User Experience in Medical Search Engines

Róbert Andri Kristjánsson

Kongens Lyngby 2014
IMM-MSc-2014-????

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk IMM-MSc-2014-????

Summary (English)

FindZebra.com is a specialized search engine for rare diseases that has been developed to as an improvement to standard search engines. FindZebra.com has been shown to improve diagnostic quality when compared to traditional search engines. We believe that while improving the relevance of results in a search, the presentation of results and interaction with the search engine are equally important. In its current form, the results in FindZebra.com are presented as raw text from articles as well as ranking of the results likelihood to match the query.

Web search engines are now the second most frequently used online computer application [29] and a wide range of innovative interface ideas have been developed. The goal of the project is to improve user interactions on the FindZebra.com website using state of the art user experience engineering methods as well as machine learning for customised results.

Faceted navigation using symptoms extracted with machine learning algorithms has been designed, implemented and tested in real life settings using diagnostic cases to evaluate the performance of the feature. The faceted navigation has been tested against pagination and shown to be an improvement in medical cases that prove difficult in retrieval for the search engine. In addition to the faceted navigation, the display of results has been improved by grouping multiple instances of the same disease together.

Summary (Danish)

FindZebra.com er en specialiseret søgemaskine for sjældne sygdomme. FindZebra.com har vist sig at forbedre den diagnostiske kvalitet sammenlignet med traditionelle søgemaskiner. Det er vores hypotese at en forbedring af præsentationen af søgeresultaterne og brugerinteraktion er lige så vigtig som forbedringen af relevansen af søgeresultaterne. I sin nuværende form bliver resultaterne vist med rå tekst fra artiklerne og rangeret udfra match med søgestrengen.

Internet baserede søgemaskiner er nu den næstmest brugte online applikation [29] og en bred vifte af innovative interfaceideer er blevet foreslået. Målet med dette projekt er at forbedre brugerinteraktionen på FindZebra.com ved hjælp af state-of-the-art metoder indenfor brugeroplevelse og maskinlæring.

Facetteret navigation med brug af symptomer ekstraheret ved maskinlæringsalgoritmer er blevet designet, implementeret og testet i et test bruger set-up. Medicinske kasuistikker er blevet brugt til at evaluere resultatet af den foreslåede funktionalitet. Den facetterede navigation er blevet sammenlignet med sidevisning. Det viser sig at den facetterede navigation er bedre til at finde de diagnoser som søgemaskiner har svært ved. Derudover er visningen af resultater blevet forbedret ved at gruppere resultater for den samme sygdom.

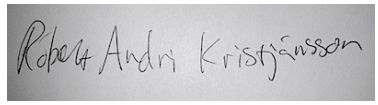
Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in fulfilment of the requirements for acquiring an M.Sc. in Digital Media Engineering.

The thesis deals with user interface engineering of medical search engines.

The thesis consists of design and implementation of new interface features as well as testing of those features.

Lyngby, 01-April-2014



Róbert Andri Kristjánsson

Acknowledgements

I would like to thank Ole Winther for this great opportunity as well as guidance throughout the project. I would also like to thank Dan Svenstrup and Philip Henningsen for the invaluable work they have done for the findzebra website and for changing their priorities in the development of the website to facilitate the making of this thesis. I would like to make special note of Dan Svenstrup for his good advice as well as answering my calls and questions selflessly during the middle part of the thesis.

I would also like to thank Henrik L Jørgen, as well as my group of testers for user interfaces, Tómas Vignir Ásmundsson, Ásmundur Jónasson, Ásdís Árnadóttir, Oddbjörg Ragnarsdóttir and Ingimundur Árnason.

Lastly, a big thank you to all of my family for bearing with my madness during the latter part of this project.

Contents

Chapter 1

Introduction

1.1 Search engines

Search engines (SE) have become an integral part of the daily lives of people. A whopping 73% of all Americans use search engines and on any given day 59% of americans use search engines [46]. There is so much revenue to be had from search engines that some of the worlds largest firms have risen from ad revenue on search engines [48]. General search engines have historically been so far ahead in search technology as well as user interface than specialised ones when it comes to user interface and usability that users have trended towards using the general search engines for all user needs. These specialised search engines that focus on a specific segment of content are referred to as vertical search engines and the search engines that focus on more general content are horizontal.

When it comes to search engines, the market has been dominated by a single search engine, Google, as far back as the year 2004 [46]. Google appears to have been so effective, it seems, that vertical search engines have not caught on until recently. These vertical search engines have implemented more unique features and have seen popularity increases in the past two years [36, 40, 25, 34]. Recent trends show that users are migrating away from the giant and into some of the specialised search engines.

The Search User Interface (SUI) is the gateway to any Information Retrieval System. It is the face of any search engine and considered a very important feature of the tool. Today, the SUI is more than a tool used by professionals, it is responsible for "teaching" novices to use the Information Retrieval System. Todays computer standard is that computers should be usable by anyone and it is the goal of any user experience design that the system pose no barriers to newer users [42].

The basic objective of any SUI is to aid users with the formulation of their queries. To further help them with their information needs an SUI needs to present the search results as well as keep track of the users search progress. It is important that the interface rids the common users of any need for a user manual. These goals need to be met by finding the optimal complexity of interaction for the user. Google's success has been linked with their extremely simple and intuitive interface [39]. With the rise of popularity for Google, other search engines have followed suit when it comes to user interface and thus a process of iteration where search engines compete for the best interface.

1.1.1 Search subjects

Search has become a part of software and is available in all operating systems [29]. No longer is search considered an advanced feature in operating systems, it is an intuitive and an integral part of the operating system use. Search is used to find any website a user desires, for example products for online shopping, suggestions about activities or vacations, educationally or for any informational requirement [29, 9]. Search engines have met the requirements of users and predict what a user is searching and display it above search results so that in some cases, a user is not even required to visit any other page than the search engine website [6].

Findzebra is a search engine that focuses on a specific and narrow search subject. The objective is to provide a more specialised tool that is to be used with that subject alone. Findzebra is unique in it's specialisation in only rare diseases. This means that the Findzebra project offers a lot of opportunities to design a user interface since it is different from the bunch in the content it displays as well as in how the search engine is used.

1.2 User interfaces

User interfaces have been realised to be an essential part of any software that is developed. It is essential to the correct use of any software that its user interface allow the user to easily understand how the software is used. A popular modern design guideline is that a user should not be required to read a user manual before using software. Search interfaces have evolved towards a very simple initial query interface with most of the interface hidden [29]. Once a query has been entered the search interface becomes relevant and is revealed. Research has shown that more specific queries lead to more relevant results and therefore a large part of the goal of a user interface it to aid the user in refining the query [21].

1.2.1 Search user interfaces

The goal of this thesis is to research, modify and apply modern methods in search engines to a medical search engine and to assess and test how well these features perform in a medical search engine. This is done by adding query enhancement features as well as a more efficient way to display results. Machine learning will be used to generate relevant suggestions based on the matching query. This will aid the user in sorting the results for a given query. Emphasis will be on not disrupting the current regular users of the search engine. This will be done by disrupting the current user interface as little as possible.

1.2.1.1 Faceted navigation

Faceted navigation is a navigation method that uses groups of filters that have a common theme to filter results. The filters are often grouped together and multiple filters can be applied to the same search. A good example of this is in online shops, if a user searches for a laptop, a faceted navigation interface will often display a brand category with filters for the most common laptop brands. Other possible facets can be screen size or operating system.

Faceted navigation can be implemented readily in web sites that utilise manual input of articles into a large database. That way, when a new article or item is added to a database, it's associated facets can be put in at the same time. This method of manual input is not scalable for a search engine that delivers results from multiple sources. A method that has been used by some search engines [4] is to use clustering methods to try to sort the search results into clusters of

items with a common theme and offer navigation between those themes. This method offers a lot of scalability, however it runs a risk of creating clusters that are not sensible. This happens when the search results are vastly different in content, which is the case of general search engines.

This project will apply faceted navigation to disease symptoms in a scalable way using machine learning to extract symptoms out of articles. An interface for the navigation will be designed and implemented.

1.3 Machine learning

Machine learning is a branch of artificial intelligence that focuses on the study of systems that can learn from data. This field of study is very useful in the generation of scalable "smart" solutions.

The field of machine learning is used widely by in search engines. It is used by google in the ranking on websites, affecting the search ranking based on users historical behaviour [1, 35]. It is further used by many vertical search engines such as Amazon's product search and Youtube.com video search to suggest search results to users. Machine learning is also widely used in advertising online to provide targeted advertisements for each user to be displayed [8].

The problem with machine learning is that it can be difficult to predict subjects that are very open in nature. This makes the tool perfect for a specialised search engine. Machine learning methods can be applied to find relevant results that can be useful for the user. This is under development in the Findzebra project, one use for machine learning is to extract symptoms from articles and present to the user.

Chapter 2

Literature Review

Because of these strict standards that have formed over the years, a lot of research has emerged in SUI development. A vast amount of features have been tried and measured with various methods.

In this literature review, the author will go through the state of the art technology and methods used in SUI today. However, to better appreciate where we are today, a brief historical summary of where we have been will be given as well.

Search engines today welcome their users with a simple search bar as well as a few words of instructions [6, 3]. This presentation has been researched heavily and simple designs yield better results [20, 28, 41, 53]. Through years of development, a trend has developed towards more intuitive querying systems. That is: the user should not have to learn a specific language and the query (or search term) should be as natural to the user as possible [41]. In addition to these human language queries, websites have offered more complicated queries to accompany the simple querying system. These additional features receive limited use by their users and many of the users utilising these advanced features seem to misunderstand them [27, 31, 50].

In displaying results, similar rules apply. The user should not be presented with information that the user has difficulty understanding in addition to avoiding

that the user be presented with irrelevant information. Designing a search UI is all about the perfect balance between information and simplicity.

Modern search engines include features such as natural queries, instant preview of results (Google Instant) [6], related search predictions, spelling corrections and query reformulation aids, result previews, click free search, categorisation and clustering, displaying summaries and more. Each of these technologies are the result of development and testing. The following sections will go through these features and write about those that are "state of the art" in each genre. The sections are split into the different sections of using a search engine, i.e. query formulation, query reformulation and results navigation.

2.1 Query formulation

This section is focused on technologies in query formulation. The primary goals with UI features in query formulation is to aid the user in being able to search the terms that give him the most relevant results.

2.1.1 Natural queries

With more development in the supporting technologies, search engines have started to support natural queries to a greater extent. Good examples of this are voice search tools such as Apple's Siri voice search and Google Now. These modern search tools even go so far as to try to predict what information the user seeks before the user even enters any search using behavioural patterns as well as search history [16, 7]. Many natural language queries posted to google yield an instant result on the right side of the page that is a direct answer to the asked question. A search of "Who is Chris Paul?" for example yields a result on the right side of the website showing figures as well as a mini biography for the basketball player Chris Paul, this can be seen in figure 2.1.

Figure 2.1: Search results for the query "Who is Chris Paul?"

2.1.2 Click free results

While the example in figure 2.1 is shows a natural query in action, it is also an example of another feature that was relatively recently added to google. That feature is the click free search. After typing the query, the user was never required to click a thumbnail from search result because the search engine knows answers to common questions and displays summaries when appropriate.

The two aforementioned features are a part of a transition in search engines from "Give me what I typed" to "Give me what I want." [26].

2.1.3 Real-time search

The subject of response time is one that has been researched over the years and affects how successful queries are [47]. Google has implemented the feature

'Google Instant' which shows search results while the query is being typed into the search box. This blurs the line between query formulation and query reformulation and allows for the user to evaluate the top links while the query is being formulated.

Google claims this real-time search feature can save 2-5 seconds per search [13]. While the author was not able to find research to support those claims directly, various research has shown that showing results immediately is important to user satisfaciton [30, 44, 32]. Furthermore this research also shows that the user should not have to look at supplementary information before viewing results as it slows down the searching process.

2.2 Query reformulation

Unfortunately, the first query a user issues does not always yield the desired results. While some users have a tendency to give up after that, a more common strategy is to refine ones query [45]. This can be done in various ways, a user can manually retype a new query, use spelling correction or try some of the queries that are suggested by the search engine.

2.2.1 Spelling correction

Spelling correction has converged towards suggesting the correct spelling of misspelled words in a minimal manner between the search box and the results. Spelling correction is often done automatically but not in all search engines. In the more common web search engines such as google, yahoo and bing [6, 19, 3], the correction is done automatically after which the user is offered to revert the correction at the top of the search results. This is a design move based on "give me what I want" which also suggests that more often than not, a typo is made rather than a differently spelled query was desired.

2.2.2 Query suggestions

Query suggestions is a feature is more varied between different search engines. Search engines have historically altered this feature more often and there is a more significant difference in the presentation of this feature between search

engines [6, 19, 3]. Query suggestions have been shown to improve both search results and search speed [52, 23].

Google, yahoo and bing for example offer query suggestion at the bottom of the search result pages with full queries. The user is offered 4 queries that are complete. A search project called the BioText project has experimented with different types of query suggestions. One method is to offer words with tick boxes so the user could make up his own custom query using a list of suggested additions. This is a method that allows a great number of possible sentences to be formed while keeping the user interface clear of clutter [23].

2.2.3 Categorization, Clustering and faceted navigation

Another way of refining a query is the faceted navigation. That is where the user is presented with a categorised view of his search results with filters in each category. In addition to this, some search tools have numbers indicating how many results are under each category. The user can then select categories or tags and continue to refine his search (as long as any results falls under the search) and narrow down the results before starting to look through thumbnails. This is especially popular for large online sellers such as amazon and ebay [2, 5]. The faceted navigation is especially good for collections of results that have a definable finite length and are easily categorised based on features.

Experiments have been made towards using machine learning technologies to categorise results to be able to offer faceted navigation for search results that are not stored in a categorised collection. User responses to categorised search results with faceted navigation have been positive and the categorisation is thought to be appealing [25, 34, 33, 54].

2.3 Result navigation

When a user has entered a search query, the SUI displays the results as a list of thumbnails with a description text below each one. The description text has varied in length over the years as is explained in detail in section 2.3.1.

2.3.1 Query-oriented Summaries

Results pages on search engines have gone from displaying the first lines of text on a webpage towards a more contextualised summary of webpages. The stub of text that appears below a thumbnail today includes the words that were used to form the query in the more common search engines [6, 19, 3]. They are further made apparent by bolding the search terms. There is research supporting this methodology and it shows that query-oriented summaries improve both recall and precision while participants viewed fewer documents in order to get to their result[49, 51].

2.3.2 Instant preview of websites

A feature that has been replaced for thumbnails was an instant preview of the website when the result was clicked. These previews were available without leaving the search result page and therefore without disrupting the overview of what has been viewed at what time. Why this feature was abandoned is not known to the author however it is likely that the feature offered too much clutter to the user and thereby did not meet a required balance between information and simplicity.

2.3.3 Sorting results

A robust alternative to using clustering to categorise results would be to sort them according to known factors, such as when a webpage was archived, whether a user has visited the page and what type of webpage the it is, that is whether it is a video or a forum or such definable features. This is available in googles "search tools" below the search box. These search tools were previously offered in a sidebar but were moved to a more discrete location, hidden at the top of the page until revealed. This is an example of SUIs becoming simpler with time.

2.4 Medical search engines

Today, some medical search engines are commonly used, for example iMedisearch, Healthline, WebMD, PopoFrog, Pubmed, Healthfinder [12, 11, 15, 14, 18, 10]. Those that have an interface that differs from traditional search engines are focused on individuals without medical background or education. The iMed

intelligent medical search engine is a research project for a medical search engine and one that has had a lot of innovation. The iMed intelligent medical search system asks the novice users a series of questions, much like going to the doctors office would be for that patient. The patient gets to choose from a list of predefined symptoms and the search engine tries to find a matching disease to the patients descriptions [37, 38].

2.5 Jakob Nielsen's heuristics for user interface design

There is without a doubt a lot of research on search engines and a lot of development that has been done in the past years. Implementing these features can become difficult and objectives can be lost quickly. Jakob Nielsen has designed heuristics for user interface design that are proved to produce user interfaces that yield better adoption. During this project, these design guidelines have been used extensively. Nielsen's evaluation methods have been found to be successful in aiding with design of cohesive user interfaces as well as improving the user interaction [43]. Nielsen's heuristics can be summarised in the ten most general of his guidelines and those are listed below and used for reference in this thesis.

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognise, diagnose, and recover from errors
10. Help and documentation

Chapter 3

Design decisions

After an interview with Physician Henrik L Jørgensen, who is a collaborator on the Findzebra project, a need for a way to query Findzebra in a more specific manner was expressed. Jørgensen talked about needs to ask the search engine for a specific type of disease and a way to ask the search engine to retrieve articles of specific types, such as results of medical testing. In the meeting with Jørgensen, ideas of advanced query formulation were discussed. The focus of the discussion was to find a user friendly method to express to the search engine limitations on the search results to be presented. Ideas of implementing these advanced search filters using the query, using an advanced search options or if there were other ways. Complex search queries have historically failed to be used correctly [27, 31, 50] and is therefore a subject of a different research to find an improvement or a more natural way to apply these filters.

This projects focus is on meeting these requirements while trying to improve the user interface for the site in an as general way as possible. Advanced options in search interfaces often add a layer of complexity that can make a site less attractive to the more basic user and are often used incorrectly [27, 31, 50]. This is an extremely difficult situation as the advanced users are often minorities, yet they are often the users that use the tool the most. The requirements of Jørgensen were that it should be possible to query the search engine with some configured options of content such as type of disease or affected age groups. In addition to Jørgensen's wishes, it was decided to make result browsing more

robust that instances of the same disease from multiple sources, be combined in the display of symptoms.

To accommodate these wishes, two design ideas were created. The first design, which is an advanced querying tool that would be accessed via an "advanced options" button from the regular site, was a query box with drop-down menus for filters. A paper prototype of this design is shown in figure 3.1.

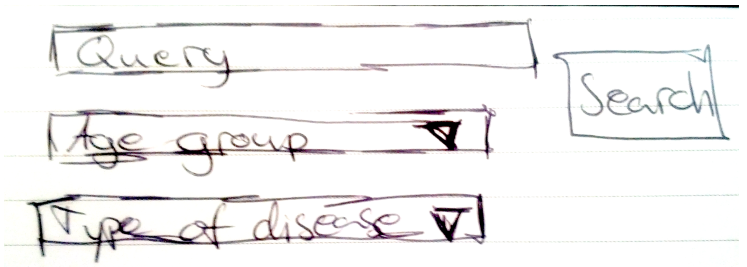


Figure 3.1: Advanced querying system with dropdown boxes.

The second option was a filtering system based on faceted navigation that would be revealed once a user has entered a query. While this system may not necessarily provide as predictable results for the users that know exactly what they are searching, it allows for the tweaking of the search query after it has been searched as well as the possibility of being more prevalent to the basic users. This can lead to a wider adoption of the feature by the SE users. This option has an array of possible filters that are displayed post query entry and can be clicked to apply each filter. These filters show how many results are associated with each one of them, preventing the user from getting empty result sets. A paper prototype of this configuration can be seen in figure 3.2.

Figure 3.2: Advanced querying system with dropdown boxes.

These options were evaluated with paper prototypes as is discussed in section 5.3

3.1 Faceted navigation

When designing the faceted navigation interface, decisions need to be made regarding the implementation. First of all, an alignment of the navigation needs to be chosen, whether it is a vertical sidebar or a horizontal bar. Secondly, how the filters are displayed with regard to how many filters are shown as well as number of lines and arrays used for display needs to be chosen. Thirdly, the

placement of the bar needs to be placed, a vertical bar can be placed on either side and a horizontal bar is generally placed either at the top of search results or at the bottom. Finally the bars visibility needs to be considered, it needs to be visible to the user but cannot remove visibility from other features of the SE. The search bar can be either hidden and shown with a press of a button or can remain visible at all times.

3.1.1 Alignment

When designing the faceted navigation, the first design decision was whether or not the navigation should be of vertical or horizontal alignment. The findzebra website in its current version already has a sidebar that displays search results and to follow Jakob Nielsen's heuristics of consistency and standards, as well as aesthetic and minimalist design a third sidebar would not match standards. Therefore a horizontal navigation was chosen.

3.1.2 Filter arrangement

Once an alignment has been decided for navigation, the filters need to be arranged into a suitable number of rows and columns to take up an appropriate amount of space on the website. It was decided to have three rows of filters based on the fact that this is was the smallest width that still allowed all filter names to fit in a single line. The number of rows was decided to be five because a larger number of rows means the filters will take up too much screen space on low resolution screens.

3.1.3 Placement

While the navigation can be placed anywhere, top or bottom are standard in websites and allow for better visibility and recognition. Top or bottom placement is consistent with Jakob Nielsen's heuristics of visibility of system status, consistency and standards and recognition rather than recall. A group of five potential users were interviewed to research this subject. A detailed overview of the evaluation can be read in section 5.4
The design options that needed to be evaluated were:

- **Configuration 1:** Top placement visible at all times

- **Configuration 2:** Bottom placement visible at all times
- **Configuration 3:** Top placement hidden and revealed with button press
- **Configuration 4:** Bottom placement hidden and revealed with button press

3.1.3.1 Configuration 1: Top placement visible at all times

This configuration places the navigation between the textfield for query input and the search results and can be seen in figure A.1. Placing the navigation at the top of the search results offers the best visibility according to the tested users. While this is very good, most users found the navigation to take away space from the search results. Four of the users expressed that they found the search bar easy to confuse with search results. This could possibly be remedied by a different design for the faceted navigation or a clearer indication for it. Two of the testers were concerned about performance at lower screen resolutions because only a few of the search results were visible after searching. The faceted navigation is not a critical feature used by all users and therefore this design was rejected based on the need for users to scroll down to analyse results. This is especially true since analysing results is usually required before applying filters as the search query may result in the correct result being in the top pages.

3.1.3.2 Configuration 2: Bottom placement visible at all times

This configuration places the navigation right below the last displayed result and can be seen in figure A.2. Placing the navigation at the bottom takes little space from the search results and is viable as it can be seen as the user has scrolled through the results. However three of the surveyed users failed to notice the faceted navigation during the period of using the search engine. This also requires a user to scroll far down to be able to use the feature and provides cumbersome for the users that use the feature. A note to be added is that the users that were presented with this configuration first did not have a problem with differentiating the faceted navigation from the search results despite the fact that the navigation is identical in appearance as configuration 1.

3.1.3.3 Configuration 3: Top placement hidden and revealed with button press

To try and give the visibility of having the navigation at the top while removing as little as possible from the initial display of results a hidden configuration for the faceted navigation was tested. This configuration is placed in the same spot as configuration 1 and can be seen in figure A.3. This scored marginally worse in how visible the feature is and how much effort it required to be used compared to configuration 1. However the four users that expressed that configuration 1 could be confused with search results all stated that this configuration was much clearer in differentiating between search results and the faceted navigation.

3.1.3.4 Configuration 4: Bottom placement hidden and revealed with button press

This configuration places the navigation hidden in the same spot as configuration 2 as can be seen in figure A.4. Placing the navigation below the search results and hidden performed by far the worst in the user testing. As was expected, it had the least visibility but performed identical in disrupting the interface. Users failed to recognise the filter button. This configuration requires even more effort to be used than configuration 2.

3.1.3.5 Placing the navigation above the search box

While this was not tested with users, the author has decided to dedicate a section to this idea, as it is theoretically possible. This design however was thought to disrupt the use of the website too much and thought to be too far from conventions. Another benefit of having it below the query is that it is fitting with the intended use of the tool i.e. entering the search query first and filtering after.

3.1.3.6 Configuration decision

Since configuration 3 performed best on average in the user testing, it was decided to use a top placement for the navigation that remains hidden until a button is pressed.

3.1.4 Status display

To conform with Jakob Nielsen's first user interface design heuristic, visibility of system status, a status display was designed that shows the applied filters. Once a status bar like this has been implemented it was thought logical that functionality to remove filters be added to the interface for convenience. This addition conforms with Jakob's heuristic of user control and freedom. When a filter is applied, it is revealed under the navigation panel and above the search results as can be seen in figure A.5.

3.1.5 Bootstrap

The findzebra website is currently implemented using a framework for front end webpage development called bootstrap. Bootstrap contains HTML and CSS-based templates for typography, forms, buttons and more. Bootstrap was developed by Mark Otto and Jacob Thornton as a part of Twitter and was later made available open source. Bootstrap provides a familiar ground in terms of web design as it is widely used and used by a large website such as Twitter. It was chosen to use this framework for the development of the faceted navigation as it confirms with the current design of FindZebra as well as being widely used and therefore familiar to most users of the web.

3.1.6 Faceted navigation post-impressions improvements

Once the faceted navigation had been designed and implemented, it entered testing, however it was quickly realised that the faceted navigation with 15 filters available sorted in the order from highest prevalence to the lowest was problematic. It failed to provide access to the diagnosis articles in the diagnostic cases, mostly due to the fact that the required symptoms were not available. Often times, the highest ranked facets were unable to provide relevant filtering. After impressions from Henrik L Jørgensen as well as the FindZebra team it was decided that more filters needed to be available to the user. To aid with this problem, a scrolling feature was implemented. Instead of the initial 15, 45 facets were made available by displaying three pages of facets, which were side scrollable. To conform with Jakob Nielsen's heuristic of visibility of system status, a small indicator of facet page number was implemented, matching the colour scheme of the rest of the site.

The scrollable feature can be seen in figure 3.3.

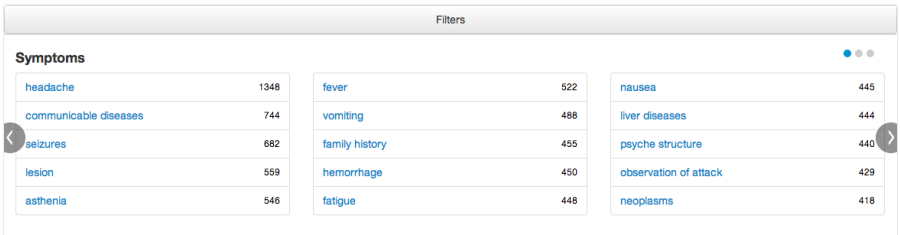


Figure 3.3: Scrollable faceted navigation.

3.2 Combining diseases

Combining diseases was implementing using a design that was already in FindZebra as a test feature. The appearance and function of the feature was not designed by this author, but only implemented into the website. This feature was left as-is because the author believes in it's current form, the feature is sophisticated and there isn't much design to add.

An alternative to this method of displaying multiple sources, the author would suggest using machine learning algorithms to combine the multiple sources into one larger article with more information and citing the various sources of input.

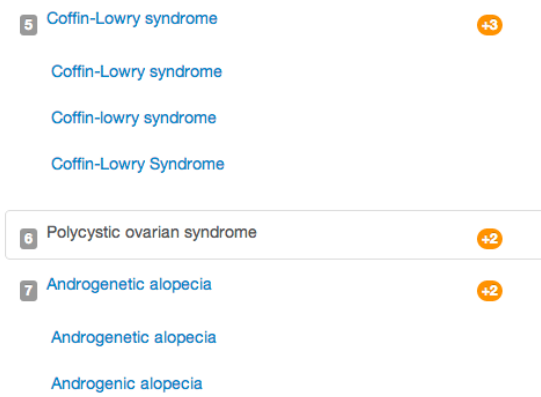


Figure 3.4: Scrollable faceted navigation.

Figure 3.4 shows the multiple sources in action. The feature is designed so that the highest ranked match of the multiple sources is shown with a badge that

acts as an indicator that there are more sources. Clicking on the item in the list, expands the list and shows all of the sources with their respective names (which are often the same name with only slight differences). The same disease articles are ordered according to rank inside that list.

Chapter 4

Implementation

Once the user interface features have been designed and the evaluation described in section 5.5, the features need to be implemented in order to be further tested in real world conditions. The implementation of the website is split into front end implementations of the webpage, back end implementations of the website as well as queries to the database. Figure 4.1 shows a diagram of how the findzebra website interacts with the web client as well as the Solr gateway as an overview.

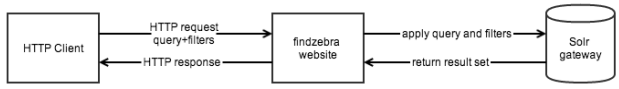


Figure 4.1: Findzebra network diagram

4.1 Webpage User Interface (front end)

The interface of the website was implemented using Twitter's bootstrap framework discussed in section 3.1.5 for initial design with some minimal changes

made manually in CSS. The bootstrap framework provides graphical designs as well as typography for buttons and the best fitting buttons were chosen in each place.

4.1.1 Hiding the navigation

The navigation was implemented into an accordion. An accordion is a component of bootstrap that uses JQuery to reveal it's contents when an activation button is pressed. An accordion is a segment of a website that is loaded along with the website but is not shown to the user until it is activated. The accordion is designed in such a manner that the activation button spans the whole width of the button, the contents of the accordion are enclosed in a black border. This provides a large button for the filters, spanning the entire width of the site. When the accordion is clicked, the accordion reveals and or hides it self with a sliding animation.

4.1.2 The navigation array

The navigation array was implemented using three vertical navigation elements, each with 5 items. These navigation elements provide a hover effect when the mouse is over the element, aiding with selection. The text inside each element is blue. Further, plain text formatted black instead of blue and aligned to the right was used to indicate the number of results associated with each of the filters available.

4.1.3 Scrolling the faceted navigation

The scrolling feature was implemented using a carousel. A carousel is a component of bootstrap implemented in JQuery to reveal only an active item of an html list. In addition to this, a JQuery function was implemented to display the status of the scrolling, i.e. which page of scrolling the user is located on. This was thought especially important because the scrolling can go "end-to-end" meaning that scrolling forward from the last item on the list brings up the first item of the list which was confusing to users.

4.2 Search platform

The search platform used in this project is Apache Solr. This search platform supports faceted navigation. The faceted navigation is implemented so that for each element that is faceted, 1000 facets are generated, each with an associated number of results. The generation and counting of the facets is handled by Solr completely and communication to the search platform is described in section 4.3.2.2. For further information about faceting in solr, please refer to the solr wiki [17].

4.3 Webpage controller (back end)

The webpage controller was implemented in python using the web2py framework. Web2py is a free open source framework that offers easy development of secure and database-driven websites. Web2py applications follow a model-view-controller (MVC) architectural pattern which was held to during development. As the website was already developed, this was not a design choice made by the author.

4.3.1 Solr facets

The facets generated by Solr are provided unsorted in the format *UMLS ID*:*Symptom name*. Regular expressions are used to separate the two and store in variables. The symptoms are then sorted in an order from highest number of associated results to the lowest. This choice was made as it was believed that the more frequently appearing symptoms would more often be relevant to the searcher. However it is possible that the most frequent symptoms are too general to be of benefit in narrowing down search.

4.3.2 Handling URLS

It was decided that the faceted navigation be associated with the URL in a way that if filters have been applied, these filters are apparent in the URL. This way should a URLs be shared among users, the applied filters are preserved.

In order to implement this, a request variable was introduced to store the filters as they are separate from the query request variable. The request variable is a

string that is then further parsed to acquire the appropriate filters. The filters are supplied to the Solr engine as UMLS ids.

4.3.2.1 Facet request variable

The facet variable is to contain UMLS IDs followed by a star and a comma. All unwanted symbols are removed using regular expressions. The request variable is split by commas into a list of strings which is then further fed into the Solr gateway one by one into a function that applies each filter.

4.3.2.2 Solr methods

Three methods were created to interact with the Solr gateway. One to apply a facet, one to remove a previously applied facet and one to clear all applied facets.

Chapter 5

Evaluation and user testing

The evaluation and user testing for this project was carried out in two phases. The first phase was during the design process, which was done on only a handful of subjects. This phase utilised functional testing as well as usability testing on prototypes later in the process. The latter phase was the evaluation of the final product. The evaluation of the final product was more focused on performance during tasks that the users performed rather than a users experience. The main goal of the evaluation is to see whether or not the project was successful in improving the tool in extraction of correct diagnoses for a set of difficult diagnostic cases where the search engine has previously failed.

5.1 Diagnostic cases

In a previous test of the FindZebra search engine, 56 queries with known correct diagnoses were used to test the search engine. These cases were created from difficult clinical cases where the query text was extracted direcly from the patient symptoms. The cases varied from being created by clinicians, being taken from journals as well as being taken from articles. These cases can be read about in full detail in the article "FindZebra: A search engine for rare diseases" [24] published by the International journal of medical informatics.

Some of these diagnostic cases have proved difficult in extraction of the correct result, despite improvements in search algorithms, these cases are of special interest and will be the subject of testing in section 5.5

5.2 User testing subjects

The same five subjects were used for the testing in sections 5.3 and 5.4. The test subjects were chosen of as different backgrounds, occupations and ages as the author found possible with a set of five. These were a 57 year old male physician, a 26 year old female nurse, a 52 year old female preschool teacher, a 41 year old male engineer and a 27 year old male economics student.

5.3 Functional test

Paper prototyping was done for the functional test. Subjects were handed two different ideas to implement a more advanced query system. The first option was a traditional advanced querying system where the user check a box to be handed more options for his query. The second option was a faceted navigation system that became available once a query had been entered.

5.3.1 Traditional advanced querying

Advanced querying was designed in a traditional manner with a textfield for the search query as well as dropdown fields to filter the desired results before they are displayed. This is an advanced search configuration that is very common in search. The paper prototype of the advanced querying system is shown in figure 3.1.

5.3.2 Faceted navigation

Faceted navigation was designed so that after an initial query has been submitted, the user is provided with search options below the search bar to further filter the results should so be desired. For the paper prototype, a horizontal layout for the faceted navigation was chosen to fit the current design of findzebra as much as possible. The faceting system displays facets based on what could

be filtered above the search results and as filters are clicked, the search results are narrowed down. The paper prototype for the faceted navigation is shown in figure 3.2

5.4 Usability testing

A few functional versions of the website were implemented and tested on the five previously mentioned test subjects. The users were shown the the versions of the website in random order and were surveyed with a list of questions. The questions were formed so that the answers would be of the form "very good", "good", "neutral", "bad" and "very bad". The questions asked in this test were

- **Question 1:** The visibility of the function was:
- **Question 2:**The visibility of results and other search options were:
- **Question 3:**The simplicity of using the function was:
- **Question 4:**The simplicity of using the website was:

This test was designed to acquire some insight on whether or not the feature would be noticeable by the user and whether or not it would be obscuring the visibility of the original website. The users were encouraged to think out loud and comments were encouraged.

5.5 Performance evaluation

The final evaluation was done using the diagnostic cases described in section 5.1. The queries and correct diagnoses can be seen in chapter 6. For the chosen subset of cases that have previously proved problematic, it is tested whether or not, using the information provided in the query, it is possible to get the correct diagnosis to be displayed on the first page of results. This means that for the faceted navigation to be successful, the result must be displayed in the first 20 results.

5.6 Faceted navigation or pagination

FindZebra does not provide the user with pagination, i.e. the option to look at results beyond the 20 highest ranked. This was an initial design decision made by the FindZebra team as statistics show that very few users browse to the next page of results and even fewer users give a next page of results any credibility [22, 29]. Faceted navigation is a way to browse more results than the first 20 without giving the user a feeling of having navigated too far away from the initial query, i.e. the user is reformulating his query instead of browsing towards lower ranked pages.

Because of this design, the author found pagination to be a great benchmark vs faceted navigation to see whether faceted navigation can retrieve the correct diagnosis in fewer navigational clicks than pagination. Applying a single filter from the faceted navigation is therefore equal to browsing one page and so on.

In selecting facets, facets should be chosen based on the symptom query, i.e. facets that are related to the query string should be selected. Each applied facet will be backed up, referencing the query.

If the diagnosis does not appear in the first 100 results, pagination is considered to have failed completely, in those cases, the author allowed for experimenting further with facets to see if the disease can be found via facets. This is explained in the results section for those cases. This decision was made based on two reasons, the first being that a medical professional is superior in the selection of symptoms than the author. The second one is that if a patient is not diagnosed correctly, that patient will be interviewed and examined further, opening up a possibility for more symptoms.

Chapter 6

Results

This section contains the results of the evaluations described in chapter 5.

6.1 Functional test

Four out of five subjects preferred the faceted navigation on the grounds that it allowed them to examine the advanced options before applying them as well as being able to apply the options after a regular search query had been entered.

6.2 Usability testing

The results of this test are presented in four tables below, the result for each configuration has a dedicated subsection were relevant comments are added. The first column of each table shows the question that was answered, indicated with a number. The questions are shown in section 5.4.

6.2.1 Configuration 1 results

The results of the user testing for configuration 1 is shown in table 6.1

Table 6.1: Results of user testing for configuration 1

Question	Very good	Good	Neutral	Bad	Very bad
The visibility of the function was	3	2	0	0	0
The visibility of results and other search options were	0	2	3	0	0
The simplicity of using the function was	0	2	1	2	0
The simplicity of using the website was	0	2	1	2	0

Four users commented on that they confused navigation filters with search results. Three of those users expressed that they found the faceted navigation need a visual indication that it was a filtering system. Two users expressed concerns that the navigation was taking up space from the results and mentioned a concern with lower resolution screen.

6.2.2 Configuration 2 results

The results of the user testing for configuration 2 is shown in table 6.2

Table 6.2: Results of user testing for configuration 2

Question	Very good	Good	Neutral	Bad	Very bad
The visibility of the function was	0	0	2	3	0
The visibility of results and other search options were	3	2	0	0	0
The simplicity of using the function was	0	2	1	2	0
The simplicity of using the website was	0	2	3	0	0

Three users commented on that they rarely scroll down to the bottom of results and therefore were concerned with the visibility of the navigation. Two users expressed a concern with using the function to apply multiple filters, having to scroll down to use the function.

6.2.3 Configuration 3 results

The results of the user testing for configuration 3 is shown in table 6.3

Table 6.3: Results of user testing for configuration 3

Question	Very good	Good	Neutral	Bad	Very bad
The visibility of the function was	3	1	1	0	0
The visibility of results and other search options were	2	3	0	0	0
The simplicity of using the function was	2	2	0	1	0
The simplicity of using the website was	0	3	2	0	0

The four users that confused the filters with search results in configuration 1 did not confuse them with the search results. Thereof were two users that tested configuration 1 before configuration 3 that stated that this design was much

clearer in indicating what part of the site was the faceted navigation. One of the users commented on that even though he did notice the filter button, he was concerned about the visibility of the feature.

6.2.4 Configuration 4 results

The results of the user testing for configuration 4 is shown in table 6.4

Table 6.4: Results of user testing for configuration 4

Question	Very good	Good	Neutral	Bad	Very bad
The visibility of the function was	0	0	2	2	1
The visibility of results and other search options were	3	2	0	0	0
The simplicity of using the function was	0	1	1	3	0
The simplicity of using the website was	0	2	3	0	0

One user tested this configuration first and failed to notice the faceted navigation. Three other users commented that this design was cumbersome to use, requiring scrolling and clicking.

6.2.5 Final choice

Comparing the results of the usability testing, configuration three edges out both in the evaluation of the results of tables 6.1 to 6.4 as well as getting the most positive comments from the users. Of special importance is the potential confusion of facets and search results in configuration 1 which means that the actual added visibility of that configuration may come at a price.

6.3 Results of performance evaluation

This section contains the results of the main evaluation that was performed on the deployed product of this project.

6.3.1 Diagnostic cases

This section displays the diagnostic cases used for the evaluation of FindZebra in table 6.5. The first column is a reference number for the query, the second contains the query string, the third the correct diagnosis and the fourth shows where the correct diagnosis is ranked, if it is shown on the first page.

Table 6.5: Results of diagnostic cases without facets

No.	Query	Diagnosis	Rank
1	Boy, normal birth, deformity of both big toes (missing joint), quick development of bone tumor near spine and osteogenesis at biopsy	Fibrodysplasia ossificans progressiva	1
2	Normally developed boy age 5, progressive development of talking difficulties, seizures, ataxia, adrenal insufficiency and degeneration of visual and auditory functions	Adrenoleukodystrophy autosomal neonatal form	2
3	Boy age 14, yellow, keratotic plaques on the skin of palms and soles going up onto the dorsal side. Both hands and feet are affected	Papillon Lefevre syndrome	13
4	Jewish boy age 16, monthly seizures, sleep deficiency, aggressive and irritable when woken, highly increased sexual appetite and hunger	Kleine Levin Syndrome	1

Continued on next page

Table 6.5 – continued from previous page

No.	Query	Diagnosis	Rank
5	Male child, malformations at birth, midfacial retraction with a deep groove under the eyes, and hypertelorism, short nose with a low nasal bridge and large lowset ears, wide mouth and retrognathia, Hypertrichosis with bright reddish hair and a median frontal cutaneous angioma, short neck with redundant skin, Bilateral inguinal hernias, hypospadias with a megameatus, and cryptorchidism	Schinzel-Giedion Syndrome	3
6	6 year old, girl, weight length head circumference below the third percentile, atrophic and hyperpigmented skin lesions, pointed nose, aberrant thumbs with diminished flexion, bilateral glue ears, purulent rhinitis	Rothmund-Thomson syndrome	-
7	13 year old, teenage girl, skeletal muscle defects (muscle weakness), mild mental retardation, ophthalmoparesis	Autosomal recessive centronuclear myopathy (ARCNM)	1
8	14 year old, teenage boy, mild mental retardation, proximal muscle weakness, unable to walk (wheelchair-bound), premature ventricular complexes, ophthalmoparesis	Autosomal recessive centronuclear myopathy (ARCNM)	10
9	35 year old, female, progressive disturbance of gait (difficulties in walking), recurrent diarrhea, bronchitis, growth retardation, mild retardation of psychomotor development in infancy, bilateral juvenile cataracts, swelling of the Achilles tendons, high arched feet, exaggerated tendon reflexes	Cerebrotendinous xanthomatosis (CTX)	1

Continued on next page

Table 6.5 – continued from previous page

No.	Query	Diagnosis	Rank
10	25 year old, woman, conjunctival hyperaemia, interstitial keratitis, moderate bilateral sensorineural hearing loss, tinnitus, dizziness, nausea and vertigo	Cogan's syndrome	6
11	11 year old, boy, severe psychomotor retardation, seizures, strabismus, inverted nipples, dilated cardiomyopathy, hypotonia, wheelchair-bound	CDG (Congenital Disorders of Glycosylation) syndrome type Ic. (Synonyms: Carbohydrate deficient glycoprotein syndrome type Ic, Congenital disorder of glycosylation type 1c (or Ic))	1
12	17 year old, woman, congenital right pulmonary hypoplasia, right hip dysplasia, absence of uterus, rudimentary uterine horn	Mayer-Rokitansky-Küster-Hauser syndrome	3
13	10 year old, girl, thrombocytopenia, splenomegaly, headache, itching rubeoliform rash	Congenital hepatic fibrosis (CHF)	-
14	11 year old, girl, intermittent abdominal pain, mild dorsal scoliosis, low serum phosphate/hypophosphatemia, hypercalcuria, elevated serum 1,25 dihydroxyvitamin D	Hypophosphatemic rickets with hypercalciuria	2
15	4 month old, boy, epistaxis, haematemesis, haematochezia, subconjunctival bleeding, petechiae, haematomas, haemangioma, slightly enlarged liver, elevated serum transaminases	Type I tyrosinemia. (Synonyms: Fumarylacetoacetase deficiency, Hepatorenal tryosinosis/tyrosinemia)	-
16	7 year old, boy, dysmorphic signs, blue sclerae, high-arched palate, bifid uvula, joint hypermobility, muscular hypotrophy, translucent skin, aortic root dilatation, camptodactyly and ulnar deviation	Loeys-Dietz syndrome (LDS) type I	3

Continued on next page

Table 6.5 – continued from previous page

No.	Query	Diagnosis	Rank
17	48 year old, woman, aortic aneurysm, haematoma, translucent skin, bilateral venous varicosities, recurrent wrist dislocations	Loeys-Dietz syndrome (LDS) type II	9
18	8 months old, male, progressive signs of respiratory distress, tachypnea, pulmonary hypertension, tortuosity of aortic arch, facial dysmorphisms	Arterial tortuosity syndrome (ATS)	1
19	5 year old, male, dyspnoea, asthenia, pulmonary hypertension, severe stenoses elongation and tortuosity of pulmonary arteries branches aortic arch sovraortic trunks and iliac arteries, dysmorphic features, joints hypermobility	Arterial tortuosity syndrome (ATS)	1
20	64 year old, male, inflammatory back pain, flares of arthritis, multisegmental spondylitis	Whipple's disease. (Synonyms: Intestinal lipodystrophy, Intestinal lipophagie granulomatosis, Secondary non-tropical sprue)	-
21	70 year old, male, massive hemoptysis, respiratory distress, anemia, hemodynamic instability, renal failure, intense headache, arthralgia, myalgias, ecchymoses over arms and abdomen, acidosis, pleural effusions, blood tinged secretion from lungs	Pulmonary hemorrhage syndrome associated with dengue fever/dengue hemorrhagic fever	9
22	46 year old, female, ptosis, acanthocytosis, history of diarrhea, ataxia, paresthesia	Abetalipoproteinemia (ABL). (Synonyms: Bassen-Kornzweig disease, Homozygous familial hypobetalipoproteinemia (HoFHBL))	5

Continued on next page

Table 6.5 – continued from previous page

No.	Query	Diagnosis	Rank
23	16 year old, girl, persistent diarrhea, acanthocytosis, mild dysarthria, reduced muscle bulk, bilateral proximal muscle weakness, absent deep-tendon reflexes, upgoing plantar reflexes, reduced sensitivity to light, dysdiadochokinesia	Abetalipoproteinemia (ABL). (Synonyms: Bassen-Kornzweig disease, Homozygous familial hypobetalipoproteinemia (HoFHBL))	-
24	teenager, girl, hypotonia, dehydration, acidosis, massive ketonuria, hyperammonemia	Methylmalonic acidemia (MMA). (Synonyms: Methylmalonie aciduria)	1
25	girl, hypotonia, seizures, dehydration, polypnea, acidosis, massive ketonuria, hyperammonemia	Propionic acidemia (PA). (Synonyms: Propionic aciduria, Ketotic glycinemia, Propionyl-CoA carboxylase defficiency)	4
26	27 year old, woman, blindness, obesity, type 2 diabetes, renal dysfunction, chronic pyelonephritis, hypertension, hirsutism, retinitis pigmentosa, cataract	Alstrom syndrome (Alström syndrome)	1
27	17 year old, boy, lysinuric protein intolerance, mild restrictive functional impairment, digital clubbing, atypical abdominal and thoracic pain, ground glass attenuation, interlobular septa thickening, moderate restrictive ventilatory defect, mild anemia, thrombocytopenia, increase in lactate dehydrogenase	Pulmonary alveolar proteinosis (PAP)	1

Continued on next page

Table 6.5 – continued from previous page

No.	Query	Diagnosis	Rank
28	girl, pronounced microcephaly, short stature, psychomotoric delay, distinctive facial appearance, thrombocytopenia, anemia, leukocytopenia, pancytopenia, growth retardation, telecanthus, epicanthal folds, ptosis, infections of the inner ear and respiratory tract, hypoplastic marrow with cellular dysplasia	Ligase IV defficiency syndrome (LIG4 syndrome) (Synonyms: Ligase 4 syndrome)	-
29	5 year old, boy, congenital malformations, malformations of the hands and feet, bilateral strabismus, small tongue, impaired coordination, expressionless face, prominent forehead, depressed nasal bridge, hypoplastic thumbs, bilateral adactyly of the feet, short stature, severe myopia	Oromandibular-limb hypogenesis-Möbius syndrome	6
30	21 year old, female, irregular menses, menorrhagia, hand and foot malformation, ovarian cyst, basic cognitive function	Terminal deletion of chromosome 4q	-
31	Acute Aortic regurgitation, depression, abscess	Infective endocarditis	-
32	oesophageal cancer, refractory hic cups, nausea, vomiting	Gastric Linitis plastica	-
33	hypertension, adrenal mass	Cushings secondary to adrenal adenoma	1
34	hip lesion, older child	Osteoid osteoma	7
35	HRCT centrilobular nodules, acute respiratory failure	Hypersensitivity pneumonitis	-
36	fever, bilateral thigh pain, weakness	Ehrlichiosis	-
37	fever, anterior mediastinal mass and central necrosis	Lymphoma	2
38	multiple spinal tumours, skin tumours	Neurofibromatosis type 1	-
39	ulcerative colitis, blurred vision, fever	Vasculitis	2

Continued on next page

Table 6.5 – continued from previous page

No.	Query	Diagnosis	Rank
40	nephrotic syndrome, Bence Jones, ventricular failure	Amyloid light chain	3
41	hypertension, papilledema, headache, renal mass, cafe au lait	Pheochromocytoma	-
42	sickle cell, pulmonary infiltrates, back pain	Acute chest syndrome	4
43	fibroma, astrocytoma, tumor, leiomyoma, scoliosis	Endometriosis	-
44	pulmonary infiltrates, cns lesion	Aspiration pneumonia and brain abscess (polymicrobial)	-
45	CLL, encephalitis	West Nile fever	20
46	portal vein thrombosis, cancer	Pylephlebitis	1
47	cardiac arrest, exercise, young	Hypertrophic Obstructive Cardiomyopathy (HOCM)	-
48	ataxia, confusion, insomnia, death	Creutzfeldt-Jakob disease (CJD)	2
49	wheeze wt loss, ANCA, haemoptysis, haematuria	Churg Strauss	-
50	myopathy, neoplasia, dysphagia, rash, periorbital swelling	Dermatomyositis secondary to NHL	5
51	renal transplant, fever, cat, lymphadenopathy	Cat scratch disease	2
52	buttock rash, renal failure, edema	Cryoglobulinaemia	-
53	polyps, telangectasia, epistaxis, anemia	MADH4 mutation (HTT + juvenile polyposis)	1
54	bullous skin conditions, respiratory failure, carbamazepine	Toxic Epidermal Necrolysis Syndrome (TENS)	1
55	seizure, confusion, dysphasia, T2 lesions	MELAS	1
56	cardiac arrest sleep	Brugada	-

6.4 Diagnostic cases of interest

Once the website has been tested, a few cases that are of special interest have been detected. Namely, those cases that fail to display the correct disease on the first page of results. These are the following cases: 6, 13, 15, 20, 23, 28, 30, 31, 32, 35, 36, 38, 41, 43, 44, 47, 49, 52 and 56.

6.4.1 Pagination vs faceted navigation on special cases

This section contains the results of the testing of faceted navigation versus pagination on the special cases that fail to display the correct diagnoses on the first result page. Below the results, table 6.6 shows a summary of the results.

6.4.1.1 Case 6: 6 year old, girl, weight length head circumference below the third percentile, atrophic and hyperpigmented skin lesions, pointed nose, aberrant thumbs with diminished flexion, bilateral glue ears, purulent rhinitis

Correct diagnosis: Rothmund-Thomson syndrome.
Pagination: This query produces a diagnoses in result 86.
Faceted navigation: Faceting on *dwarfism* and *short stature* produces the diagnosis on the first page, result 13. Faceting directly on *short stature* does produce the correct result as result 15, however the facet *short stature* is not available as one of the 45 facets available when the query is entered (it becomes available once dwarfism is selected). Both of these symptoms were chosen because of the query description "weight length head circumference below the third percentile" as it describes smaller size.

6.4.1.2 Case 13: 10 year old, girl, thrombocytopenia, splenomegaly, headache, itching rubeoliform rash

Correct diagnosis: Congenital hepatic fibrosis (CHF).
Pagination: This query does not produce a result in the first 100 results.
Faceted navigation: This query can be solved using facets, however the facets chosen are not necessarily logical and can not be applied without more data about the patient. A way towards the result will be shown nontheless as well as a method that can yield a result using the improvement of the system discussed in section 8.2.1.3.

Faceting on *splenomegaly*, *thrombocytopenia* and *platelets* which are all symptoms entered in the query brings the result up to rank 82, it is the lowest ranked result retrieved however, which means that the query term is receiving a very low score for the correct diagnosis. Further, if a symptom removal feature would be used, *malignant neoplasms* could be filtered out, bringing the desired result to rank 20. The final result is however that the faceted navigation fails to retrieve the correct diagnosis and a different system is a subject for different research.

6.4.1.3 Case 15: 4 month old, boy, epistaxis, haematemesis, haematochezia, subconjunctival bleeding, petechiae, haematomas, haemangioma, slightly enlarged liver, elevated serum transaminases

Correct diagnosis: Type I tyrosinemia.
Pagination: This query produces a diagnosis in result 60.
Faceted navigation: Faceting on *hemorrhage* and then *vomiting* provides a result ranked number 13. Further faceting on *kidney failure* brings it up to fourth. All these facets are directly related to the query string.

6.4.1.4 Case 20: 64 year old, male, inflammatory back pain, flares of arthritis, multisegmental spondylitis

Correct diagnosis: Whipple's disease.
Pagination: This query produces a diagnosis in result 91.
Faceted navigation: Despite being able to facet on both *Artheritis* as well as *swelling*, the faceted navigation fails in being able to bring this result higher in the rankings. Further, the article does not have the symptom *inflammation* associated with it, which could have been used for further filtration but instead eliminates the result.

6.4.1.5 Case 23: 16 year old, girl, persistent diarrhea, acanthocytosis, mild dysarthria, reduced muscle bulk, bilateral proximal muscle weakness, absent deep-tendon reflexes, upgoing plantar reflexes, reduced sensitivity to light, dysdiadochokinesia

Correct diagnosis: Abetalipoproteinemia (ABL).
Pagination: This query does not produce a result in the first 100 results.
Faceted navigation: First facet applied is *retinal diseases* due to "reduced

sensitivity to light". Then *mass of body structure* was applied due to "reduced muscle bulk" bringing it to the 12th rank.

6.4.1.6 Case 28: girl, pronounced microcephaly, short stature, psychomotoric delay, distinctive facial appearance, thrombocytopenia, anemia, leukocytopenia, pancytopenia, growth retardation, telecanthus, epicanthal folds, ptosis, infections of the inner ear and respiratory tract, hypoplastic marrow with cellular dysplasia

Correct diagnosis: LIG4 syndrome.
Pagination: This query produces a diagnosis in result 52.
Faceted navigation: Faceting on *anemia*, *developmental delay* and *microcephalies* produces a result on the first page, rank 20. This requires one click more than the pagination.

6.4.1.7 Case 30: 21 year old, female, irregular menses, menorrhagia, hand and foot malformation, ovarian cyst, basic cognitive function

Correct diagnosis: Terminal deletion of chromosome 4q
Pagination: The author was unable to locate this diagnosis in the disease database.
Faceted navigation: N/A

6.4.1.8 Case 31: Acute Aortic regurgitation, depression, abscess

Correct diagnosis: Infective endocarditis
Pagination: This query produces a diagnosis in result 26.
Faceted navigation: Faceting on *hemorrhage*, which is bleeding and thereby associated with "regurgitation" from the query brings the diagnosis to result 2.

6.4.1.9 Case 32: oesophageal cancer, refractory hic cups, nausea, vomiting

Correct diagnosis: Gastric linitis plastica
Pagination: This query produces a diagnosis in result 32.

Faceted navigation: Faceting on *carcinoma* which is a general facet for cancers and matches symptoms from the query moves the result to rank 9.

6.4.1.10 Case 35: HRCT centrilobular nodules, acute respiratory failure

Correct diagnosis: Mycobacterium avium.
Pagination: This query fails to produce a diagnosis in the first 100 results.
Faceted navigation: Faceting *lung diseases*, *pneumonia*, *respiratory distress*, *rapid breathing*, *coughing* produces the correct diagnosis in result 16. This is an extreme case where the search engine provides a very low initial rank for the correct diagnosis. These are symptoms of "respiratory failure".

6.4.1.11 Case 36: fever, bilateral thigh pain, weakness

Correct diagnosis: Ehrlichiosis .
Pagination: This query produces a result ranked 65.
Faceted navigation: Faceting on *fever*, *headache* and then *myalgia* does bring the result to rank 21, which is one away from being on the correct page. Despite being close, with the information provided the case is unsolvable.

6.4.1.12 Case 38: multiple spinal tumours, skin tumours

Correct diagnosis: Neurofibromatosis type 1.
textbfPagination: This query produces a result ranked 48.
Faceted navigation: Applying *neoplasms*, which tumors are and then *carcinoma*, which tumors are associated with bring the result to rank 16.

6.4.1.13 Case 41: hypertension, papilledema, headache, renal mass, cafe au lait

Correct diagnosis: Pheochromocytoma.
Pagination: This query produces a result ranked 42.
Faceted navigation: Faceting *headache* results in rank 8.

Another way to get it to display is to facet on *carcinoma*, renal mass is often associated with cancer as it can be a tumor. Cafe au lait spots can be associated

with cancer as well. This makes carcinoma a valid choice as well.

6.4.1.14 Case 43: fibroma, astrocytoma, tumor, leiomyoma, scoliosis

Correct diagnosis: Endometriosis.
Pagination: This query does not produce a result on the first page.
Faceted navigation This case fails the faceted navigation - the reason is mainly that too many of the facets that "should" lead to his result actually exclude the result due to not being in the list of associated symptoms. A good example of this is that if one facets on *neoplasm*, this article is excluded, despite including both the symptom filters *malignant neoplasm* (too broad) and *malignant neoplasm of the brain* (not available in facet UI). Similarly the article has *endometrial carcinoma* (not available in UI) associated with it but not *carcinoma*, which means that faceting on *carcinoma* again excludes the article which it should not.

Faceting on *malignant neoplasm of brain* and *fibroid tumor* (which is fibroma) does bring the diagnosis to result 1 but neither of those symptoms were offered in the navigation UI due to lack of presence in other articles.

6.4.1.15 Case 44: pulmonary infiltrates, cns lesion

Correct diagnosis: Aspiration pneumonia.
Pagination: This query produces a result ranked 77.
Faceted navigation: Faceting on *pneumonia* which is a symptom of pulmonary infiltrates brings the diagnosis to rank 17.

6.4.1.16 Case 47: cardiac arrest, exercise, young

Correct diagnosis: Hypertrophic Obstructive Cardiomyopathy (HOCM).
Pagination: This query produces a result ranked 70.
Faceted navigation: Faceting *heart disease* which would be logical because of a cardiac arrest and then *cardiomyopathy* which is similarly related brings the result in number 16. *cardiomyopathy* could be exchanged for *cardiac arrest* for the same result, however *cardiac arrest* is not one of the 45 symptoms offered.

6.4.1.17 Case 49: wheeze wt loss, ANCA, haemoptysis, haematuria

Correct diagnosis: Churg Strauss.
Pagination: This query produces a result ranked 24.
Faceted navigation: Faceting on *hemorrhage* (bleeding) which both "haemoptysis", "haematuria" are associated with bring the diagnosis to rank 15.

6.4.1.18 Case 52: buttock rash, renal failure, edema

Correct diagnosis: Cryoglobulinaemia.
Pagination: This query does not produce a the diagnosis in the first 100 results.
Faceted navigation: No sensible facets were found that provided a path to the diagnosis.

6.4.1.19 Case 56: cardiac arrest sleep

Correct diagnosis: Brugada.
Pagination: The query produces the diagnosis as the 100th result.
Faceted navigation: Faceting on *cardiac arrest* and then subsequently on *heart diseases* provides the correct result ranked 20. The choices of facets are obvious in this case as the query is very limited and those are one of the few that provide an exact match to the query.

6.4.1.20 Summary of pagination vs faceted navigation

Table 6.6 shows a summary of the results of the difficult diagnostic cases. Column two shows what rank the diagnoses had, if it was shown in the first 100 results. Column three shows what facets were applied and column four shows what the result number for that case was with the facets applied.

6.4.2 Comparison of navigational actions needed

Table 6.7 shows a comparison of the cases where both pagination and faceted navigation were sucessful in retrieving articles and shows the difference in navigational actions (clicks) needed to retrieve the correct diagnoses articles.

Table 6.6: Results of testing difficult cases

Case no	Pagination result	Facets	Facet result
6	86	Dwarfism + Short stature	13
13	N/A	hereditary diseases + anemia +fibrosis	15*
15	60	hemorrhage + vomiting	13
20	91	N/A	N/A
23	N/A	retinal diseases + mass of body structure	12
28	52	anemia + developmental delay (disorder) + microcephalies	20
30	N/A	N/A	N/A
31	26	hemorrhage	2
32	32	carcinoma	9
35	N/A	lung diseases + pneumonia + respiratory distress + rapid breathing + coughing	16
36	65	fever + headache + myalgia	21*
38	48	neoplasms + carcinoma	16
41	42	headache	8
43	N/A	inflammation + pathogenesis + inflammatory response	2
44	77	pneumonia	17
47	70	cardiac arrest + heart diseases	19
49	24	hemorrhage	15
52	N/A	N/A	N/A
56	100	cardiac arrest + heart diseases	20

Table 6.7: Comparison of navigational actions needed

Case no	Page clicks	Facet clicks	Difference
6	4	2	-2
15	2	2	0
28	2	3	1
31	1	1	0
32	1	1	0
38	2	2	0
41	2	1	-1
44	3	1	-2
47	3	2	-1
49	1	1	0
56	4	2	-2
Total	-	-	-7

Chapter 7

Conclusions

Faceted navigation that uses machine learning to automatically generate and predict navigation groups for the user does produce sensible and useful navigation. Faceted navigation does indeed improve the diagnosis of medical cases where the correct diagnoses does not appear in the initial 20 results. Faceted navigation does better than pagination on average in terms of page navigations needed by the user, saving a total of 7 page clicks for the diagnostic cases where both methods are able to retrieve a correct diagnosis on the display page. There are two cases where faceted navigation can retrieve articles that pagination is unable to retrieve and two cases where pagination can retrieve an article but faceted navigation can. In addition to that there are three cases where neither can serve as an improvement.

Since faceted navigation is a form of query reformulation, the users are not presented with browsing results beyond a first page or a feeling of distancing themselves from the query. Browsing results using faceted therefore should yield more positive response for users than browsing using a more traditional pagination. Users are very unlikely to trust results that do not appear on the next page of result. Therefore faceted navigation is a way to display more results in a controlled manner that gives the user a feeling of control and the results receive credibility.

Chapter 8

Discussion

8.1 Faceted navigation discussion

The faceted navigation feature is considered a success by the author. A good example of where the feature proves very useful would be diagnostic case 35, the correct diagnosis has never showed up on the first page of result for any iteration of the FindZebra search algorithm. The author believes this happens for two reasons, first of all the query is very short and has very specific symptoms, maybe a medical professional could extend the query by adding symptoms of the medical terms mentioned in the query. The other reason is that the article for mycobacterium avium is very short, and therefore does not provide a lot of information for search algorithms. This case can however be solved using the faceted navigation by using only symptoms that are related to the query.

While the faceted navigation may be a success it is not without flaws. Its greatest flaw is probably that the faceting excludes all articles that don't include the symptom that is faceted on. This means that if a patient is showing a symptom that may be unrelated to the disease and a doctor facets on that symptom, the correct diagnosis will be removed.

Another flaw of the faceted navigation is that it does not do anything to the ranking of articles, it only removes articles. This means that there can be a bit

of luck involved with faceting. In order to receive a result page that includes the correct diagnosis the user of the site may require to apply different amounts of facets depending on which facets the user chooses. If there are a lot of diseases in the results that involve the same symptoms and are ranked higher for the query given, the faceted navigation can have troubles bringing the correct diagnosis to the first page. This is apparent in diagnostic case 36.

8.1.1 Other fields for faceting

The faceted navigation provides opportunities for expansion as other fields than symptoms could possibly be used, for example age groups. Some of the diseases are linked to certain ages. This is already being handle

8.2 Further improvements to the FindZebra user experience

This section is dedicated to discussions regarding future improvements possible to the FindZebra user interface that were realised during the testing and implementation of the features that this thesis concerns.

8.2.1 Improvements to faceted navigation

As has been discussed, the faceted navigation is not yet perfect, even though it is believed by the author that in its current form it does provide an improvement for the users.

8.2.1.1 Symptom linking to articles and symptom hierarchy

Case 43 made it apparent that the symptom extraction is not working perfectly. One way to have a possible positive effect on the symptom extraction would be to create a hierarchy for the symptoms in the database. This hierarchy would give symptoms an association, where the most broad terms would have the sub-symptoms that are associated to them and further. For example *neoplasm* is a general term which would then have associated sub-symptoms. *Malignant neoplasm* is a neoplasm and therefore would be linked to *neoplasm*. *Malignant*

neoplasm of brain is a *malignant neoplasm* and would therefore be linked to it, and thereby *neoplasm* as well. This could then be run on the database which would then look for instances of sub and sub-sub symptoms and associate the higher level symptoms to the articles.

8.2.1.2 Ranking facets

Diagnostic cases 6 and 43 are good examples of cases where the faceted navigation does in theory provide a navigational way towards the result but the current system of ranking the facets may be flawed. One of the largest flaws of the current system of ranking the facets is that it favours those symptoms that appear in many articles. A possible way to mend this would be to correct the score for the facet by dividing the number of articles the facet appears in with the total number of appearance for that facet. This means that if a facet is in fewer articles but very prevalent in the search result for the current query, that facet may be ranked higher then in the current ranking system.

8.2.1.3 Removal facets

Another way to improve the faceting which was suggested when the faceted navigation was shown to doctors that are FindZebra collaborators is to allow the user to select a facet for removal, i.e. to filter on a facet in a way that all occurrences of that symptom are removed instead of showed. This method would have helped with the diagnosis of case 13.

8.2.1.4 Possible improvements to user interface implementation

An improvement that could be made to the user experience of FindZebra would be to implement the faceted navigation in a asynchronous way with querying done on the go. This would improve the efficiency of the web application by reducing the queries to the web server as well as help with display of system status by eliminating page reloads. This was not done in this project as the projects purpose was a test of whether faceted navigation with machine learning would aid with diagnosis of difficult medical cases.

8.2.2 Improvements to multiple source articles

In this project, a feature was implemented where articles for the same disease are combined and displayed as a list instead of individually in their respective ranks. A possible improvement to this would be to combine the articles on a database level, this would help with faceted navigation as the symptoms associated with the multiple sources would be combined and therefore each article would have more associated symptoms. That would possibly eliminate some of the cases where a symptom should be associated with an article but isn't.

Appendix A

Faceted navigation placement

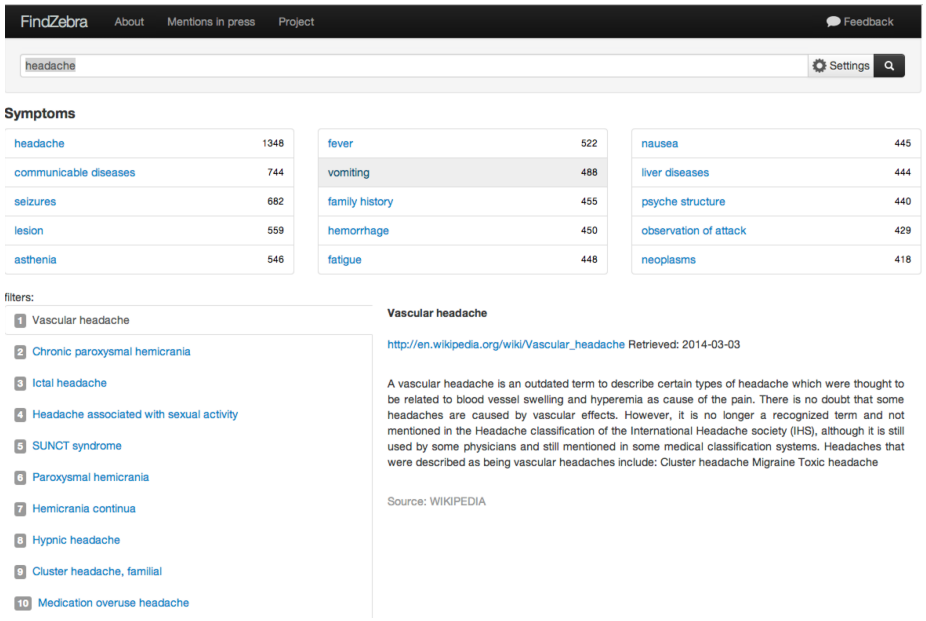


Figure A.1: A screenshot showing the implementation of configuration 1

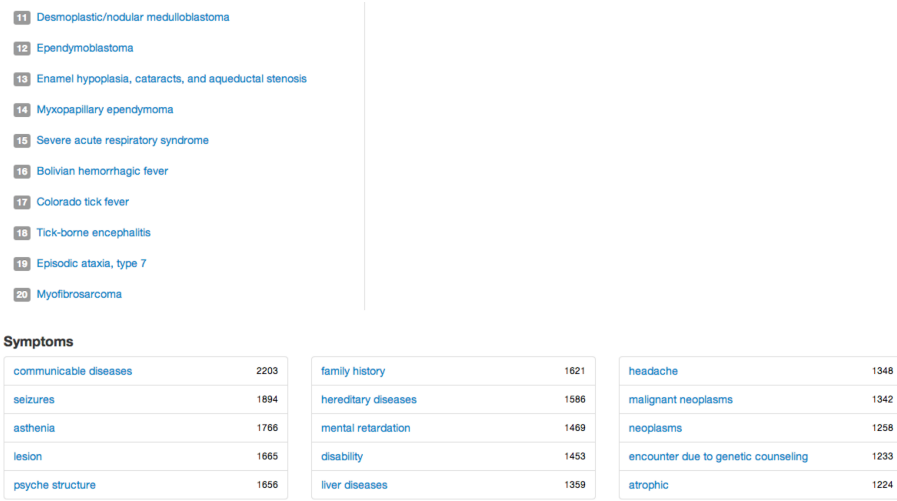


Figure A.2: A screenshot showing the implementation of configuration 2

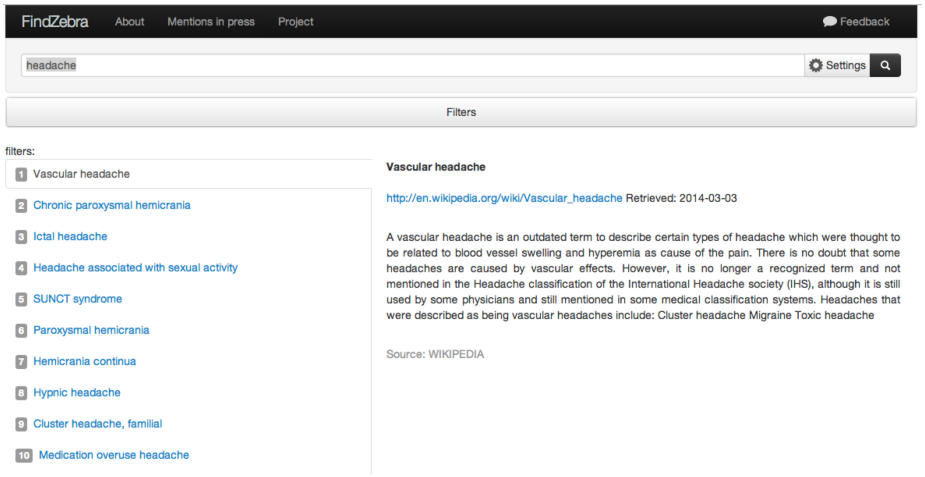


Figure A.3: A screenshot showing the implementation of configuration 3

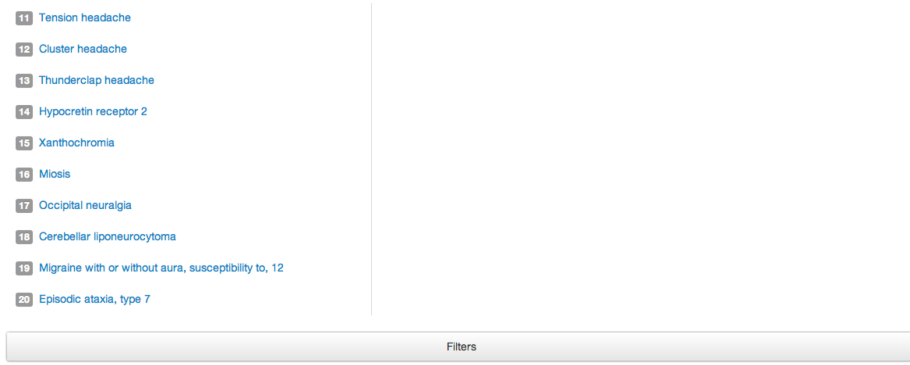


Figure A.4: A screenshot showing the implementation of configuration 4

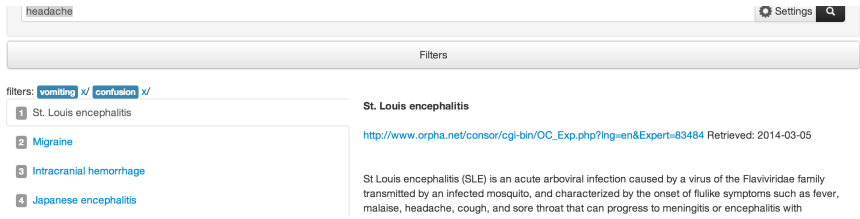


Figure A.5: A screenshot showing the implementation of the status display

Bibliography

[1] Personalized search for everyone. http://googleblog.blogspot.com/2009/12/personalized-search-for-everyone.html, December 2009.

[2] 1 2014. URL www.amazon.com.

[3] 1 2014. URL www.bing.com.

[4] 2014. URL http://clusty.com/.

[5] 1 2014. URL www.ebay.com.

[6] 1 2014. URL www.google.com.

[7] 2014. URL http://www.google.com/landing/now/.

[8] 2014. URL https://developers.google.com/prediction/.

[9] 2014. URL http://www.google.com/trends/.

[10] 1 2014. URL http://www.healthfinder.gov/.

[11] 1 2014. URL www.healthline.com.

[12] 1 2014. URL http://www.imedisearch.com/.

[13] About google instant, 2014. URL http://www.google.com/insidesearch/features/instant/about.html.

[14] 2014. URL http://www.pogofrog.com/.

[15] 1 2014. URL http://www.ncbi.nlm.nih.gov/pubmed/.

[16] 2014. URL http://www.apple.com/ios/siri/.

[17] 2014. URL `http://wiki.apache.org/solr/SimpleFacetParameters`.

[18] 1 2014. URL `http://www.webmd.com/`.

[19] 1 2014. URL `www.yahoo.com`.

[20] Patrick Baudisch, Desney Tan, Maxime Collomb, Dan Robbins, Ken Hinckley, Maneesh Agrawala, Shengdong Zhao, and Gonzalo Ramos. Phosphor: explaining transitions in the user interface using afterglow effects. *UIST '06 Proceedings of the 19th annual ACM symposium on User interface software and technology*, 19:169–178, 2006.

[21] Nicholas J Belkin, Diane Kelly, G Kim, J-Y Kim, H-J Lee, Gheorghe Muresan, M-C Tang, X-J Yuan, and Colleen Cool. Query length in interactive information retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 205–212. ACM, 2003.

[22] Collin Cornwell. The importance of page-one visibility. Technical report, icrossing, 2010.

[23] Anna Divoli, Marti A Hearst, and Michael A Wooldridge. Evidence for showing gene/protein name suggestions in bioscience literature search interfaces. In *Pacific Symposium on Biocomputing*, volume 13, pages 568–579, 2008.

[24] Radu Dragusin, Paula Petcu, Christina Lioma, Birger Larsen, Henrik L Jørgensen, Ingemar J Cox, Lars Kai Hansen, Peter Ingwersen, and Ole Winther. Findzebra: A search engine for rare diseases. *International journal of medical informatics*, 82(6):528–538, 2013.

[25] Susan Dumais, Edward Cutrell, and Hao Chen. Optimizing search by showing results in context. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 277–284. ACM, 2001.

[26] Saul Hansell. Google keeps tweaking its search engine. *New York Times*, 2007.

[27] Eszter Hargitai. Classifying and coding online actions. *Social Science Computer Review*, 22(2):210–227, 2004.

[28] Marti Hearst. Design recommendations for hierarchical faceted search interfaces. Technical report, School of Information U.C Berkeley, 2006.

[29] Marti Hearst. *Search User Interfaces*. Cambridge University Press, 2009.

[30] Hilary Browne Hutchinson, Benjamin B Bederson, and Allison Druin. The evolution of the international children's digital library searching and browsing interface. In *Proceedings of the 2006 conference on Interaction design and children*, pages 105–112. ACM, 2006.

[31] Bernard J. Jansen, Amanda Spink, and Sherry Koshman. Web searcher interaction with the dogpile.com metasearch engine. *Journal of the American Society for Information Science and Technology*, 58(5):744–755, 2007.

[32] Mika Käki. *Enhancing Web search result access with automatic categorization.* Citeseer, 2005.

[33] Mika Käki. Optimizing the number of search result categories. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems*, pages 1517–1520. ACM, 2005.

[34] Bill Kules and Ben Shneiderman. Users can change their web search tactics: Design guidelines for categorized overviews. *Information Processing & Management*, 44(2):463–484, 2008.

[35] Steven Levy. Ted 2011: The 'panda' that hates farms: A q&a with google's top search engineers. http://www.wired.com/2011/03/the-panda-that-hates-farms/2/, March 2011.

[36] Andrew Lipsman, Carmela Aquino, and Stephanie Flosi. U.s. digital future in focus. Technical report, comScore, 2013.

[37] Gang Luo. Lessons learned from building the imed intelligent medical search engine, 2009.

[38] Gang Luo. Design and evaluation of the imed intelligent medical search engine. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, pages 1379–1390. IEEE, 2009.

[39] Harry McCracken. How long did it take for the world to identify google as an altavista killer? *technologizer*, 2009.

[40] Claire Cain Miller. As web search goes mobile, competitors chip at google's lead. *The New York Times*, 2013.

[41] Jakob Nielsen. *Usability Inspection Methods.* John Wiley & Sons, 1994.

[42] Jakob Nielsen. *Designing Web Usability: The Practice of Simplicity.* Indianapolis: New Riders, 1999.

[43] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256. ACM, 1990.

[44] Catherine Plaisant, Gary Marchionini, Tom Bruns, Anita Komlodi, and Laura Campbell. Bringing treasures to the surface: iterative design for the library of congress national digital library program. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 518–525. ACM, 1997.

[45] Annabel Pollock and Andrew Hockley. What"s wrong with internet searching. 1997.

[46] Kristen Purcell Kristen Purcell, Joanna Brenner, and Lee Rainie. Search engine use 2012 search engine use 2012 search engine use 2012. Technical report, Pew Research Center, 2012.

[47] W. Thies, J. Prevost, T. Mahtab, G. T. Cuevas, S. Shakshir, A. Artola, B. D. Vo, Y. Litvak, S. Chan, S. Henderson, M. Halsey, L. Levison, and S. Amarasinghe. Searching the world wide web in low-connectivity communities. *Proceedings of the 11th International Conference on World Wide Web (WWW'02)*, 2002.

[48] David G Thomson. *Blueprint to a billion: 7 essentials to achieve exponential growth*. John Wiley & Sons, 2010.

[49] Anastasios Tombros and Mark Sanderson. Advantages of query biased summaries in information retrieval. *ACM SIGIR*, (2-10), 1998.

[50] Ryen W. White and Dan Morris. Investigating the querying and browsing behavior of advanced search engine users. *Proceeding SIGIR '07 Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007.

[51] Ryen W White, Joemon M Jose, and Ian Ruthven. A task-oriented study on the influencing effects of query-biased summarisation in web searching. *Information Processing & Management*, 39(5):707–733, 2003.

[52] Ryen W White, Mikhail Bilenko, and Silviu Cucerzan. Studying the use of popular destinations to enhance web search interaction. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 159–166. ACM, 2007.

[53] Max Wilson, Paul André, and m.c. schraefel. Backward highlighting: Enhancing faceted search. Technical report, University of Southampton, 2008.

[54] Oren Zamir and Oren Etzioni. Grouper: a dynamic clustering interface to web search results. *Computer Networks*, 31(11):1361–1374, 1999.