorder)
Malignant neoplasm, primary (morphologic abnormality)
Cancer confirmed (situation)
Psychosocial counseling about cancer (procedure)
Cancer education (procedure)
Cancer hospital (environment)
Cancer procoagulant (substance)
Cancer registrar (occupation)
Cancer service (qualifier value)
Cancer staging (tumor staging)
Exophiala cancerae (organism)
Cancer attitude inventory (assessment scale)
Family history: Stomach cancer (situation)
Suspected urological cancer (situation)

Figure 3.5: *Screenshot of the Ontoserver concept look-up and autocomplete feature*

version of concept extraction process of the search option number two, but it is limited to obtaining just a single concept at a time. The screenshot of the Ontoserver concept look-up feature in action is shown in Figure 3.5.

There are two more possible sources of concept IDs present in the interface that the user may take advantage of. One is the “Extracted concepts” section that shows what concepts were extracted from the user query (including the descendant concepts if that option was activated). And the other is the “Applied filters” section above the charts that’s populated if some concepts in the charts were selected.

Having the three search options gives the user flexibility in how they can complete their searching tasks.

3.7 Summary of the resulting system architecture

3.7.1 Stage 1 - concept extraction and indexing

- Text from medical files is put through QuickUMLS for concept extraction.
- Extracted concept IDs are sent to Ontoserver API to obtain preferred concept names. These names are stored in a file for future use in the interface.
- JSON files are constructed with seven sections: original text, marked-up text, disorders (list of concept IDs and preferred names), symptoms, treatments, tests, filename.
- JSON files are indexed into Elasticsearch using Elasticsearch PHP client.
- Concept extraction, contacting Ontoserver API and JSON file creation are implemented using Python.

3.7.2 Stage 2 - the search interface

The search interface is a web application running in a modern web browser (tested in Google Chrome and Firefox).

- When using Free text search and Concept identifier search options, the search queries are sent to Elasticsearch as is, using Elasticsearch PHP client.
- When using Extract concepts search option, the search queries are sent to QuickUMLS for concept extraction and then the extracted concepts are sent to Elasticsearch for searching.
- Ontoserver API is sent the text of the search query to obtain concepts suggestions and their concept IDs using AJAX calls.
- The charts are made using Google Charts and their interactions are processed using AJAX calls to the backend server. The previously created file with preferred concept names is used to populate concept names in the charts.
- HTML, CSS, Bootstrap 4 and jQuery are used in the interface page.
- PHP is used for the backend.

Chapter 4
Results and discussion

To evaluate the effectiveness of the system, comparisons were made between three searching approaches: (1) searching for concept name using the “Free text search” option (referred to as Text search for the rest of this chapter); (2) searching for concept IDs extracted from the query using “Extract concepts from text” search option (referred to as Extraction search for the rest of this chapter); and (3) searching for concept IDs with descendants using “Extract concepts from text” search option with “include concept descendants” switch active (referred to as Extraction with Descendants search for the rest of this chapter). The number of returned documents when searching for different medical concepts using these three methods are presented in Table 4.1. In the case of Text search when the query contained several words, the search was first done with words separated by space (equivalent to logical OR search and referred to as word search for the rest of the chapter). Then it was run as a phrase search – the phrase surrounded by quote marks and the search engine looking for the exact match of words in the specified order). Discussion of the search results is as follows.

Searching for concept “cancer” returned 62 documents using Text search, 37 documents using Extraction, and 91 using Extraction with Descendants. With Extraction approach, 2 concepts were extracted: malignant neoplastic disease (disorder) and malignant neoplasm, primary (morphologic abnormality). With Extraction search, the expectation is that documents that contain any synonyms of cancer should be returned in the results, and the total number of returned documents should increase. There was a drop in the number of returned documents, however, compared to Text search, from 62 to 37. Observing the Extraction search results, synonyms cancer, malignancy and malignant neoplasm are returned as expected. However, some of the Text search’s 62 documents containing the word “cancer” have been excluded by the Extraction search. The explanation for this lies in the granularity mismatch manifesting itself in the realm of concepts. In some of the 62 mentions of the word “cancer” this word was part of another medical concept: breast cancer, colon cancer, cervical cancer, lung cancer, etc. All those medical concepts have unique concept IDs, and hence were not found when searching for the extracted 2 general cancer concept IDs. Extraction with Descendants, designed to overcome the granularity mismatch, did exactly that. The IDs of all the sub-concepts of the 2 general cancer concepts were added to the search query (5,120 of them) and resulted in 91 returned documents containing terms like breast cancer, melanoma, chronic myelomonocytic leukemia, pulmonary metastases, etc.

Text search for “hip fracture” as word search returned 48 results with most of the returned documents containing only “hip” or only “fracture” and hence being irrelevant. Phrase search returned 5 documents. Extraction search did not add any new documents to the results with 2 extracted concept IDs being searched. Extraction with Descendants searched for 76 concept IDs and returned 8 documents with the addition of “fracture of hip” and “femoral neck fracture” appearing in the results.

Searching for “flu” provided a more expected result with Extraction search finding documents containing the synonym “influenza”. Extractions with Descendants did not find any new documents due to none of the descendant concepts being mentioned. Notably, one of the 2 documents returned in the Text search for “flu” did not appear in the other two searches, as the word “flu” was part of “flu-like illness”, which is related to a different concept. This shows that searching using concept IDs can be very specific.

Searching for “infection” also shows an expected result. Extraction search picks up the synonym “infectious disease”. The Extraction with Descendants search finds various infectious diseases: AIDS, pneumonia, Hepatitis C, strep throat, respiratory tract infection, etc.

Another search example worth mentioning is searching for “hepatitis”. Text search returns 41 results. Extraction search does not add any synonyms and returns 4 results still containing just the word “hepatitis”. Extraction with Descendants mainly adds results containing “hepatitis C” and some “hepatitis B”. As can be observed, 17 documents from the Text search are still not shown in the Extraction with Descendants. Upon inspection, many of the excluded documents contain “hepatitis B vaccine” and “hepatitis surface antigen negative” which were recognised by QuickUMLS as separate concepts. If the search for hepatitis was to find instances of the disease in patients, the Extraction and Extraction with Descendants effectively excluded the irrelevant documents.

Finally, searching for “coronary artery disease” returned many irrelevant results in the Text’s word search with 271 documents. Text search of the phrase returned 86 documents. The Extraction search narrowed this down to 77 documents picking up one synonym “coronary atherosclerosis”. The Extraction with Descendants did not add new documents but did highlight coronary artery disease sub-concepts (two, three, and multi-vessel coronary artery disease) in already matched documents. Again, the latter two searches returned fewer documents than the phrase search. Upon inspection, it was found that the excluded documents contained “family history of coronary artery disease” or “history of coronary artery disease”, both of which were classified by QuickUMLS as “family history of coronary artery disease” concept. It can be argued that the Extraction and Extraction with Descendants searches removed the somewhat irrelevant results if purpose of the search was to find patients currently treated for coronary artery disease.

Table 4.1: *Comparison of sample search results using three search methods.*

	Text search	**Extraction search**		**Extraction with Descendants search**	
Term/phrase searched	No. of returned Documents: word search/phrase search	No. of searched concept IDs	No. of returned documents	No. of searched concept IDs	No. of returned documents
“Cancer”	**62**	2	**37**	5122	**91**
“Hip fracture”	**48/5**	2	**5**	76	**8**
“Flu”	**2**	1	**9**	27	**9**
“Infection”	**91**	1	**120**	6719	**207**
“Hepatitis”	**41**	1	**4**	146	**24**
“Coronary artery disease”	**271/86**	1	**77**	21	**77**

As a summary, ESS handles the vocabulary mismatch by finding the synonyms of medical terms in text by searching for concept IDs. The expected increase in the number of found documents, however, is not always observed. That is explained by the fact that medical concepts in a medical ontology are very specific. A medical word (e.g. cancer, infection, hepatitis) may be part of a symptom, finding, or disorder or their numerous sub-concepts, and each of them have a unique concept ID. The concept extraction system classifies the found concepts accordingly and allows for very specific searching with ESS. The Extraction with Descendants search effectively deals with the granularity mismatch problem by searching for all the specific variations of a medical concept as well as its general form.

Family history aspect of the semantic gap is also being partially dealt with. The clear statements (for the concept extraction system) in medical text saying “family history of” some concept get correctly classified and excluded from searches for that concept. QuickUMLS, however, is not 100 percent accurate in its concept extraction and sometimes misclassifies “history of …” concepts with “family history of…” concepts. It also may mark a disorder or symptom concept with both the family history of that concept and the actual concept. This misclassification (or dual classification), however, should not negatively affect the search results for the non-family-history concept, since its ID is still present in the extraction.

Overall, the analysis of the search results above suggests that Extraction with Descendants is the most effective search option of ESS. The limitation of this search option is the inability to use Boolean combinations of concepts, which would allow for even more precise searching. Nevertheless, the specific nature of medical concepts and their systematised classification in a medical ontology allows ESS to perform some precise searching as it is while overcoming vocabulary and granularity mismatches of the semantic gap.

Chapter 5
Conclusion

For this thesis, a system was developed to help overcome some of the semantic gap challenges of searching plain-text electronic health records while providing a simple user interface with useful interactive visualisations of data in the search results. The core method of dealing with the vocabulary and granularity mismatch is to match plain text contained in medical documents and the user queries to medical concepts and to search for concept identifiers instead of text. The system achieves this by using QuickUMLS, a medical concept extraction system, that matches text to concepts in a medical ontology SNOMED CT. ESS augments the original text of clinical records with the extracted concepts and indexes specially constructed files into Elasticsearch. By utilising the experimental Mapper Annotated Text plugin, the system is able to find and highlight the spans of text where the medical concepts were extracted from, even when searching for concept IDs. Thus, the system is able to provide a familiar user experience, found with using regular keyword-based search interfaces. ESS deals with the granularity mismatch by extracting concepts from user query and adding all descendant concept IDs to the query. ESS also provides an overview of information in the search results by way of presenting the user with interactive charts showing 5 most common disorders, symptoms, treatments and tests found in the matched documents. The charts also work as filters. Users can narrow down the search results by clicking on one of these top 5 concepts and remove all documents which do not contain the selected concept.

Possible future work includes adding the ability to combine concept extraction with Boolean operators for some very precise searching. In addition, dealing with negated concepts which are currently not recognised by QuickUMLS, should be addressed. Especially when SNOMED CT includes some negated concepts.

Bibliography (incomplete)

[1] C. Patel *et al.*, "Matching patient records to clinical trials using ontologies," in *The Semantic Web*: Springer, 2007, pp. 816-829.

[2] B. R. Koopman, "Semantic search as inference: applications in health informatics," Queensland University of Technology, 2014.

[3] B. Koopman, J. Russell, and G. Zuccon, "Task-oriented search for evidence-based medicine," *International Journal on Digital Libraries,* 2017.

[4] B. Koopman and G. Zuccon. "The Health Search Tutorial," GitHub. [Online]. Available: https://github.com/ielab/health-search-tutorial. Accessed: Aug. 12, 2018.

[5] V. Nguyen, S. Karimi, S. Falamaki, and C. Paris, "Benchmarking Clinical Decision Support Search," *arXiv preprint arXiv:1801.09322,* 2018.

[6] S. International. "Snomed CT, the global language of healthcare." [Online]. Available: https://www.snomed.org/snomed-ct. Accessed: Aug. 22, 2018.

[7] U. S. N. L. o. Medicine. "Unified Medical Language System (UMLS)." [Online]. Available: https://www.nlm.nih.gov/research/umls. Accessed: Aug. 22, 2018.

[8] B. Koopman and G. Zuccon, "Understanding negation and family history to improve clinical information retrieval," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 2014, pp. 971-974: ACM.

[9] L. Kuhn and C. Eickhoff, "Implicit negative feedback in clinical information retrieval," *arXiv preprint arXiv:1607.03296,* 2016.

[10] B. Koopman and G. Zuccon, "Document timespan normalisation and understanding temporality for clinical records search," in *Proceedings of the 2014 Australasian Document Computing Symposium*, 2014, p. 85: ACM.

[11] L. Soldaini and N. Goharian, "Quickumls: a fast, unsupervised approach for medical concept extraction," in *MedIR workshop, SIGIR*, 2016.

[12] H. Hassanzadeh, A. Nguyen, and B. Koopman, "Evaluation of Medical Concept Annotation Systems on Clinical Records," in *Proceedings of the Australasian Language Technology Association Workshop 2016*, 2016, pp. 15-24.

[13] Elastic. "Elasticsearch," [Online]. Available: https://www.elastic.co/products/elasticsearch. Accessed: Aug. 22, 2018.

[14] B. Koopman and G. Zuccon. "SIGIR Health Search Tutorial - Hands on session instructions," GitHub. [Online]. Available: https://github.com/ielab/health-search-tutorial/tree/master/hands-on. Accessed: Aug. 12, 2018.

[15] CSIRO. "Shrimp/SNOMED CT." [Online]. Available: http://ontoserver.csiro.au/shrimp. Accessed: Aug. 22, 2018.

[16] A. Stubbs and Ö. Uzuner, "Annotating risk factors for heart disease in clinical narratives for diabetic patients," *Journal of Biomedical Informatics,* vol. 58, pp. S78-S91, 2015/12/01/ 2015.

[17] Ö. Uzuner, B. R. South, S. Shen, and S. L. DuVall, "2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text," *Journal of the American Medical Informatics Association : JAMIA,* vol. 18, no. 5, pp. 552-556, Sep-Oct.