Universiteit Leiden

ICT in Business

Building and Evaluating an Enterprise Search Application –
A Case study at UL Transaction Security

Name: Ralph Bos
Date: 03/07/2017

1st supervisor: Kraaij, W.

2nd supervisor: Le Fever, H.

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)

Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

We are living in an information age, where we want to grasp all the relevant information and want it as quickly and convenient as possible. We are all spoiled by companies, such as Google, in the way we think about search and search applications. It is strange, that many organizations lack an enterprise search solution to easily find organizational documents or important information. Another problem is the absence of an easy method for evaluating the performance of an enterprise search application. In this case study at UL Transaction Security we composed requirements for an enterprise search solution in collaboration with employees. These requirements were implemented and finally a framework for evaluation has been developed to measure different aspects of the enterprise search application, i.e., technical performance, query performance, usability & accessibility and user satisfaction. We used the evaluation framework to evaluate the performance of two enterprise search applications.

Keywords: Enterprise Search, Information Retrieval, Enterprise Search Evaluation, Enterprise Search Case Study.

Acknowledgements

First I would like to thank UL Transaction Security, especially Henrique di Lorenzo Pires for the opportunity he gave me in doing a project at their organisation. Second I would like to thank all the UL Transaction Security employees for their lasting support. In these six months I have got to know a lot of enthusiastic people I would not soon forget. They always motivated me and were eager to provide me with feedback on how to improve this project. I would like to highlight Paul Koutinas for all his thorough feedback and joyful meetings.

Furthermore I would like to express my gratitude towards Wessel Kraaij for being my supervisor and all the lively discussions we had about this thesis and for letting me gain new insights and knowledge in the information retrieval field.

Last I would like to thank all my friends and fellow students who supported me during the past six months, I could have never done this without you.

Table of Contents

1 Introduction

In this section we will explain more about the background of this research. Here questions will be answered, such as, where is the research performed? What is the general topic of this thesis? A motivation for the chosen topic will be given, along with the research objective. Further we discuss how relevant this research is, when comparing it with previous research done in the field of enterprise search. After that, research questions will be introduced together with the scope of this research. Finally a thesis outline will be given for further reading.

1.1 Background

Employees spend a lot of time searching for information, but are not always successful in finding the right information. A report from IDC from 2005 [1] showed, that people spend nine to ten hours per week searching, but fail at least half the time in retrieving the desired result. This includes external information, i.e., information that the organization does not own, by using search engines to explore for information online, and internal information, i.e., information that the organization owns, e.g., presentations, reports, project folders.

The phenomenon to search within an organizations' data is called: Enterprise Search [2]. With an organizations' data we mean digital data, that the company owns, e.g., database records, shared documents and intranet.

Information retrieval is an important aspect in organizations, especially now, when organizations are data driven, with data stored in data centres. Because there is an abundance of data, problems may arise, when a specific piece or pieces of information cannot be retrieved, or is rather difficult to retrieve, because most information is unstructured. Most organizations are knowledge driven, therefore it is a huge problem, when information cannot be retrieved. To give an example, if information from past projects cannot be retrieved, projects, that have been done before, need to be completely redone, this results in unnecessary costs. Second, many employees do not know, where to search, because an organization often has multiple information sources, e.g., e-mail, internal network storage, intranet and social media.

To resolve this problem, organizations can develop an enterprise search engine, which lets them search throughout the entire company's information sources. The performance of an enterprise search engine must be measured, in order to evaluate whether the implementation was successful. However, an enterprise search engine is not a one-fits-all solution. Every organization is different, each enterprise search engine is a custom product, this results in different implementations.

Even if every enterprise search application is different, there are methods to evaluate the performance of an enterprise search engine, because in the end, a manager wants to know, if the enterprise search application adds any value to the organization.

This research is a case study, that has been done in collaboration with Leiden University and UL Transaction Security [3]. In UL Transaction Security there was a need for an enterprise search application.

1.2 Research objective

The aim of this research is to contribute to the field of enterprise search with the main intent to evaluate the performance of an implementation of an enterprise search application. A framework for implementing and evaluating an enterprise search engine will be made. This framework is useful for organizations, that consider an enterprise search application and want to measure if the enterprise search application works accordingly.

What is new, is that this is a case study, where an Enterprise Search Engine is being designed, according to certain business requirements being derived from a user survey. The performance of the enterprise search application will be measured with an evaluation framework, where a final grade will be given on how well the implementation of the enterprise search application is performed.

1.3 Research Relevance

The field of enterprise search is a special branch of information retrieval, which focusses on organizations. Now problems around Enterprise search occur more often. Since it is cheaper to store data, organisation have more data available which leads to an increase of data retrieval problems in organizations.

There are already studies being done in the field of enterprise search. Much can be found in the book Modern Information Retrieval, where there is a chapter dedicated to Enterprise Search, written by Hawking [4]. Hawking found that the evaluation of enterprise search is not different, from other types of search. Another important aspect, is that enterprise search evaluation requires the collaboration of an organization, because an Enterprise Search application is tailored for an organization. The employees of the organization need to participate in the evaluation of the Enterprise Search application, because they will be the main users of the search application, this differs from the evaluation a web search platform, where we can use random participants in evaluation.

Other relevant research was done by Saastamoinen and Järvelin [5], who looked into the search behaviour of employees at a specific company, this contained the search frequency and analysis of the queries being used. Performance of the enterprise search application was not studied. Another study, where Sirotkin [6] evaluates the performance methods of web search applications, where he criticizes current methods and defines his own metrics, and researches certain aspects of user behaviour, but this is in the field of web search applications, which is somewhat similar to enterprise search, but is not the same.

A book that bundles enterprise search aspects with a more business oriented view by White [7] introduces best practises on how to implement and evaluate an enterprise search application for managers. However many of the statements of White are not supported by research and are more general ideas than findings.

1.4 Research Questions

The research question that will be answered in this thesis will be:

How to evaluate an implementation of an enterprise search application?

To answer this question, there are sub-questions, that need to be taken into account, such as:

1) *Does the enterprise search application work accordingly (objective view)?*
2) *What is the opinion of employees on an enterprise search application (subjective views)?*

1.5 Research Scope

The scope of this research is a case study at UL Transaction Security Advisory Europe. In this department, sometimes information cannot be retrieved properly. There are various reasons for this, such as bad performance of current search engines or the abundance of files. This research will focus on what the added value will be of having another search mechanism, in particular an enterprise search application. The purpose of this application is to increase the retrievability of information and increase the speed with which information can be retrieved. The enterprise search application will be developed with open-source software and the application will be evaluated in the UL Transaction Security environment with a purposely designed framework, which will subsequently, for comparison reasons, also be applied on a different search application, in a Dutch research institute.
In this research, we will not focus on search behaviour, but focus on the performance of the enterprise search application and how satisfied the users are with the performance of the enterprise search application.

1.6 Thesis Outline

In the first section an introduction is given, which provides a background on this research, including research objectives, research questions and the scope of the research. The second section includes a theoretical framework, that explains more about enterprise search and information retrieval in general. In section three we continue with the research methodology that is being used to conduct this research.. In section four we describe how the enterprise search application is being implemented, including the requirements, design and tools used for the implementation. Section five will contain the results, which consists of a search behaviour survey and results of the evaluation of the implemented search engine at UL Transaction Security together with the performed evaluation at a Dutch research institute. A discussion about the results can be found in section six and we end with a conclusion and future work in section seven.

2 Theoretical Framework

In this chapter a theoretical framework will be presented, that includes theory needed to understand the aspects of information retrieval and enterprise search. First there will be theory presented about information retrieval in general, second the concept enterprise search will be explained and third, methods for evaluating the performance of enterprise search and search in general will be presented.

2.1 Information Retrieval

Information retrieval can be anything, that a user does to retrieve information, that is stored somewhere. This can be using, e.g. a web search engine to lookup some information, or look in your wallet to see how much money you have in your pocket. Of course now there is a large shift of how we handle information. In the past information was physical, i.e., books and other written information. Now most of the information is stored digital, i.e., on servers or other digital storing devices.

The term information retrieval is first mentioned in the academic field by Calvin Mooers in 1950 [8]. Information retrieval, how we are going to use the term throughout this paper is defined by Manning et al: "Information Retrieval is finding material (usually documents) of an unstructured nature, that satisfies an information need from within large collections, usually stored on computers."[9] Meaning that we focus on how to retrieve digital information. If we relate this to a business example, it can be to search for information about a past or similar project. If this information is present, a project does not have to start from scratch, which prevents unnecessary costs.

2.1.1 Precision and Recall

Information retrieval has some main concepts, that need to be understood. First, there is an information need, where there is a need for information looking at it from an user's perspective. Without an information need there is no need for information retrieval. A user, who is searching for information wants to fulfil their information need. Here is where the term relevance steps in. Relevance can be seen as the fulfilment of the information need[9]. Relevance is subjective, because it can differ per user. Not every user may find a specific document relevant, e.g., some users are not content with the results presented by an information retrieval engine.

Precision and recall are widely used metrics for measuring the performance of an information retrieval engine. Precision and recall are introduced by Cleverdon and Keen in 1966[10]. Precision is all the relevant retrieved results when comparing it with all the other retrieved results of a query.

$$Precision = \frac{relevant\ retrieved\ results}{all\ retrieved\ results}$$

Where recall is, all the relevant retrieved results of a query, when comparing it with all the relevant results, that are present in the system, e.g., a database.

$$Recall = \frac{relevant\ retrieved\ results}{relevant\ results\ in\ system}$$

2.1.2 Indexing

When looking for a specific query in a collection, where a collection is referred to as a set of documents, it can be very time consuming to look into each document in a collection and then looking line for line, if a specific query has a match. One way to solve this problem is to index documents. In information retrieval, using an inverted index has become the standard. This approach maps terms from documents to a database. It sorts al the terms alphabetically and counts its occurrences [4]. An example can be found in Figure 1.

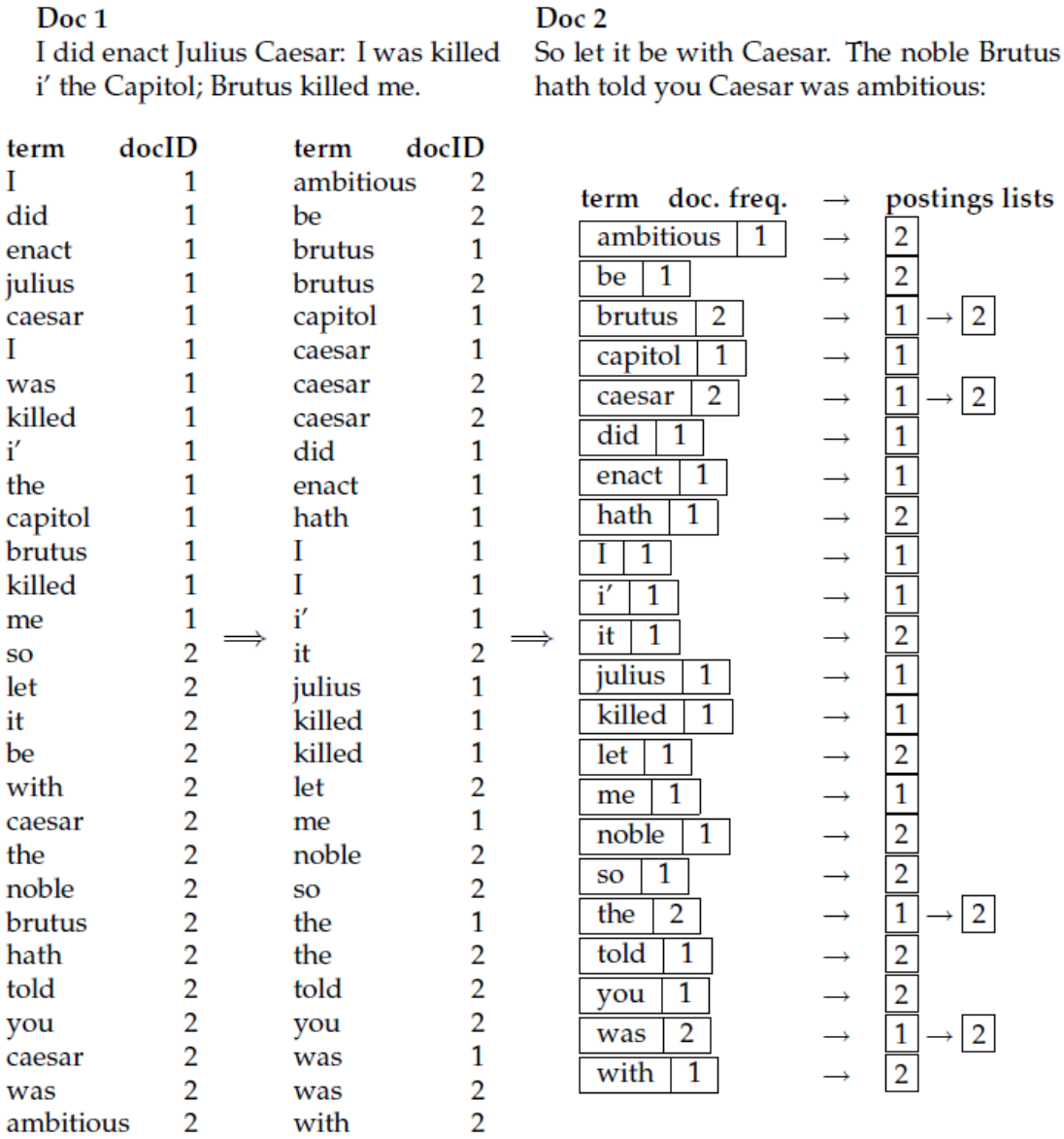

Figure 1: Example of Inverted Indexing [9].

2.1.3 Vocabulary

All the terms, that are present in a collection are called a vocabulary [9].Each term is being mapped to a location in the dataset, as can be seen in Figure 1. To build a vocabulary it can be efficient to ignore stop words. Stop words are words, that are commonly used, e.g., a, an, this or that. Ignoring stop words is more efficient in terms of storing and decreases seek time.

2.1.4 Stemming and Lemmatization

Indexing a document can be quite troublesome. There is the case, where you have one word, but with different grammatical notations, e.g., discuss, discussion, discussed, all have the same stem "discuss". To make sure these different grammatical notations are being mapped to the same term, a stemming or lemmatization algorithm is used. In a stemming algorithm, such as Porter's Stemming algorithm[11], heuristics are being used to chop the end of a word. What remains is the stem.

Different from stemming is lemmatization. Where stemming chops endings of words, by looking at the patterns in the ending of a word, lemmatization analyses words based on known vocabulary, shortens these words and maps them to the right term in the database.

2.1.5 Search Types

We already talked about information need, but there are also different search types. We need to keep in mind, that not every user has the same search intent. There are three basic search types, defined by Broder[12]. Navigational, transactional and informational. In navigational search, users are exploring a topic, they do not know, what they are searching for yet, but they use search in an exploratory way. In transactional search, the main intent is to purchase something, or e.g., renew a license. The last type of search, is the most common type of search, i.e., informational search. Here the main intent is gathering information, that helps to get a work task done.

2.2 Ranking results

Getting results, when searching for information in an information retrieval engine can be quite satisfactory. What most people want, is that the most relevant result appears on top of the presented list. Here we will present some theory on how results are being ranked, and what you can do to improve ranking. One way to improve ranking is to use scoring methods. Giving a higher score to specific characteristics of a result, a higher score means higher relevance.

2.2.1 Metadata

First, there is metadata. Metadata can help to get more information on a document. E.g., if you are looking for a document made in 2014 with the words "Mobile Payment", but the year is not mentioned in the text at all, but is present in metadata. Here metadata can help to improve rankings. Second, by going through the text and filter for specific words to put them in a separate field in the database. An example is looking for countries in a text, this can be relatively easy, because a list of countries is very straightforward. The list of countries will be compared with the terms in a text. If there is a match with a country in the text, this match will be put in a database field.

2.2.2 Bag-of-words model

The bag-of-words model, as being described by B. Ribeiro-Neto and R. Baeza-Yates[4] puts all the terms, that are present in a document in a so called vritual bag-of-words. Where the amount, and exact ordering of the terms are being ignored. Here similarity can be found in the smilarity of the bag-of-words, by comparing the bag-of-words of different documents.

2.2.3 Vector Space Model

Being introduced in 1958 by Luhn[13] and further developed by Salton & Buckley, the vector space model is a widely used scoring method in information retrieval. Here documents can be seen as a weighted vector and the total collection of documents can be seen as the space, in

which the vectors exist. In the vector space model, documents can be ranked, based on a specific query, or be clustered together. Each term in a document has an axis in the vector space. It is difficult to compare the terms between documents, because not every document has the same length. Therefore cosine similarity is being applied to their vector representations (effectively measuring the angle between the vectors), here the length of the document is been taken into account for adding weights. By looking at the term distribution and the length of the document[9].

$$sim(d1, d2) = \frac{\vec{V}(d1) \cdot \vec{V}(d2)}{|\vec{V}(d1)||\vec{V}(d2)|}$$

2.3 Enterprise Search

A special form of information retrieval within organizations is being called enterprise search. Enterprise search can be seen as searching through all the information, that is present in a company [2] Before we mentioned the term collection. In Enterprise Search a collection is all the digital data, that is present in the company. Off course employees use Google for web search, but you do not want all valuable company information, such as projects and business proposals, on the open web.

An organization has many sources of information, e.g., intranet, network shared folders, organization's social media. The essence of an enterprise search system aims to find all the information, that is available in all the sources of an organization.

How does an enterprise search system look like? David Hawking gave an overview in a chapter about enterprise search in the book Modern Information Retrieval[4]. Here we have the different stages: gathering, extracting and indexing, these can be found in Figure 2.

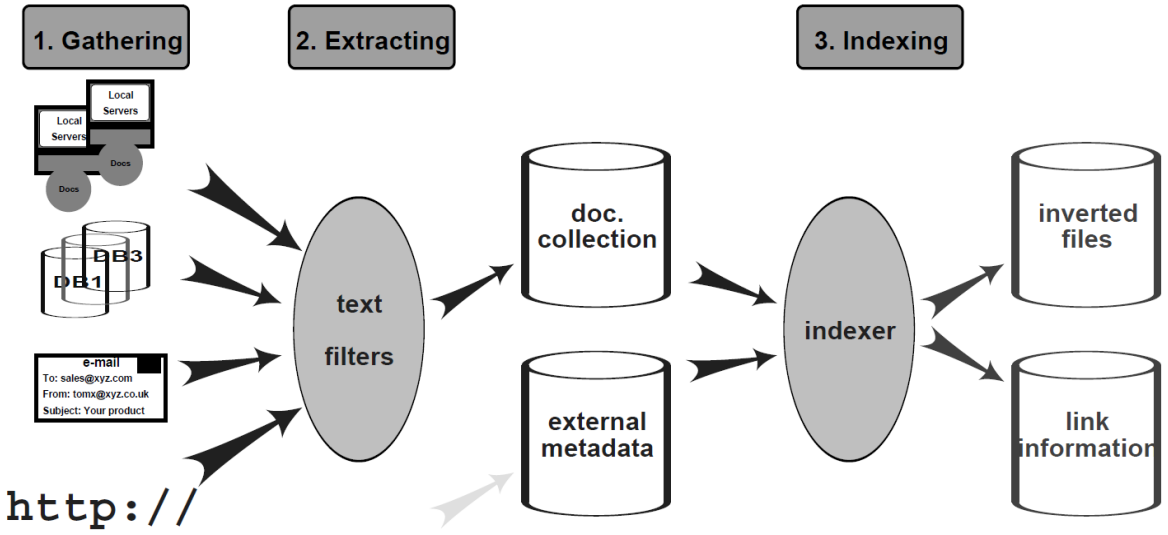

Figure 2: Enterprise Search System [4].

2.3.1 Gathering

The first part of the enterprise search system is the gathering of information components, that need to be part of the search system. This can be searching through a file system, crawling the company intranet or crawling e-mails. It is necessary to keep security in mind. Not every piece

of information can be seen by everybody. Therefore it is important to know beforehand, what to include in the gathering stage.

2.3.2 Extracting

Second is the extracting and filtering stage. In this stage all the information is being extracted, from the gathered sources. By building an enterprise search system, the system architect defines what is being extracted. Usually this is been done together with the client. Things that include in the extracting stage are, e.g., author, document body, document title, description, date of creation and application type. These fields can contribute to a better search experience. Another option is to extract subsets of elements in the text. What if you want to include all the countries, that are mentioned in a document? Then you have to extract them from the body of the text and map them to a database field.

2.3.3 Indexing

After the gathering and extraction phase, we continue with the indexing phase. In this phase all the files are being indexed and all the information is mapped to their corresponding field in a database, by using inverted indices[14]. The indexing phase can take a while, depending on the size of your data collection, or indexer that is being used. Especially, when you are indexing all of the organization's data.

2.3.4 Result Representation

Another important part is result representation, because employees in an organization need to use the enterprise search system, to perform day to day tasks, the user interface and operations needed to perform a single search need to be as simple as possible. One downside, is that everyone in an organization has Google web search as their baseline, meaning they have very high expectations for an enterprise search application. There are features that can be added to increase the usability of the enterprise search application, e.g., adding facets, query completion and spelling correction. Do not forget, that employees want to have the most relevant result on top, therefore ranking is extremely important, especially in an organization, where time pressure plays a part.

2.3.5 Research in Enterprise Search

It can be very challenging to research enterprise search. Firstly, it is difficult to gather datasets, due to confidentiality We do not want company information to fall in the wrong hands. Secondly, since every organization differs, there is not one enterprise search solution, that is applicable to every organization. Thirdly, We do not know, if the search solution works in other companies. In the past, there has been a data collection made public, with its purpose for testing and performing research in the information retrieval field. This data collection is the TREC Enterprise Track collection [15]. There are others, who have studied enterprise search, but with its main focus on user search behaviour, e.g., Saastamoinen and Järvelin[5] studied, what were the search tasks employees performed at different organizations. A study done Hertzum and Pejtersen [16], where they did a case study at two companies and looked at the information seeking behaviour of the employees. Same as Hansen and Järvelin[17], where they performed a case study in a Swedish patent registration office, to study the user behaviour of the employees.

2.4 Evaluation of Enterprise Search system

How do we know, that the enterprise search system performs as it is supposed to do. An evaluation of the enterprise search system must be done. As with every type of search, the most important part of the search system, is that the information need of the end-user is being fulfilled. Users, and in the case of enterprise search employees, play a big role in the evaluation of the enterprise search system. We need to understand, that not every user is the same or has the same information need. Evaluation is an important aspect, because after evaluating, the enterprise search tool can be made better, to fulfil the needs of the user. According to Martin White[7], there are five components of enterprise search evaluation:

1) Technical performance
2) Query performance
3) Usability and accessibility
4) Search satisfaction
5) Business impact

We will elaborate more on the five components mentioned before, in combination with evaluation methods, that can be used to assess these components.

2.4.1 Technical Performance

The technical performance of the enterprise search system must be measured. Here metrics are being used, such response time, i.e., the time it takes for the system to present results. Not only the response time, but the consistency of responses is important.. Another aspect, is freshness of the results. When new information or documents are being added to the data collection they must be indexed by the system as soon as possible. Otherwise, users, who are searching for this new information do not get the correct results.

2.4.2 Query Performance

Different from technical performance is query performance. By looking at the query performance we measure how well the enterprise search system is behaving, when a query is used for input. Are the results, that are being presented relevant? Here are metrics being used, such as recall and precision. This requires input from users, who know the content, to assess if a document is relevant or not. By analysing the search logs, that are being saved by an enterprise search system, we can see how many results are generated, compared to the use of all the queries by the user. If there is a query, that is frequently used, but has zero or low results, this can be an indication, that there is information missing, or a need for information, that is not yet present in the organization. Search logs can also be used to understand the information need of the employees.

2.4.3 Usability and Accessibility

Another aspects is usability and accessibility. Users want the enterprise search tool to be as simple as possible. With a clean interface, that is easy to use, to get the most relevant result, without going through multiple menus. One way to measure this, is to do a survey, to understand, what the user thinks of certain components of the enterprise search system. Next to surveys is another method to obtain a measurement of the usability to perform interviews with users, who have spent some time with the enterprise search tool. The main purpose of this survey or interview is to evaluate, how the user thinks about aspects of usability and accessibility, i.e., user interface, different aspects that are present in the search tool, query completion.

2.4.4 User Satisfaction

It is great, if your enterprise search system works as it should work, if it has a high usability and accessibility, top performance and query processing, but if users are not satisfied, they are probably not going to use the search application. The users, who need to use the search system need to be satisfied. To measure satisfaction surveys and interviews can be done, e.g., with a likert scale, asking them how satisfied they are with the search tool. A different approach is, as done by Thomas and Hawking [18] is to evaluate by comparing two different search tools. One search tool can be the baseline, i.e., the search tool that was there before the enterprise search solution and the other search tool can be new enterprise search implementation. Satisfaction can be measured, when users do a side-by-side comparison and use the same query in both search tools. An idea from White[7], is to implement a feedback box in the search application, where users can input their feedback to improve the enterprise search tool. Processing feedback can increase user satisfaction.

2.4.5 Business Impact

The last component is business impact. Enterprise search is about a search application in an organization, meaning we want to measure the impact the application has on the business. To measure the impact company KPI's can be compared before and after the implementation of a search tool. E.g., if employees now spend less time on searching for information, because the right results are being showed, they have more time for their work tasks. If the information retrieval rate of a company is higher, i.e., no more documents about previous projects are lost, it saves money, because an employee does not have to start a new project from scratch, that is similar to an old project.

3 Research Methodology

In this section we elaborate on what research methodology is being used to perform this research. First we explain more on the research approach, second we explain how the literature review is performed. After that, we continue on how we used surveys to gather data. After that we talk about how we researched the implementation part of the enterprise search engine and finally we discuss, what evaluation methods we used to measure the performance of the enterprise search tool.

3.1 Research Approach

For this research we used a research design framework, as described by Michael Sedlmair et al. [19]. This method, also called the nine-stage framework, divides the nine stages into three main stages. Here the three main stages are the precondition phase, consisting of the learn, winnow and cast phase. The core phase, that includes the discover, design, implement and deploy phase. Last the analysis stage, which includes a reflect and finally a write phase. The nine-stage framework can be found in Figure 3. In our research we used an iterative approach for each stage, meaning we did multiple phases at the same time, but not skipping any phases.

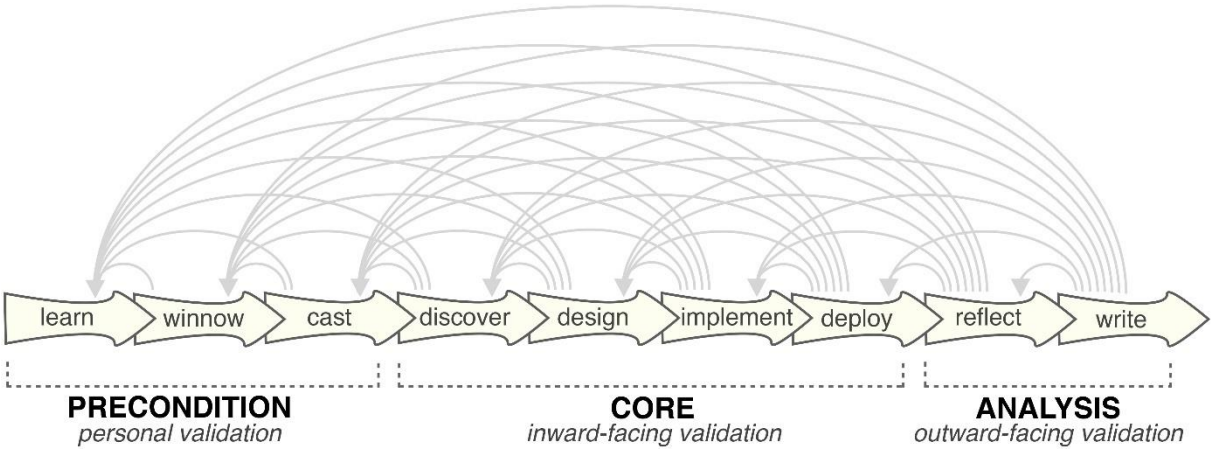

Figure 3: Nine-stage framework [19].

3.1.1 Learn

In the learn stage, we first searched for the keywords “enterprise search” on Google Scholar, and later we narrowed down our search to “enterprise search evaluation” and in general “information retrieval evaluation”, because Enterprise Search is a subfield of Information Retrieval. There is a chapter on Enterprise Search written by David Hawking in the book Modern Information Retrieval[4], where literature and relevant experiments are being described. Instead of a dedicated chapter, there is a whole book, that is about Enterprise Search[7], by Martin White, where it is written from a business perspective.

3.1.2 Winnow & Cast

The purpose of the winnow and cast stage is to select collaborators and identify the roles of these collaborators. Since this is a case study at UL Transaction Security, the collaborators and roles are already known. UL Transaction Security is the user of the enterprise search engine.

3.1.3 Discover
In the discover stage we analysed the problem, the current search situation at UL Transaction Security. Here small interviews were being conducted to analyse the problem and a survey has been send to employees, with both open and closed questions. This survey can be found in appendix A

3.1.4 Design
After the problem was being clear, we began with the design of the enterprise search tool. The same survey, that we talked about in the discover stage is being used to extract requirements, as well as talking to employees about their search experiences. With this method we created a baseline, from where we can start the implementation.

3.1.5 Implementation
This research focusses on a case study, where its main goals is to implement an enterprise search engine and to evaluate the same enterprise search engine. For this implementation the current search behaviour and information needs were being assessed and multiple design options were being considered for the implementation. An overview of the implementation and design choices can be found in chapter four.

3.1.6 Deploy
In this stage we presented prototypes of the search engine to different employees. Here we gathered feedback to improve the enterprise search engine prototype. An iterative approach is used to constantly deploy prototypes and gather feedback. After a few iterations the enterprise search tool will be presented to a larger public and will be evaluated.

3.1.7 Reflect
In the reflect stage we evaluated the implemented enterprise search engine. We need to measure if the enterprise search engine is meeting the users expectations, by evaluating it on a user and technical level. By doing so we have developed an evaluation framework.

3.2 Evaluation Framework
A framework is made to present if the enterprise search engine performs accordingly. The framework consists of the five enterprise search evaluation components, being introduced by Martin White [7]. For each component a scoring method will be used, to address how a separate component performs. An advantage of a separate scoring method can be, e.g., that recommendations on how to improve the enterprise search tool can be given on a component level, where the overall performance of the enterprise search tool can be increased if such recommendations are being implemented. Here we present again the different components, that will be used in the evaluation:

1) Technical performance
2) Query performance
3) Usability and accessibility
4) Search satisfaction
5) Business impact

Here we explain what metrics we use for assessing the different components, that are being used to evaluate the enterprise search engine.

3.2.1 Technical Performance

In the technical performance stage, we use metrics to assess the technical performance of the enterprise search application. We assess response time and freshness.

3.2.1.1 Response Time

Does the system respond accordingly. Does the user feel, that there is any lag, when performing search? Response time can be measured by looking at the backend of a search application, here a grade from one to five will be given to the metrics, where one means that we have to wait for longer than one minute and five means, that we have instant response time. Nowadays we can expect from search engines, that they have instant response time. To get to a proper grade we examine hundred different search queries and measure the response time of the system, an average of the hundred searches will be presented.

3.2.1.2 Freshness

We already talked about freshness of a search application. To measure freshness we present a one to five scoring range, where five is the highest score possible. Here a five can be related to instant automatic indexing, where in the ideal situation we want to find a file in a search application as soon as the file is being made and one is where the indexing is only done once a week.

3.2.2 Query Performance

For evaluating the query performance, classic metrics are being used, i.e., precision and recall. Another aspect of query performance is the presence of a search log and query completion. All the different aspects of the query performance are being rated. The final grade of the query performance component is the total grade, by the amount of aspects.

3.2.2.1 Precision

To measure precision we let users send queries to the enterprise search system and ask if the presented results are relevant. Here employees know best, if the results are relevant, due to domain knowledge of the organization's data. Users may use their own queries, because we want to use queries, that would represent the use of the search application in a real working situation.

For each user we let them use the search application and let them write down five search queries they are using. Assessing all the results, would take too much time, therefore we use a cut-off of the first twenty results. These results will be assessed on relevance by the user. We calculate the precision for each query, and in the end we calculate the average precision of the system.

3.2.2.2 Search log

A rating will be presented on certain aspects of the search log. First we check, if there is a search log. Second we must know, if click signals are being used to see if users actually click on the presented result. It can be a design choice to implement click signals. We always want the most relevant result on top, but not every user wants the same result on top, due to differences in information need. Here a grade will be given from one to five, to determine the performance of the search log and click signals. Where one is very bad and five is very good. One can be given if there is no search log at all, where five represents the presence of a search log, that is fully searchable and has optimal performance.

3.2.2.3 Query completion

Here it will be assessed if there is a form of query completion. If there is query completion, the accuracy must be measured. With accuracy we mean, does the query complete itself in a way that fits the user's information need. Here the five queries, that are being used in the precision and recall test will be used to evaluate the query completion. Also measure if the system gave a right prediction of completing the query. Grading is done by assessing the performance of the query completion function. Here a five point Likert scale is used, where one is very bad and five is very good.

3.2.3 Usability and accessibility

For evaluating and measuring the usability of the search application the System Usability Scale (SUS) test by Brooke[20] is being used. The SUS test has a scoring method from 0 to 100, which we can easily implement in our final score for the enterprise search engine. See Appendix B for the contents of the SUS test.

3.2.4 Search Satisfaction

Different from the usability of an enterprise search system is the search satisfaction, where search satisfaction is defined by Huffman and Hochster[21] as follows: if the information need of the user is addressed, then the user is satisfied.

To measure search satisfaction, we need to measure if the information need, of the users of the enterprise search engine at UL Transaction Security is fulfilled. Turpin and Scholar[22] showed, that precision and recall of a system fail to assess search satisfaction. Therefore to measure search satisfaction, a user survey is filled in to score how satisfied users are with the search. As a scoring. See Appendix B for the survey, that has been given to employees to measure search satisfaction.

3.2.5 Business Impact

In this research, we will not measure the impact, that the enterprise search tool has on the business, because this is not in the scope of this research. To measure the business impact an enterprise search tool must be running for months, for noticing long term differences. It is recommended to include this component, when evaluating an enterprise search tool. An organization's KPI's can be used to measure the business impact of the enterprise search solution.

3.3 Survey

In this research surveys are being used as a research method. Having direct access to employees of UL Transaction Security helps to get participants. The advisory group of the Leiden office counts sixty people. Where we can use this group to get survey responses. Surveys are being used to create a baseline and as a form of evaluation on different components. The surveys are anonymous, and a mix exist of open and closed questions, here its main focus is on closed questions, to measure search usage. Closed questions are automatically put in graphs, were open questions were analysed and grouped on similarity. Before sending out a survey, an iterative approach of survey design is being used to increase survey response.

3.4 Validation

We only have one organisation where we did an approach to a design method for implementing and evaluating a search application, because this is only being done at UL Transaction Security we want to validate our evaluation framework by applying it to another enterprise search application. Here we will evaluate a search application in a Dutch research institute.

4 Implementation

In this chapter we discuss the implementation of the Enterprise Search Application at UL Transaction Security. We explain how the process of implementing went, the design choices that were considered and show the final version of the application where the evaluation framework is being applied.

4.1 Process

First, before even thinking about an implementation of an enterprise search engine, we analysed the current search systems and search behaviour by talking to different employees of UL Transaction Security and by conducting a survey. This step is necessary, because in the end the employees will be the main users of the search tool. The survey can be found in Appendix A. In this survey employees could add comments on each question, if they felt things were missing in the survey. In total forty advisors participated in the survey. Questions were asked to assess the usage of search on different mediums and asked to rank the performance and usefulness of each information retrieval medium.

From the survey we analysed main intent of information retrieval for each medium, the information need and components, that must be present in the enterprise search tool. An element in the survey was, that we asked forty employees, what they felt was required in a search application in the form of an open question. The results of this survey will be presented in chapter five. After the analysis was done, requirements were made for the enterprise search tool.

For the actual implementation we used an iterative approach, where employees gave feedback after each iteration. After two iterations the enterprise search application was finished.

Figure 4: Magic Quadrant for Enterprise Search [23].

4.2 Requirements

Requirements for the Enterprise Search application are being derived from the search behaviour survey, see chapter five for the results. Employees have high standards, when it comes to a search application, they expect an Enterprise Search engine to be as good as, e.g., Google.com[24], with a clear user interface, keyword completion, spelling checker, and advanced search options. Users expect, that the most relevant result will always be on the first page. The following requirements have been derived:

- Simple User Interface;
- Fast Indexing;
- Multiple data sources support;
- Scalable;
- Instant results;
- Relevant results, not only looking at the title of a file, but search in every field possible;
- Clear representation of results;
- File name;
- Contents of the file;
- Creation date;
- Author;
- Company.
- Open files from within the search application;
- Real time Performance;
- Real time Keyword completion;
- Sort function;
- Facetted Search on:
- Persons;
- Company;
- Year;
- File Type.

4.3 Technology

There were multiple contenders for the technology, that was being used to develop the enterprise search tool. We did not want to build an enterprise search solution from scratch, but to build on an already existing solution. A simple search leads to the Magic Quadrant for Enterprise Search[23], see Figure 4. Here we can see, that there are leaders, challengers, niche players and visionaries in the Magic Quadrant. By looking up most of the leaders and challengers it seems that you have to pay to use the enterprise search software. One of the requirements, is that the system must be open-source, because it is free to use and in an open-source community there is much information to be found on the application online. The open-source community can help you if you have any questions.

After trying multiple open-source search solutions, i.e., Elastic Search[25], Lucidworks Fusion[26], Datafari Enterprise Search[27], we went for Lucidworks Fusion. Lucidworks Fusion was the most easiest, out of the box search tool, when comparing it with Elastic Search and Datafari. Lucidworks provides users with lots of documentation, that is needed to implement an enterprise search solution in your organization.

Lucidworks Fusion consists of five main components, that must run to be able to use Fusion:

- Apache Zookeeper;
- Apache Solr;
- Connectors;
- User Interface;
- Lucidworks REST API.

Apache Zookeeper

According to the Apache Zookeeper website Zookeeper is: "a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services."[28] Which makes sure that all of the different components of the search engine can work together.

Apache Solr

Apache Solr[29] is an open source enterprise search server, which is built upon Apache Lucene for indexing and search technologies. Apache Lucene is a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform.[30] Apache Solr has the option to add facets to your search, meaning you can filter on specific database fields, thus enriching your search. Another built in option is query suggestion, which is called typeahead. Users get suggestions, when they are typing. Apache Solr uses near real-time indexing, to present results as fast as possible. The technique used for indexing is based on inverted indices. As a scoring method, Apache Solr uses a combination of the Vector Space Model and Boolean Retrieval for determining relevance[30]. Another importance, is that Apache Solr has an integration with Apache Tika[31]. Apache Tika is a data handling toolkit, that can extract metadata and text from files, not only from text documents, but as well from PDF, via Optical Character Recognition.

Connectors

Lucidworks has more than over fifty connectors, e.g., Network Folders, Google Drive, Sharepoint and Twitter, what we can expect from an Enterprise Search application. By having multiple connectors we can use different data sources as input for the Enterprise Search tool, creating one platform, where all information is present and we do not have to look in multiple places.

User Interface

Lucidworks Fusion has a user interface, which makes it easy to design your search application. It consists of a search, analysis and a developer suite. In the search suite you can design your index and query pipeline, by having access to multiple out of the box components, e.g., signal processing, recommendation booster, faceted search. Using the User Interface saves you hardcoding and makes it easy to user for non-programmers as well.

Lucidworks REST API

The REST API has multiple components. Each component is responsible for sending requests from the Fusion User Interface to the server components, i.e., Apache Solr and Zookeeper.

4.3.1 Architecture

An overview of the architecture of Lucidworks Fusion can be found in Figure 5. The architecture consists of connectors, that are needed to connect information sources to the application. An indexing stage, and query stage, that are both fully customizable. The Fusion API is being used

to connect the user interface to the actual application and to send requests to the Apache Solr server.

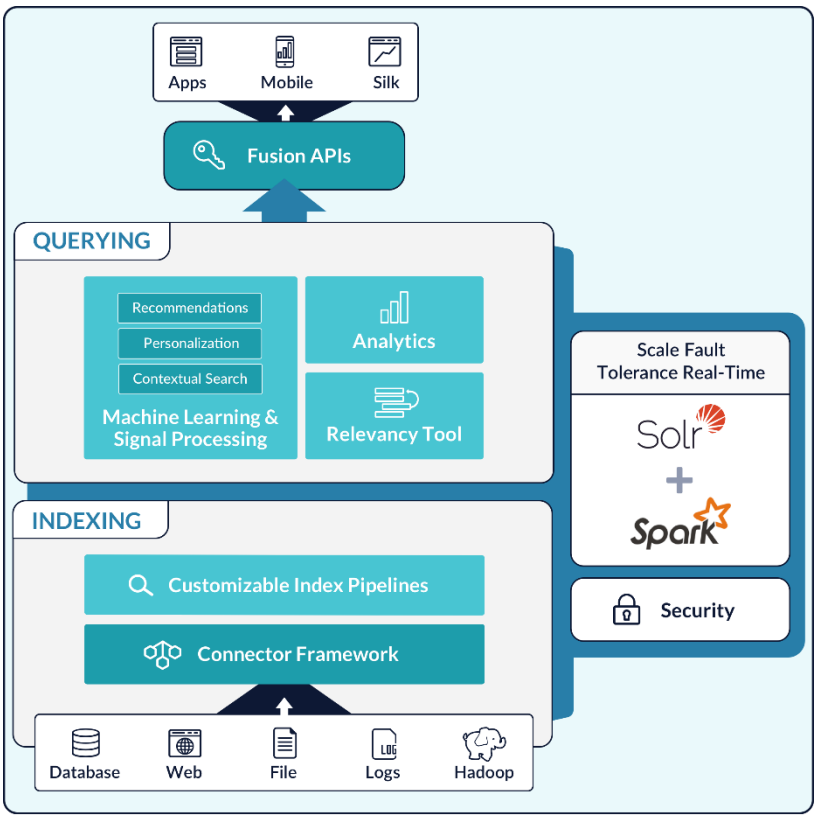

Figure 5: Lucidworks Fusion Architecture[26].

4.4 UL TS - Enterprise Search Application

Here we present how we built the application in Lucidworks Fusion and implemented all the requirements in the enterprise search application. The Enterprise Search Engine has four main components, that need to be implemented separately. Each component is made after each stage is complemented, before moving on to the next one. These components are:

1) Data sources;
2) Indexing stage;
3) Query stage;
4) Front-end.

4.4.1 Data sources

For this research we did not used all the available data, that is present in UL Transaction Security, i.e., data from Sharepoint, E-mail, corporate website, network shared folders, because the main purpose of this research is to evaluate an Enterprise Search Application and it would be too time consuming. Instead, we only focussed on the Network Shared Folders as a data source. From the Network Shared Folders a sample has been chosen, together with a manager from UL Transaction Security to represent a diverse collection of organizational data. The sample set was set up to include projects, training material, payment schemes, white papers, webinars, case studies and organizational templates.

As for file extensions, we choose to only include Microsoft PowerPoint, Word, Excel and PDF, due to the information need of the employees, that was assessed prior to the implementation. We skipped analysing images and did not include files, that were larger than 5 MB.

4.4.2 Indexing stage

In the indexing stage there are lots of options to index the files, that have been chosen in the data source stage.

First we included Apache Tika, to process all the documents and read out their corresponding metadata, e.g., author, file type, body and company. Apache Tika can process all the Microsoft Office documents and PDF files. Second we included a title conversion, that would be more convenient in the representation, instead of the actual network address of the file, we chopped of the first part, that was irrelevant. Third we used a regular expression to extract the year from the date of the document, by using a regular expression, this to later apply the year to a facet in the query stage. The last stage in the index pipeline is the Solr indexer, which sends the documents to the Solr database.

4.4.3 Query stage

For the query stage we needed to design how we wanted to present the indexed results, and what kind of features we wanted to include in our actual search application.

The first stage in the query pipeline stage is adding field facets to the search application. Field facets can help a user get a general idea of the information around a specific query. The facets are dynamic, meaning that they change per query. For the search application, the following field facets have been used:

- Year
- Application Type
- Company
- Author
- Last Author

The facets are in this particular order, because due to intermediate evaluation, this order was found most practical. A query field stage is present, that which let you set up the parameters for the queries. A logging stage is, that sends the queries to a log database, for further analysis. By having a log query, a general idea of the information need can be mapped. The last stage in the query pipeline is the Solr query stage, that sends the query requests to the Solr database.

4.4.4 Front-end

Lucidworks Fusion only serves as the backend of the search application. Lucidworks developed a front end application, that connects seamlessly with Fusion. This application is called Lucidworks View[32]. View comes with an interface and gives a representation of the query stage, that has been developed in Fusion. View is built upon Angular JS[33] and uses SASS: Syntactically Awesome Style Sheets[34]. For this implementation alterations have been made to the source code of View, to make it fit the requirements, that we have presented before.

First the UL Transaction Security style is been applied to the search application, second we made some changes to the interface, to make it more clear for the users. In formatting the results we made the actual result clickable. By clicking the title the corresponding document will be opened directly from the Network Share. Other data has been added to support relevance assessment

for the user on the generated result, i.e., author, contents of the document, creation date, and company. Another feature, that is presented in View is the query completion. By coding the field used for completion, users get a highlighted dropdown menu under the search bar, that presents results when they are typing. The last feature we added was a sort function, that gives users the option to have different sorting of the presented results. This function sorts the document on title or year. After the first feedback session it was clear, that what was the most relevant result for the search application, was not the most relevant result for the users in some cases. Users always want to have the newest version on top, even if by assessing relevance the search application finds something else. The source code of the front-end, of all the alterations, that have been made to the original front-end of Lucidworks View can be viewed in Appendix C.

Screenshots can be found in Figures 6, 7, 8 and 9. In Figure 6 the login screen can be viewed. Different users can be configured, making sure, that not everybody has the same rights, or that only organization's employees can access the enterprise search application. In Figure 7 we present the welcome screen. Here the search query "*" is used, that shows all results present in the Solr database.

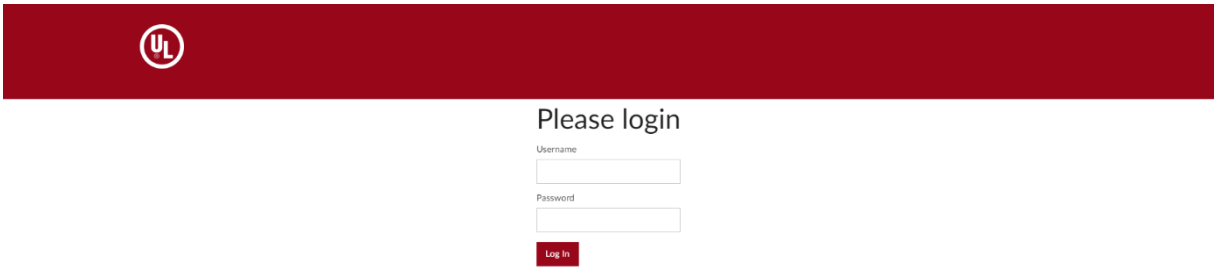

Figure 6: Screenshot Login Screen.

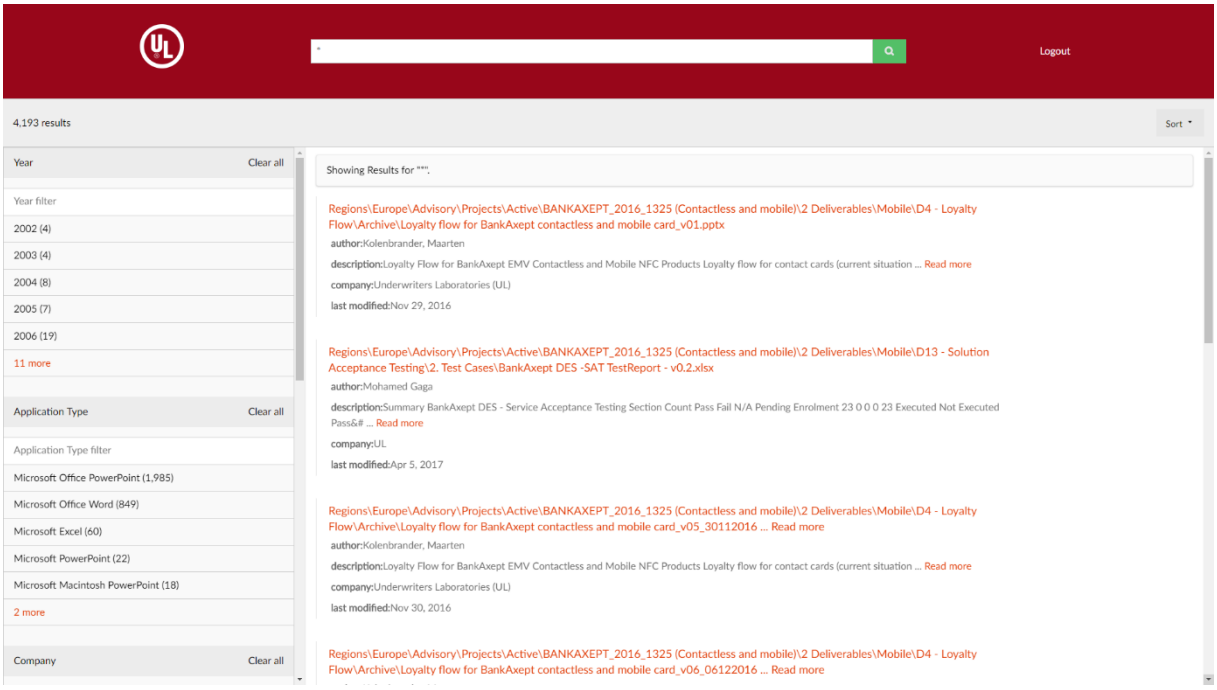

Figure 7: Screenshot Welcome Screen.

In Figure 8 we used the query “white papers”, to give an impression of the query completion feature. If the keyword has a match, it highlights in the query completion dropdown menu.

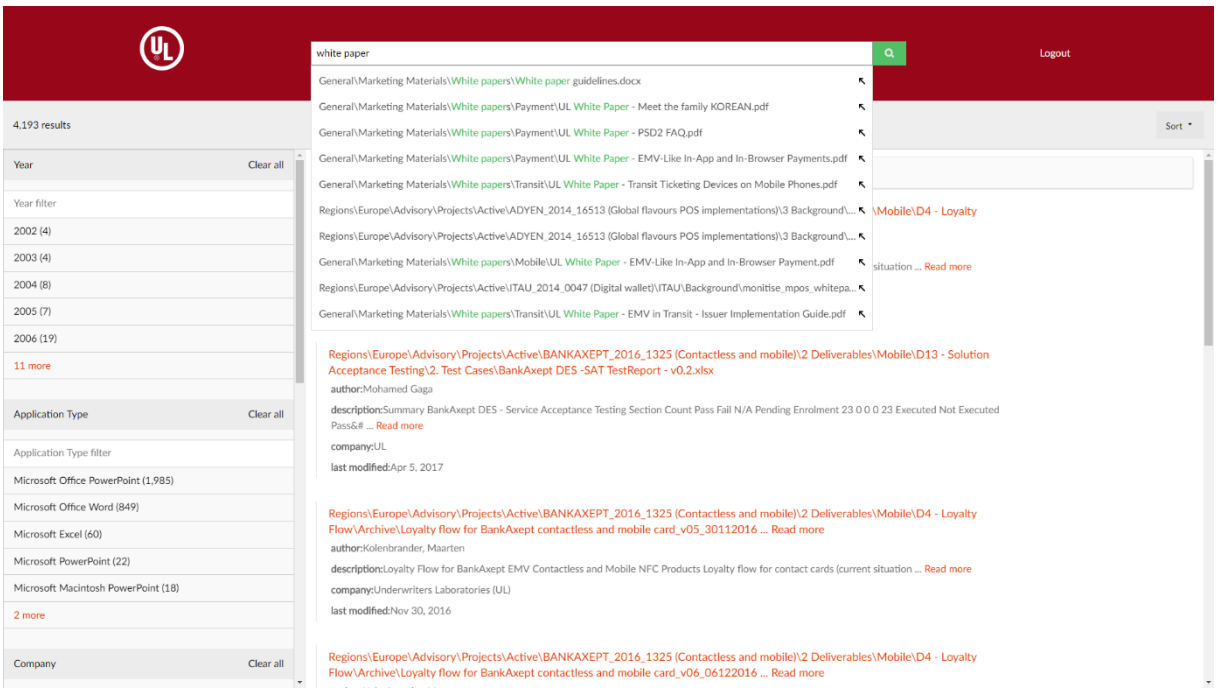

Figure 8: Screenshot Query Completion.

Figure 9 shows the sort function, that can be used to, e.g., list the newest results involved with a certain query on top. At any time the default sort can be selected, that uses the Vector Space Model.

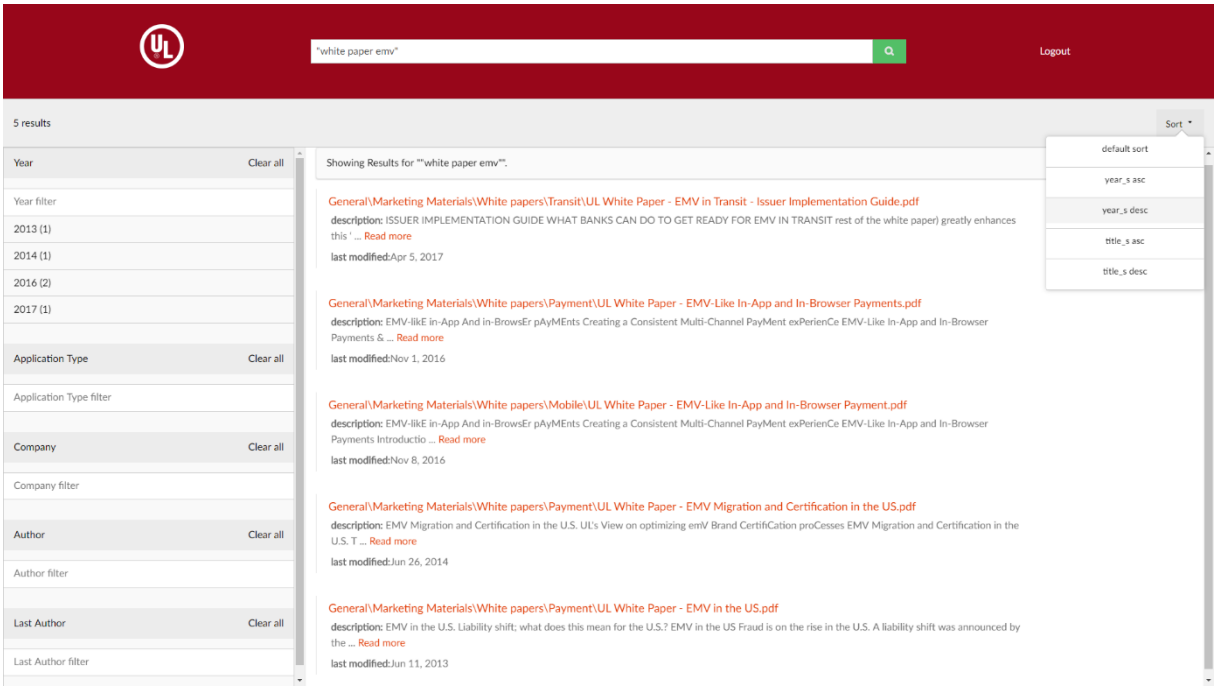

Figure 9: Screenshot Sort Function.

5 Results

In this chapter we present the results of all the experiments, that were being conducted. First there will be the results presented from the search behaviour survey, second the results of the evaluation framework will be presented. In chapter six we analyse the results.

5.1 Survey Search Behaviour

Here the results will be presented for the search behaviour survey. The results are being split per question. The open questions were analysed and grouped by relevance. Comments on each question are being presented. In total forty advisors participated in the search behaviour survey, which is sixty percent of the Leiden office advisory group of UL Transaction Security.

Question 1: What medium(s) do you use for information retrieval?

E-mail	92.50%	37
Yammer	47.50%	19
Sharepoint	20.00%	8
External sources, e.g., Google, Wikipedia	97.50%	39
Network Shared Folders	90.00%	36

Table 1: Search Medium Results.

Comments:

- "Conversations with colleagues is not in the list."
- "The UL Transaction Security website was not included."
- "Scheme sites with specifications maybe need to be included."

Question 2: Rate in terms of importance, how useful the information is, provided by the specific medium.

	1	2	3	4	5	Weighted Average
E-mail	0.00%	2.50%	15.00%	50.00%	32.50%	4.13
Yammer	9.68%	45.16%	25.81%	19.35%	0.00%	2.55
Sharepoint	28.00%	36.00%	16.00%	12.00%	8.00%	2.36
External sources, e.g., Google, Wikipedia	0.00%	17.50%	27.50%	30.00%	25.00%	3.63
Network Shared folders	0.00%	5.00%	12.50%	32.50	50.00%	4.28

Table 2: Usefulness of specific information retrieval mediums.

Question 3: What do you think about the performance of the information retrieval mediums?

	Very Poor	Below Average	Average	Above Average	Excellent	Weighted Average
E-mail	2.50%	20.00%	32.50%	32.50%	12.50%	3.33
Yammer	6.45%	32.26%	35.48%	25.81%	0.00%	2.81
Sharepoint	19.23%	42.31%	23.08%	15.38%	0.00%	2.35
External sources, e.g., Google, Wikipedia	0.00%	5.00%	25.00%	42.50%	27.50%	3.92
Network Shared Folders	12.50%	20.00%	32.50%	20.00%	15.00%	3.05

Table 3: Performance of information retrieval mediums.

Question 4: What is the main intent of information retrieval for each medium?

	To extend your knowledge (not project related)	To extend your knowledge (project related)	To look for specific documents	To look for specific people	I do not use this medium
E-mail	5.13%	25.64%	46.15%	23.08%	0.00%
Yammer	75.00%	5.56%	0.00%	2.78%	16.67%
Sharepoint	14.29%	5.71%	17.14%	2.86%	60.00%
Internet search engine, e.g., Google	43.59%	48.72%	7.69%	0.00%	0.00%
Wikipedia	63.89%	30.56%	2.78%	0.00%	2.78%
Network Shared Folders	0.00%	32.50%	67.50%	0.00%	0.00%

Table 4: Use intent of information retrieval mediums.

Question 5: Are you able to find the information of documents, that are relevant to your work?

Answer Choices	Responses
Always	5.00%
Most of the time	72.50%
Sometimes	22.50%
Never	0.00%

Table 5: Information retrievability.

Question 6: If you cannot find the information you were looking for, what might be the case?

Answer Choices	Responses
I maybe used the wrong keywords	30.00%
I do not know how to search	17.50%
I might be using the wrong information retrieval medium	40.00%
The information may not be present in any system	60.00%
The information retrieval medium isn't working properly	25.00%
Other (give further explanation)	30.00%

Table 6: Causes for a search miss.

Respondents were free to come up with their own reason, why they can't find the information they were looking for. In the responses, there were answers such as:

- Information is badly structured;
- Folder restrictions may apply;
- Didn't know the information existed;
- The information might be archived and not present on the network anymore;
- E-mail search doesn't always work correct.

Question 7: Where do you find the most relevant information for your work?

This was an open question, where the respondents can give answers to their own need. Here we present the answers After deriving key concepts and bundling them together with the amount of people that use the answers are:

Answers	Responses
Network Shared Folders	50.00%
Colleagues	7.50%
Stackoverflow	2.50%
Specs & Documents	25.00%
HIS (for standards)	2.50%
E-Mail	5.00%
At the customers' premises	2.50%
Web search engine	5.00%

Table 7: Location most relevant information for work.

Question 8: What is the main challenge in retrieving this information?

Again, it was not known before, what the main challenge is in retrieving information that is needed for work. If the main challenge is known, there can be solutions presented to help employees with this challenge. Main challenges are:

Answers	Responses
Understand where to find the information	40.00%
Using the right search terms	2.50%
Finding the right version of a file	25.00%
Finding similarity	2.50%
Accessing information from outside/Connectivity	5.00%
Lack of index	5.00%
Lack of a search platform	17.50%
Some information cost money	2.50%

Table 8: Main challenge in retrieving information.

Question 9: Any suggestions on how to improve information retrieval? What kind of tools do you need, and what kind of resources?

By letting the respondents come up with their own suggestion or solution, we can fulfil their search need. What are the current IR mediums lacking? What is needed in a future IR medium? This leads to the following suggestions:

Answers	Responses
Searchable index, information summary;	20.00%
Efficient access to documentation;	7.50%
Overview of projects, that are already present;	10.00%
Document storing conventions;	7.50%
Do not restrict access;	2.50%
Yammer should be used as a Network Shared Folder;	2.50%
Machine learning for automated keyword recognition;	5.00%
Document traceability to document owners;	7.50%
Google like company search engine	20.00%
Everybody needs to put their documents in the Shared Folders	5.00%
Too many different platforms, one cross platform search engine	12.50%

Table 9: Improvements for information retrieval.

5.2 Enterprise Search Evaluation at UL Transaction Security

Here the results will be presented of the experiments, that needed to be conducted for evaluating the enterprise search application. Fifteen employees participated in the experiment, this group is much lower, than the search behaviour survey. With all participants private sessions were mandatory, because the application could only run locally. The participants have different roles in the organization, i.e., a junior consultant, senior consultant, and group leader. In the end a graph will be presented with the final scores of the enterprise search application.

5.2.1 Technical Performance

For the technical performance part we measured response time. The average response time of one hundred searches is 426.78 ms. The grade for the response time is a 5, due to real-time response time.

Freshness for the enterprise search application is real-time, meaning that it indexes files that are new in the data source(s). The grade for the freshness is a 5.

5.2.2 Query Performance

For the query performance participants were asked to assess the relevance of a query. Participants could choose their own query. An overview of the precision and their corresponding query can be found in Table 10.

Query Performance	***Relevant***	***Results***	***Precision***
EMV in Transit	20	20	100
EMV	3	20	15
Bankaxept Des	20	20	100
Host Card Emulation	20	20	100
Mobile Payments	5	20	25
CVM	2	20	10
Wallet Interoperability	20	20	100
Android HCE AND Implementation	20	20	100
Blockchain	1	1	100
Authentication	13	20	65
Mobile Payment Matsterclass	0	10	0
North America EMV	9	20	45
Transit Test Approach	10	20	50
HCE Workshop	20	20	100
EMV Next Gen	9	20	45
TFM & eReceipt	20	20	100
Android Pay	6	20	30
Risk Assessment	3	20	15
PSD2	20	20	100
Mifare Training	18	20	90
HCE Workshop	10	20	50
Card Skimming	15	20	75
Visa cloud payments pdf	5	10	50
MDES	19	20	95
What is PSD2	11	20	55

Globalplatform Access Control	20	20	100
Implementation Guide	16	20	80
VAS Consumer ID	20	20	100
QR	3	20	15
MCBP	16	20	80
Samsung Pay	16	20	80
Apple Pay	15	20	75
Bankaxept Architecture	8	10	80
Digital Enablement Services	18	20	90
What is Android Pay	12	20	60
TSP Requirements	11	20	55
Secure Element	20	20	100
MCBP Security	7	20	35
Digital Onboarding	0	20	0
Bluetooth BLE Whitepaper	2	10	20
What is Androis Pay	12	20	60
Token Service Provider	20	20	100
CTAP Acquirer Discretionary Data	18	20	90
HCE	3	20	15
Testing and Certification	20	20	100
Smart Tachograph Specification	4	20	20
Mifare DESfire	9	20	45
Contactless Acceptance Germany	1	10	10

Table 10: Precision of certain keywords in the Enterprise Search Engine at UL Transaction Security.

With forty-eight different queries the average precision is 63.02. This gives a grade of 3.2 for the precision of the enterprise search engine.

Search Log

In the enterprise search engine a back-end is present, that lets you explore different search logs, this includes response time and the query used for the search. It is not possible to easily search through the search log, therefore a grade of 3 will be given. Here we give a grade of 3, because a searchable search log would be desirable, but is not present. Click signals could be used to increase the ranking of a query, however click signals are not being used, due to design choices.

Query Completion

Here the question was asked: "What do you think of the performance of the query completion?" Here we take the weighted average grade 3.6, that can be seen in Table 11.

Very bad	Bad	Neutral	Good	Very good	Weighted Average
0.00%	13.33%	20.00%	60.00%	6.67%	3.60

Table 11: Query completion rating at UL Transaction Security.

5.2.3 Usability & Accessibility

The results for the usability SUS test can be found in Table 12. Remember that the final score needs to be calculated according to the SUS method[20]. After conversion of the averages a total average of 29.66 is obtained. The total SUS score of 74.15 is obtained. Converting this to a grade for final evaluation a grade of 3.7 is obtained.

	Strongly disagree	Disagree	Undecided	Agree	Strongly Agree	Weighted Average
I think that I would like to use this system frequently	0.00%	0.00%	20.00%	73.33%	6.67%	3.87
I found the system unnecessarily complex	26.67%	60.00%	13.33%	0.00%	0.00%	1.87
I thought the system was easy to use	0.00%	0.00%	6.67%	73.33%	20.00%	4.13
I think that I would need the support of a technical person to be able to use this system	53.33%	33.33%	0.00%	13.33%	0.00%	1.73
I found that the various functions in this system were well integrated	0.00%	13.33%	33.33%	40.00%	13.33%	3.53
I thought there was too much inconsistency in this system	6.67%	53.33%	33.33%	6.67%	0.00%	2.40
I would imagine that most people would learn to use this system quickly	0.00%	0.00%	0.00%	60.00%	40.00%	4.40
I found the system very cumbersome to use	6.67%	73.33%	13.33%	6.67%	0.00%	2.20
I felt very confident using this system	0.00%	6.67%	33.33%	60.00%	0.00%	3.53
I needed to learn a lot of things before I could get going with the system	40.00%	60.00%	0.00%	0.00%	0.00%	1.60

Table 12: Usability Test results at UL Transaction Security.

5.2.4 User Satisfaction

The results of the user satisfaction test can be found in Table 13. The total average grade for the user satisfaction is 3.5

	Very dissatisfied	Dissatisfied	Neither	Satisfied	Very satisfied	Weighted Average
How satisfied are you with the user interface?	0.00%	13.33%	6.67%	66.67%	13.33%	3.80
How satisfied are you with the relevance of the results?	0.00%	20.00%	26.67%	46.67%	6.67%	3.40
How satisfied are you with the speed of the search engine?	0.00%	0.00%	20.00%	53.33%	26.67%	4.07
How satisfied are you with the filter function?	0.00%	26.67%	40.00%	20.00%	13.33%	3.20
How satisfied are you with the sorting function?	0.00%	13.33%	46.67%	40.00%	0.00%	3.27
How satisfied are you with keyword completion?	13.33%	46.67%	26.67%	13.33%	0.00%	2.40
How satisfied are you with the ease of use?	0.00%	0.00%	6.67%	80.00%	13.33%	4.07
How satisfied are you with how the results are being presented?	6.67%	26.67%	20.00%	46.67%	0.00%	3.07
How satisfied are you, when comparing this tool to the previous search application?	0.00%	0.00%	26.67%	40.00%	33.33%	4.07
How satisfied are you with the tool in general?	0.00%	6.67%	20.00%	66.67%	6.67%	3.73

Table 13: User satisfaction test results at UL Transaction Security.

5.2.5 Further Remarks

In the survey presented to the participants there was an option to give further remarks and feedback. In Table 14 an overview of these remarks are given, where similar answers were being bundled. Thirteen people provided multiple remarks, since it wasn't mandatory to answer this question.

Answers	Responses
When searching for a document, you get older versions of the same document.	23.08%
Very good for getting information on a topic.	30.77%
Less good for finding a specific document.	15.38%
There is no reset button to reset the filters.	7.69%
Option to preview the file, instead of directly opening it.	15.38%
Highlighting of keywords in the presented results would meaningful.	30.69%
When expending the description a huge block of text is appearing, this is not very user friendly.	15.38%
Results are lost, when the browser is closed by accident.	7.69%
Show the file name in query completion, instead of the whole document path.	30.77%
Sorting on title is not necessary.	7.69%
Filter function should only be available when selecting it.	7.69%
It would be nice, if the page number can be seen.	15.38%
Search only through specific fields in the database, i.e., title.	7.69%
Representation can be better, when the document path and title are being split.	15.38%

Table 14: Feedback on the Enterprise Search Application at UL Transaction Security.

Figure 10 shows the final grades being given to the enterprise search engine after the evaluation.

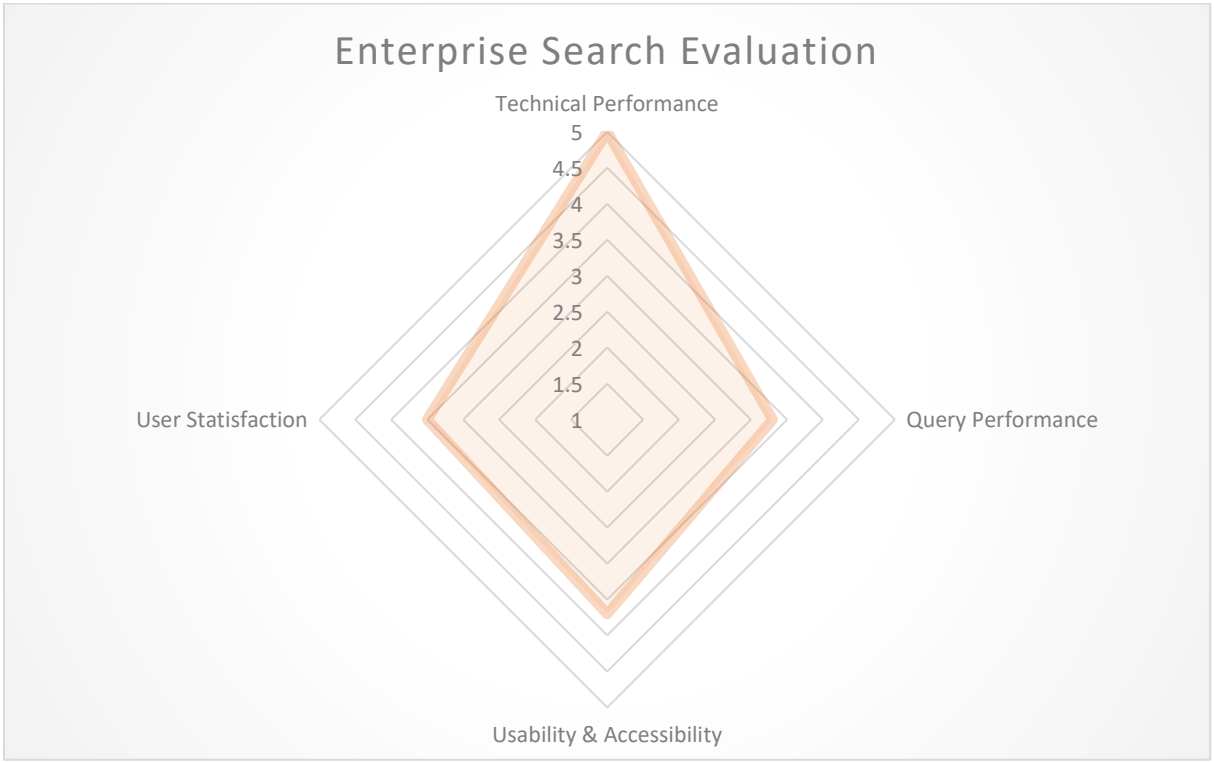

Figure 10: Enterprise Search Evaluation at UL Transaction Security Grades

5.3 Enterprise Search Evaluation at a Dutch research institute

Here we present the results of the experiments that were conducted to evaluate an enterprise search application at a Dutch research institute. Eleven employees participated in the experiments. A survey based on the survey in Appendix B is being used for these experiments. To get a general impression of the search behaviour, two introduction questions were asked.

Question 1: In Which situations do you use the search function? (e.g., serach for documents, search for experts)

Answers	Responses
Searching for colleagues.	20.00%
Searching for documents.	60.00%
Looking for a specific page.	10.00%
Never	10.00%

Table 15: Search behaviour of Dutch research institute.

Question 2: How often do you use the search application?

Answers	Responses
Daily	30.00%
Multiple times per week	10.00%
Once a week	20.00%
Multiple times a month	20.00%
Once a month	0.00%
Less than once a month	10.00%
Never	10.00%

Table 16: Search frequency of Dutch research institute.

5.3.1 Technical Performance

Due to the lack of direct access to the backend of the search application we could not measure the technical performance, i.e., response time and freshness.

5.3.2 Query Performance

Here we used the same question to measure the precision of the search system as in section 5.2.2. In total forty-three different queries were being used. Due to confidentiality we cannot display the different search queries used by the employees. Different types of queries were being used. Employees searched for people, general IT terms, and projects they were working on. A difference in precision was found, where there was a range of zero to hundred in the precision and their corresponding queries. The average precision for these queries is 38.8. This results in a grade of 1.94 for the precision of the search application.

5.3.3 Usability & Accessibility

In Table 17 we can find the results for the SUS usability test. The total SUS score is 57.5. This converts to a grade of 2.88 is obtained.

	Strongly disagree	Disagree	Undecided	Agree	Strongly Agree	Weighted Average
I think that I would like to use this system frequently	**9.90%**	**36.36%**	**9.09%**	**27.27%**	**18.18%**	3.09
I found the system unnecessarily complex	**9.09%**	**45.45%**	**18.18%**	**9.09%**	**18.18%**	2.82
I thought the system was easy to use	**0.00%**	**36.36%**	**18.18%**	**36.36%**	**9.09%**	3.18
I think that I would need the support of a technical person to be able to use this system	**54.55%**	**45.45%**	**0.00%**	**0.00%**	**0.00%**	1.45
I found that the various functions in this system were well integrated	**9.09%**	**27.27%**	**45.45%**	**18.18%**	**0.00%**	2.73
I thought there was too much inconsistency in this system	**9.09%**	**18.18%**	**36.36%**	**27.27%**	**9.09%**	3.09
I would imagine that most people would learn to use this system quickly	**9.09%**	**9.09%**	**36.36%**	**45.45%**	**0.00%**	3.18
I found the system very cumbersome to use	**27.27%**	**18.18%**	**9.09%**	**27.27%**	**18.18%**	2.91
I felt very confident using this system	**9.09%**	**36.36%**	**18.18%**	**36.36%**	**0.00%**	2.82
I needed to learn a lot of things before I could get going with the system	**54.55%**	**18.18%**	**27.27%**	**0.00%**	**0.00%**	1.73

Table 17: Usability Test results at Dutch research institute.

5.3.4 User Satisfaction

User satisfaction test results can be found in Table 18. Here some questions were left out, because the search application did not have that specific functionality or there was no previous search tool. The average user satisfaction grade is 3.15.

	Very dissatisfied	Dissatisfied	Neither	Satisfied	Very satisfied	Weighted Average
How satisfied are you with the user interface?	9.09%	27.27%	18.18%	45.45%	0.0%	3.00
How satisfied are you with the relevance of the results?	0.00%	45.45%	36.36%	18.18%	0.00%	2.73
How satisfied are you with the speed of the search engine?	0.00%	0.00%	0.00%	90.91%	9.09%	4.09
How satisfied are you with the filter function?	0.00%	9.09%	63.64%	27.27%	0.00%	3.18
How satisfied are you with the sorting function?	0.00%	18.18%	63.64%	9.09%	9.09%	3.09
How satisfied are you with the ease of use?	0.00%	27.27%	9.09%	63.64%	0.00%	3.36
How satisfied are you with how the results are being presented?	18.18%	18.18%	9.09%	54.55%	0.00%	3.00
How satisfied are you with the tool in general?	18.18%	27.27%	18.18%	36.36%	0.00%	2.73

Table 18: User satisfaction test results at UL TS.

5.3.5 Final Score

Here the final score is presented in Figure 11. Only three aspects of the evaluation framework are used, i.e., query performance, usability and user satisfaction.

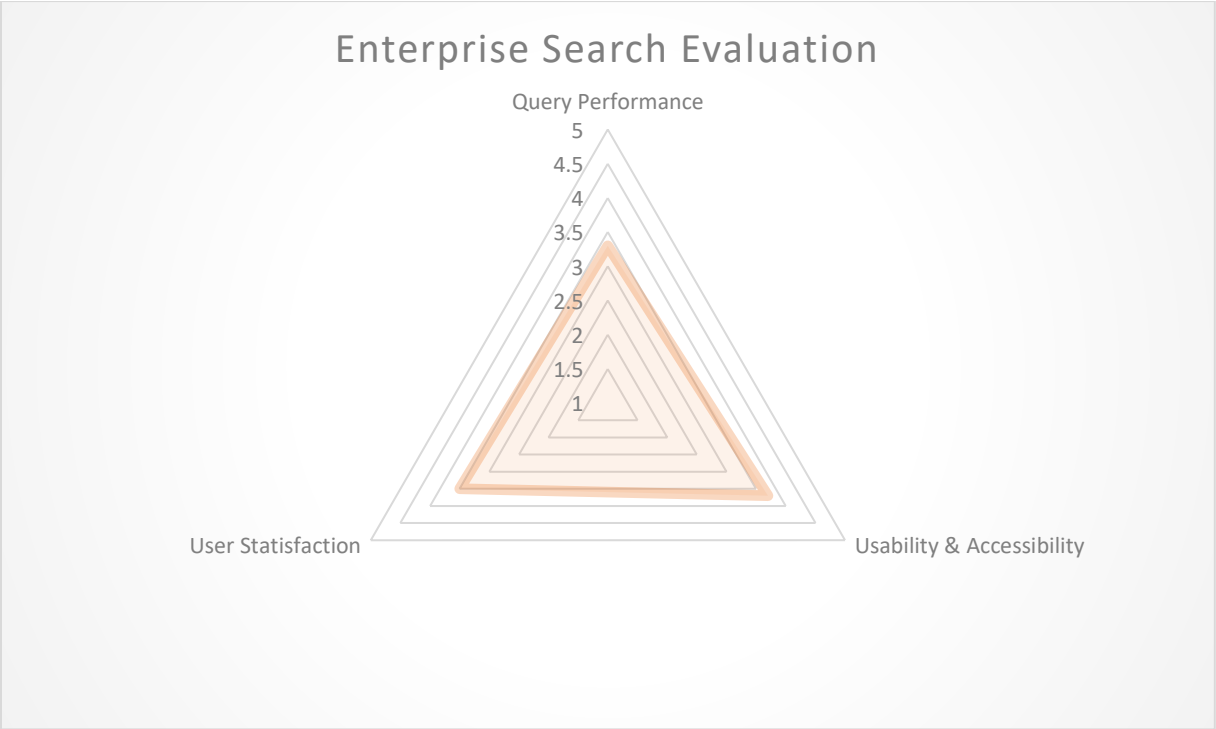

Figure 11: Enterprise Search Evaluation at Dutch research institute grades.

6 Discussion

In this chapter we discuss the results from chapter five. First we discuss the results and limitations of the search behaviour survey. Second we discuss the results and limitations of the enterprise search evaluation framework.

6.1 Search Behaviour Survey

The search behaviour survey has been conducted before any implementation was done. This was done to create an understanding of how employees are interacting with information retrieval systems and as a means to define requirements for the enterprise search application. As can be seen in Appendix A, we had already set up predefined answers, which can bias employees in only selecting the options presented. The predefined answers were defined by talking to multiple employees with different roles in UL Transaction Security.

If we look at the results of Table 1, we can derive that external sources are more popular than internal sources. There is a specific information need for external sources. It is strange to see, that only 20.00% are using Sharepoint, especially when keeping in mind, that all the files from the Network Shared Folders will be migrated to Sharepoint. E-mail is being used more than files on the Network Shared Folders, when searching for information.

By looking at Table 1, we can see that again Sharepoint is the least popular tool. The information, that is found in the Network Shared Folders is observed as most useful. E-mail is not far behind in terms of the weighted average. Of the 40 respondents, only 2 find Sharepoint the source for the most useful information. Unfortunately, we can't contact these respondents, because this was an anonymous survey.

The performance of the information retrieval medium is very important. With this question, there can be measured what the desired performance is of a search tool. What stand outs is, that if we look to the weighted average, none of the selected information retrieval mediums have a performance of excellent. Of course, performance is in this case very subjective, because we asked users how they perceive the performance of a specific IR medium.

People perceive Sharepoint and Yammer as less performing than Network Shared Folders, E-mail and External sources. Sharepoint and Yammer have a weighted average of below average, where Network Shared Folders and E-mail have a perceived performance of average and External sources are perceived as above average.

Everybody uses e-mail as an IR source. The main intent is divided between extend project related knowledge and to look for specific people. Most employees use e-mail to look for specific documents. Most employees 75.00% use yammer to extend their knowledge, for a not project related cause. Only 5.56% use it to extend their project related knowledge. Nobody uses Yammer to look for specific documents. 16.67% of the respondents do not use this medium. Sharepoint is the medium that has the highest percentage of not being used, this is 60.00%. Further it is very divided how people use Sharepoint. Employees use it to extend their knowledge, not project related 14.29% and to look for specific documents 17.14%. The main intent of an internet search engine is to extend the knowledge of the employee, not project related 43.49% and project related 48.72%. Some employees use it to look for specific documents, 7.69%. Wikipedia is being

to extend the knowledge, not project related, 63.89% and 30.56% for project related purposes. The Network Shared Folders consist of documents, therefor it isn't strange, that the main intent is to look for specific documents, 67.50%. All the other respondents answered, that the main intent was to extend their project related knowledge, 32.50%.

Looking at Table 5, we see, that nobody answered that they could never find information or documents, that are relevant to their work. So somehow the information, that is needed for their work is found.

There are respondents, who answered sometimes, 22.50%. Meaning that there are cases in which they do not find the information or documents, that are needed for their work. Sometimes being that most of the time, they can't retrieve the information.

Most respondents can find the information or documents most of the time, 72.50%. Meaning they have no problem when retrieving the information, but there might be some cases where they are unable to find the information or documents.

There are respondents, who are information finding masters, 5.00%, they can always find the information or documents, that are relevant for their work. These respondents can help the other respondents, who have trouble finding information. Is not finding information or documents a user based phenomena? Or does it have to do with the IR system?

For the question In Table 6 respondents could pick multiple answers, because when information can't be retrieved, there needs to be a certain cause. If the cause is known, that can be solved by implementing this feature in a future search engine.

The main reason, why respondents can't find information, is because they think that the information may not by present in any system, 60.00%. This can have two causes, one cause can be, that the information is indeed not present in any system. The other cause, is that the information may not be indexed properly and isn't shown by any search medium.

The second highest reason, is that respondents might be using the wrong IR medium, 40.00%. Information may be present on another medium, but the users are not aware of this.

Respondents also thought they might be using the wrong keywords for search, 30.00%. They can't find the information they are looking for with the keywords they are using. They have the feeling, that other keywords may help, but they don't know which keywords will help them. A recommender system can help, this can give users recommendations on what keywords other users are searching for, where they can get a keyword that have a higher relevance return.

Folder restrictions can be an issue, especially when working in a knowledge driven company, where everybody should have access to the same files. This isn't always a case, due to confidentiality.

Most respondents answered they get their information from the Network Shared Folder, if it is related to their work.

Using the correct search terms is indeed necessary to generate relevant results, but it can be made easier for the users, by using a recommender engine and keyword correction. This way users are being helped, when they make mistakes in their keywords.

Connecting remotely is also very important, because most of the employees are working on a project with an external customer, meaning they can't reach the network share if the network of the customer doesn't allow using a VPN.

Version control is very important; employees always need the newest version of a document. Sometimes, the newest version isn't in the Network Shared Folders, because the employee didn't upload it yet. Employees need to be actively engaged in sharing the newest version of their files and documents, because this will benefit colleagues.

Finding the correct folder is necessary, if an employee is looking for a specific project, that is related to the project the employee is currently working on. By gaining insight in how a previous project went, it can save time and effort for the current project.

Archiving is nice, but do not underestimate how important some files may be in the future. Documents, that may seem irrelevant can be a hot topic in the future. Therefore, it may be better to not archive files and let people search through all the content and not an incomplete set.

Access rights are sometimes necessary, when working with confidentiality. Working with access rights can also be frustrating, especially when you need some protected documents from a colleague for your current project.

The Windows Search engine doesn't use filters when presenting information, this makes it difficult when performing a navigational search task, or when there are lots of results presented.

Respondents stated, that they need to know where to look for information. This is indeed a problem, when a company uses multiple sources of information. It takes a lot of time, when an employee needs to search on all the different platforms. A solution can be cross-platform search. Cross-platform search lets you search through multiple platforms, such as Windows Search and Web Search by using one search engine.

If we look at the case of the Dutch research institute in Table 15 we see, that most of the employees, 60.00% use the search application to search for documents, where 20.00% use it to search for people and 10.00% to search for a specific page. Of course this does not represent the entire organisation, due to the input of eleven employees, but the majority uses it to mainly search for documents.

Looking at the usage of the search application in Table 16 we can see that most users use the application on a daily basis, 30.00%. and that 10.00% uses it less than a month. Now the real question is, why are there users who never use the application, 10.00%. Is it because this user is never in need for using search to support his/her work, or is it because this user uses external search applications. Unfortunately we could not figure this out, due to the anonymity of the survey.

6.2 Enterprise Search Evaluation

Here we discuss the results of the Enterprise Search Evaluation Framework, and their corresponding experiments at UL Transaction Security and a Dutch research institute together with limitations of this framework and how the experiments were conducted.

6.2.1 Technical Performance

First by looking at the technical performance, the average response time is 426.78 ms. The response time was only measured on company computers of UL Transaction Security. The response time may differ, if other hardware is being used. The participants of the experiments did not encounter any delays in response time. For participants it looked like the search engine provided real-time results, due to the response time of nearly half a second. The response time was being calculated, by looking at the response time during the experiments, because we wanted to get the response time in a natural search environment.

For the freshness of the system, it indexes files, directly when they are present in the predefined document source, that is being linked to Lucidworks Fusion. If a document is being placed in a source, Fusion automatically detects the new file, and indexes this file by using the indexing stage pipeline. We encountered no delay, when indexing new files, but indexing may take longer, if lots of new files are being put in a document source, or if we want to include images in the indexing stage.

If we look the Dutch research institute we could not include the technical performance, due to the lack of access to the backend. It is preferable to have direct access to the backend of the search application to fully assess the technical performance of a search application.

6.2.2 Query Performance

The average precision of the enterprise search engine, that can be derived from Table 10, is 63.02. This is above average and would mean that in general a relevant result can be found in within the first ten results. Of course this is only being calculated by looking at forty-eight different queries, that is enough to assess the precision of the system. The plus side, is that the keywords that are being used, are keywords, that will be used by employees of UL Transaction Security, because they represent a realistic situation and adapted to the information that is present in the enterprise search system. What can be seen, as that there is an overlap in keywords. In total if all participants would have used different keywords, we would have had 75 keywords (15 * 5). Another remark about precision is that, when assessing relevance, it can change per user. Relevance is subjective and user based, there can be differences in what one user finds relevant. In this method, we only picked the first twenty results and assessed their relevance. We used the average precision, an advantage of this method, is that it saves time, and relevance on more keywords can be assessed in a smaller timeframe. A disadvantage, is that we only have assessed a fraction of the results. This method was easier for the participants, because otherwise they needed to assess around 4000 results for the query “EMV”.

When sorting Table 10 by grade, we can see what keywords have a precision of hundred percent and others of ten percent. So why do these keywords have a precision of hundred and do other keywords have a precision of zero, ten or fifteen? One explanation can be, that employees were searching for keywords, that were not included in the data set, after all we only indexed a sample of all the company data. Although the participants were known of this fact, because the data set was shown. It may be the case, that not every employee knows, what is in our dataset. What

stands out, is that when searching for only one high level keyword, the information need of the user is often not fulfilled, because the search scope is too broad. When participants used more specific keywords, or questions, the information need seemed to be better fulfilled. Keyword analysis can also be used to map all the important topics for an organization.

Looking at the query completion performance in Table 11 we can see, that participants mostly found it good, 60%, but there was some feedback on the query completion. The query completion shows the whole document path, and most of the time this path does not fit in the query completion bar, making it impossible to see the actual file name. As for the relevance of the query completion, the participants found the suggested results a good indication of where to start the search. We can even discuss if a query completion is a requirement for evaluating an enterprise search engine, but most search users are spoiled, because they expect Google-like features in a search application, therefore it is required.

Regarding the case of the Dutch research institute forty-three different keywords are being used, which is only four less than the case at UL Transaction Security. Here we see that the average precision is lower, 38.8. There can be multiple causes, first the search application being used at the Dutch Research institute was different from the application implemented in UL Transaction Security. Second relevance is subjective, meaning that it could be the case that the employees of the Dutch research institute were less satisfied with the results and judged the results as less relevant. Third a different data set is being used, and the queries that the employees choose differ a lot, due to the difference in domain knowledge.

No questions were being asked about query completion, because this was not present in the search application of the Dutch research institute. In the original evaluation framework query completion is present, because users expect this functionality to be included in a search application. This is because most web search applications have a query completion function.

6.2.3 Usability and Accessibility

For the usability and accessibility we used the SUS test, as can be seen in Table 12. Here the usability score is 74.15, meaning that participants found the system highly useable. Let us zoom in more on the separate questions of the usability test. Nobody disagreed, that they would like to use the system frequently, only 20% was undecided, and 80% agreed. As for the complexity of the system, none of the participants found the system complex to use, that can be due to the simple design requirement and 93.33% agreed, that the system was easy to use. In the case of support from another person, the answers are a bit divided, most participants 86,67% thought they did not need the help from a technical person, but 13.33% thinks otherwise. So even if there is a small fraction, who thinks that they need the help of a technical person, or maybe some extra information around the system, then this needs to be provided. This can be in a small manual, or a short training.

About integration of the different functions, the answers were quite distributed. This can be due to personal differences, or due to differences in expectations of a search application. Only a small part found that there was too much inconsistency in this system. These participants wrote this done in the feedback, and came up with solutions to make it better. All agreed, that most people would learn to use the system quickly, with 40% that strongly agreed. Most people, 80% did not found the system cumbersome to use, 13.33% was undecided and 6.67% found it cumbersome to use. Participants who found it cumbersome can be contacted, therefore decreasing negative responses for usability in the future. The same holds for, confidence in using the system. Most

participants felt confident in using the system, 60.00% but there are still participants who thought otherwise, 6.67%. Strangest thing, is that 33.33% was undecided in confidence of use. Nobody needed to learn a lot, before they could get going with the system. Everybody disagreed, which makes it clear, that one of the requirements succeeded, the ability to directly use the system.

Looking at the results in Table 17 and at the SUS score for the Dutch research institute we see that the usability is below average. The norm for below and above average is 68[35]. Why is it below average? Looking at Table 17 we see that 36.36% disagreed to use the system more frequent, but this is not because the system is complex. Here 45.45% disagreed that the system was complex to use. Divided responses can be found, where 36.36% disagreed and 36.36% agreed that the system was easy to use. Although people disagreed that the system was easy to use, they did not need the help of an expert, here 45.45% disagreed and 54.55% strongly disagreed. Again there are a lot of divided answers when looking at other questions. 27.27% found that the system has too much inconsistency, where 18.18% disagreed. 36.36% disagreed to be confident in using the system, where 36.36% agreed on feeling confident.

Of course we only performed the SUS usability test with eleven people, meaning that filling it in with twenty more people could give a less divided result. It could be that some employees were negatively biased in using the search system, due to bad experiences in the past.

The SUS test takes only a few minutes to fill in, and can be used to improve certain elements of the search application and engage with employees who disagree on certain usability features, e.g., if an employee finds the system cumbersome to use, reasons can be given why, these reasons can be taken into account for the next implementation cycle, so that in the end every employee finds the application usable.

There are of course other methods for assessing the usability of a software system, but this method is, as being said by John Brooke, quick and dirty[20]. We did not want to bother participants with very complicated and lengthy surveys, thus the SUS test was chosen.

6.2.4 User Satisfaction

The results for the user satisfaction survey can be found in Table 13. Here ten questions were asked to measure the user satisfaction of the search application. Here the user satisfaction is split into the different components of a search application. During the experiment we did not explain or show much of the application, to create an environment, where the participant can be honest and critical. Still we could ask other or more questions, but the aim was to keep it as easy as possible, to not waste any resources given by the organization.

Regarding the user interface, 13.33% was dissatisfied and 80% was satisfied. The user interface was simple, but had a very large design. Participants wanted to see all the results on the first page, without having to scroll down. The reasons why a big design was chosen, was that it would be easy to use on small screens, e.g., smartphones. It is strange, that 26.67% found the results neither satisfying or dissatisfying. It could be, that participants, did not had an opinion on it, or that they would want to use the application for a longer period. Thirty minutes was given to play around with the system and to fill in the survey from Appendix B. 55% was satisfied, and 20% was dissatisfied. This can be due to the personal experience with the system, using queries, that gave relevant results, compared to queries, that gave less relevant results. If for measuring the precision of the system, standard keywords had been used, instead of the user, that could make

up their own query, the results could have been different. Letting users come up with their own queries gives a clearer representation of the whole system, and standard queries can be a biased.

Nobody was dissatisfied with the speed of the system. This can be linked to the response time, the average response time is nearly half a second. The response time explains why 80.00% was satisfied, and 13.33% was extremely satisfied. For the filter function on the left sidebar, that can be seen in the screenshot in Figure 7, participants had mixed feelings. Nobody was extremely satisfied, or dissatisfied, but in the remarks can be shown, that participants found, that the filters were not always very useful. This was because there was lots of garbage data, i.e., doubles, odd names, or the filters were just irrelevant. The reason for the garbage data is the old saying: "Garbage in Garbage out". We only indexed the file, and did not change any file naming conventions. A standard can be made by UL Transaction Security to increase the data quality, which results in less garbage data in the search application.

More people, 40.00%, were satisfied with the sorting function. This can be, because participants wanted to sort on year, instead of the raking of the search application. Ranking is based on relevance, but when working on a project you want the newest file to work with. As already being discussed during the performance of the query completion, participants were rather dissatisfied, because they could not see the file name, when using the query completion. This is a design issue, and could be easily resolved.

Nobody was negative towards the ease of use of the system, 93.33% was satisfied with the ease of use and 6.67% was neither, which is not strange, because a similar question was asked during the usability test, and had the same results, were everybody agreed on the ease of use. In the representation of the results, there were mixed feelings. There is lots of room for improvement. What can be found in the remarks, is that using the description of the document, i.e., the body of the document, is not very user friendly. As well as the title of the document, the full document path can be handy, but it takes up a big amount of space, that aside, there are still more people satisfied, than dissatisfied.

We asked participants how satisfied they were, when comparing the search application to a previous search application. Here many users found, that there was not a previous search application. Here we explained, that can be thought of regular browsing to the network shared folders, or using Windows Search. Here 73.33% of the participants were satisfied, 26.67% was neither, but nobody was dissatisfied. Meaning that nobody liked the old solution better. In general the majority, 73.33% was satisfied with the search application. 6.67% was dissatisfied and 20.00% was neither. Here the key to improve the search application is to interview the dissatisfied participants.

The similar user satisfaction survey was conducted at the Dutch research institute. The main difference was that we left two questions out. The question about the keyword completion, because the application did not have such a function and the question about a precious search application, because there was no precious search application.

In Table 18 we see that people were leaning more towards a satisfied attitude about the user interface, 45.45%, than a dissatisfied attitude, 27.27%. What stands out is that 45.45% of the people are dissatisfied with the results. This can be linked to the query performance, where the average precision was 38.8. Only 18.18% was satisfied with the relevance of the results. It could be that those people used queries, that gave a high relevance.

Everyone is satisfied with the speed of the search application, where 90.91% was satisfied and 9.09% was extremely satisfied. For the filter function and the sorting function most people where not satisfied or dissatisfied. For both the sorting and filter function this is 63.64%. It could be that people did not have an opinion about those functions. The majority were satisfied with the ease of use of the system, 63.64, but still there was 27.27 dissatisfied with the ease of use. In total there are more people who have a negative attitude towards the search application in general, which is 45.45% against 36.36% who have a positive attitude. Even if some employees seen somewhat dissatisfied the overall grade for the user satisfaction is 3.15, which expresses a more positive grade. Some employees have a stronger dissatisfaction than others, but the average seems leaning more to a positive attitude.

6.2.5 Conclusion

In Figure 10 we have seen the final grades, that has been given to the enterprise search application of UL Transaction Security. This image gives a clear view on where the search application can improve. Technical performance is at its peak. We believe, that as a search engine, the technical performance has a top performance. Query performance can be improved, if the whole dataset of the organization would be used, precision can increase, because there is more data in the application. The query completion performance can be improved, by using simple measuring to make it more clear for the user. The system is usable, but there still can be improvements, that can be found in the further remarks section (5.2.5). Not everybody is extremely satisfied, but this can improved by fulfilling the needs of the user, and be lowering expectations. The main problem with expectations, is that an enterprise search engine, is no Google, but some employees of an organization expect, that it is very easy to make such a search engine.

The final grades for the Dutch research institute can be found in Figure 11. Here the technical performance part was left out, due to lack of access to the backend. The highest grade is being given to user satisfaction, a 3.15, where the lowest grade is given to query performance, 1.94. It is not strange, that grade of the query performance affects the user satisfaction and the usability of the search application. The main findings were, that there was enough room for improvement and that employees seem somewhat divided on their opinion about the search application. Questions where employees gave a negative response can be used as an input to improve the search application.

Applying the evaluation framework to the Dutch research institute give us insight in the strong and weak points of the evaluation framework. First direct access with the organisation is a must, because you want to have access, or at least let somebody send you the search logs, to assess the technical performance. Second, a standard list of questions for the user satisfaction is not possible, because every search tool has different functionalities. There are some questions that can always be used, because some functionalities are always present in a search application. Third, by being in the actual organisation employees have the tendency to feel more involved. It is much easier to contact people directly, when present in the organisation.

By applying the evaluation framework in an application in a Dutch research institute we have seen, that employees are not eager to participate in an online survey. The main difference with UL Transaction Security was that during the experiments we were constantly present. Appointments were made with the employees at UL Transaction Security and because we were present in the organisation, it was easier to get employees to participate, because we knew them.

An alternative for surveys could be getting answers for all the questions in structured interviews. Where the interviewer fills in the questions himself. By planned interviews employees feel more obligated to participate. Using interviews takes more time, but can help to get more participants.

As we have already said before, measuring the business impact is not part of our scope, but is extremely important to see, what the actual impact of the search application has on the business, e.g., if it generates value for the organization.

In this analysis we answered two research questions. First:

Does the enterprise search application work accordingly?

By looking at the usability test, we can conclude, that there are still some flaws, but it works, people can search in a way, that is more of this age, than browsing through large network folders and be overwhelmed of all the documents that are present.

Another research question that is being answered is:

What is the opinion of employees on an enterprise search application?

The user satisfaction survey, and the reaction in the further remarks section give answers to these questions. The user satisfaction grade of UL Transaction Security is 3.5, and in the remarks can be seen that, what the opinion is, and what can be improved. Employees, that have participated felt engaged, because they felt there was input, when presenting them with a box to fill in room for improvement. Multiple employees said there was a need for an enterprise search application, and found it easy and fun to use. The Dutch research institute was having a more positive attitude as well, with a grade of 3.15.

The main research question was:

How to evaluate an implementation of an enterprise search application, as a case study?

By looking at recent studies, literature, and by talking to employees an Enterprise Search Evaluation Framework has been created. Here an attempt has been made to evaluate the enterprise search application, that we have designed for UL Transaction Security. Every search application is different, as we saw by evaluating two search applications, but we tried to create a framework that is applicable on other enterprise search engines and that gives a clear overview on how the search application can be improved. We think, that a simple image as Figure 10 and Figure 11 together with the answers to all the surveys can help to create an overview of how well the enterprise search application performs, and how this is being seen by the users of this application.

7 Conclusion

In this section we provide the main conclusion, clarify the main contributions of this research, discuss the limitations and share ideas for future work.

7.1 Main Contributions

The main contributions of this research were developing a structured design method for enterprise search including a multi-faceted evaluation, by using the employees as an input for the requirements. The enterprise search application has been evaluated by using an evaluation framework, that did not exist before. The evaluation gives a general overview on different aspects of the search application. How mature a search application is, and how to improve the different aspects of an enterprise search application. These aspects were technical performance, query performance, usability, and user satisfaction. This all has been done in one organization as a case study, i.e., UL Transaction Security. With a validation at a Dutch research institute where the same evaluation framework was being used, here we saw that the same evaluation framework cannot be applied to every organisation, and that the evaluation framework needs to be adjusted to the targeted organisation. What can be said about the enterprise search application made for UL Transaction Security, is that overall there was a need for such an application and users were satisfied with the overall performance of the tool, what can be seen in the evaluation.

7.2 Limitations

One of the main limitations, is that in UL Transaction Security we only looked at one department, with a limited data set. Besides UL Transaction Security and the Dutch research institute more organisations could be involved in assessing the validity of the evaluation framework. By comparing different organizations, one can assess the enterprise search maturity level, now with only two organizations, this is not possible. Another limitation is time, it takes time to build an enterprise search application, and it can be easier if an organization already has an enterprise search application, which we saw in the Dutch research institute. Furthermore we only performed the evaluation with fifteen employees at UL Transaction Security, and eleven at the Dutch research institute. In an ideal situation you want to include all the employees who are going to use the enterprise search application. Other methods instead of surveys can be used, to see which method generates more participants, e.g., using structured interviews.

7.3 Future Work

As already been said in the limitations, we only looked at two organisations. For future work the evaluation framework can be tested in more organisations, compare satisfaction, usability, and query performance in different departments and organizations, this to validate the correctness of the evaluation framework. What lacked in our evaluation was the business impact. It would be interesting to see how an enterprise search application would affect the business. Due to time constraints we could not include this in our research. Other methods instead of surveys can be used, to see which method generates more participants, e.g., using structured interviews.

8 References

[1] IDC, "The enterprise workplace: How it will change the way we work," no. 32919, 2005.

[2] D. Hawking, "Challenges In Enterprise Search," *15th Australas. Database Conf. ADC 04*, vol. 27, pp. 15–24, 2004.

[3] "UL Transaction Security." [Online]. Available: https://www.ul-ts.com/. [Accessed: 10-Apr-2017].

[4] B. Ribeiro-Neto and R. Baeza-Yates, "Modern Information Retrieval," *Mod. Inf. Retr.*, vol. 9, 2011.

[5] M. Saastamoinen and K. Järvelin, "Queries in authentic work tasks: the effects of task type and complexity," *J. Doc.*, vol. 72, no. 6, pp. 1114–1133, 2016.

[6] P. Sirotkin, "On Search Engine Evaluation Metrics," 2012.

[7] M. White, *Enterprise Search*, 2nd ed. O'Reilly, 2015.

[8] C. Mooers, "Coding, information retrieval, and the rapid selector.," *Am. Doc.*, vol. 1, pp. 225–229, 1950.

[9] C. Manning, P. Raghavan, and H. Schütze, *An Introduction to Information Retrieval*, no. c. 2009.

[10] C. Cleverdon, J. Mills, and M. Keen, "Factors Determining the Performance of Indexing Systems.," 1966.

[11] M. F. Porter, "An algorithm for suffix stripping.," *Program*, 1980. [Online]. Available: http://tartarus.org/martin/PorterStemmer/.

[12] A. Broder, "A taxonomy of web search.," *ACM SIGIR Forum*, vol. 36, pp. 3–10, 2002.

[13] H. P. Luhn, "The Automatic Creation of Literature Abstracts," *IBM J. Res. Dev.*, vol. 2, no. 2, pp. 159–165, 1958.

[14] J. Zobel and A. Moffat, "Inverted les for text search engines.," *ACM Comput. Surv.*, pp. 1–56, 2006.

[15] P. Bailey, N. Craswell, and I. Soboroff, "Overview of the TREC 2007 Enterprise Track."

[16] M. Hertzum and A. M. Pejtersen, "The information-seeking practices of engineers: searching for documents as well as for people," *Inf. Process. Manag.*, vol. 36, pp. 761–778, 2000.

[17] P. Hansen and K. Järvelin, "The information seeking and retrieval process at the swedish patent- and registration office," *ACM SIGIR*, 2000.

[18] P. Thomas and D. Hawking, "Evaluation by Comparing Result Sets in Context."

[19] M. Sedlmair, M. Meyer, and T. Munzner, "Design Study Methodology: Reflections from the Trenches and the Stacks."

[20] J. Brooke, "SUS -A quick and dirty usability scale Usability and context."

[21] S. B. Huffman and M. Hochster, "How well does result relevance predict session satisfaction?," *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '07*, p. 567, 2007.

[22] A. Turpin and F. Scholer, "User Performance versus Precision Measures for Simple Search Tasks."

[23] W. Andrews and H. Koehler-Kruener, "Magic Quadrant for Enterprise Search Q3 2015," 2015.

[24] "Google." [Online]. Available: https://www.google.com/. [Accessed: 10-May-2017].

[25] "Open Source Search and Analytics · Elasticsearch." [Online]. Available: https://www.elastic.co/. [Accessed: 10-May-2017].

[26] "The Power of Fusion Search Platform | Lucidworks." [Online]. Available: https://lucidworks.com/fusion/. [Accessed: 10-May-2017].

[27] "Datafari Enteprise Search." [Online]. Available: http://www.datafari.com/en/. [Accessed: 10-May-2017].

[28] “Apache ZooKeeper - Home.” [Online]. Available: https://zookeeper.apache.org/. [Accessed: 10-May-2017].
[29] “Apache Solr.” [Online]. Available: http://lucene.apache.org/solr/. [Accessed: 10-May-2017].
[30] “Apache Lucene.” [Online]. Available: http://lucene.apache.org/. [Accessed: 10-May-2017].
[31] “Apache Tika.” [Online]. Available: http://tika.apache.org/. [Accessed: 10-May-2017].
[32] “Lucidworks View.” [Online]. Available: https://github.com/lucidworks/lucidworks-view.
[33] “Angular JS Framework.” [Online]. Available: https://angular.io/. [Accessed: 15-May-2017].
[34] “Sass: Syntactically Awesome Style Sheets.” [Online]. Available: http://sass-lang.com/. [Accessed: 15-May-2017].
[35] A. S. for P. Affairs, “System Usability Scale (SUS),” Sep. 2013.

Appendix A: Search Behaviour Survey

Objective of the survey

The objective of the survey, is to gain understanding on how employees are involved in the process of information retrieval, what their opinion is about information retrieval in general, compared on different mediums available at UL-TS Advisory Europe.

Information retrieval is the activity of obtaining information resources relevant to an information need from a collection of information resources.

This survey is part of a research project about Enterprise Search and Knowledge Extraction.

* 1. What medium(s) do you use for information retrieval? (multiple options available)

- [ ] E-mail
- [ ] Yammer
- [ ] Sharepoint
- [ ] External sources, e.g., Google, Wikipedia
- [ ] Network Shared Folders

Other (give further explaination)

* 2. Rate in terms of importantance how useful the information is provided by the specific medium. (Where 5 is the most important) With importance we mean the content of the information in the medium.

	1	2	3	4	5
E-mail	◯	◯	◯	◯	◯
Yammer	◯	◯	◯	◯	◯
Sharepoint	◯	◯	◯	◯	◯
External sources, e.g., Google, Wikipedia	◯	◯	◯	◯	◯
Network Shared Folders	◯	◯	◯	◯	◯

* 3. What do you think about the performance of the information retrieval mediums?

With performance we mean, does the medium behave like expected. E.g., speed, usability.

	Very Poor	Below Average	Average	Above Average	Excellent
E-mail	◯	◯	◯	◯	◯
Yammer	◯	◯	◯	◯	◯
Sharepoint	◯	◯	◯	◯	◯
External sources, e.g., Google, Wikipedia	◯	◯	◯	◯	◯
Network Shared Folders	◯	◯	◯	◯	◯

* 4. What is the main intent of information retrieval for each medium?

	To extend your knowledge (not project related)	To extend your knowledge (project related)	To look for specific documents	To look for specific people	I do not use this medium
E-mail	◯	◯	◯	◯	◯
Yammer	◯	◯	◯	◯	◯
Sharepoint	◯	◯	◯	◯	◯
Internet search engine, e.g., Google	◯	◯	◯	◯	◯
Wikipedia	◯	◯	◯	◯	◯
Network Shared Folders	◯	◯	◯	◯	◯

Other (Give futher explaination)

* 5. Are you able to find the information or documents, that are relevant to your work?

- ◯ Always
- ◯ Most of the time
- ◯ Sometimes
- ◯ Never

* 6. If you can't find the information you were looking for, what might be the case?

- [ ] I maybe used the wrong keywords
- [ ] I do not know how to search
- [ ] I might be using the wrong information retrieval medium
- [ ] The information may not be present in any system
- [ ] The information retrieval medium isn't working properly
- [ ] Other (give further explanation)

* 7. Where do you find the most relevant information for your work? (E.g., documents, specs, previous projects, deliverables)

* 8. What is the main challenge in retrieving this information?

* 9. Any suggestions on how to improve information retrieval? What kind of tools do you need, and what kind of resources?

10. If you want to talk about this subject, please leave your e-mail.

Appendix B: Evaluation Survey

Enterprise Search Application - Evaluation

Here we will evaluate the enterprise search tool, that has been made for UL Transaction Security. This experiment consists of three parts. First the query performance will be measured by looking for relevance. Second a usability test will be conducted, and third we will measure the user satisfaction. After the experiment is done, there is room for further feedback.

* 1. Perform a search and analyse the first twenty results. How many of these twenty presented results are relevant to the query?

Present your answer: ("search term(s)", "number that is relevant").

Search term 1:	
Search term 2:	
Search term 3:	
Search term 4:	
Search term 5:	

* 2. What do you think of the performance of the query completion?

Very bad	Bad	Neutral	Good	Very good
○	○	○	○	○

* 3. Usability Test.

	Strongly disagree	Disagree	Undecided	Agree	Strongly Agree
I think that I would like to use this system frequently	○	○	○	○	○
I found the system unnecessarily complex	○	○	○	○	○
I thought the system was easy to use	○	○	○	○	○
I think that i would need the support of a technical person to be able to use this system	○	○	○	○	○
I found the various functions in this system were well integrated	○	○	○	○	○
I thought there was too much inconsistency in this system	○	○	○	○	○
I would imagine that most people would learn to use this system quickly	○	○	○	○	○
I found the system very cumbersome to use	○	○	○	○	○
I felt very confident using this system	○	○	○	○	○
I needed to learn a lot of things before I could get going with the system	○	○	○	○	○

* 4. User Satisfaction measurement.

	Very dissatisfied	Dissatisfied	Neither	Satisfied	Very satisfied
How satisfied are you with the user interface?	○	○	○	○	○
How satisfied are you with the relevance of the results?	○	○	○	○	○
How satisfied are you with the speed of the search engine?	○	○	○	○	○
How satisfied are you with the filter function?	○	○	○	○	○
How satisfied are you with the sorting function?	○	○	○	○	○
How satisfied are you with keyword completion?	○	○	○	○	○
How satisfied are you with the ease of use?	○	○	○	○	○
How satisfied are you with how the results are being presented?	○	○	○	○	○
How satisfied are you, when comparing this tool to the previous search application?	○	○	○	○	○
How satisfied are you with the tool in general?	○	○	○	○	○

5. Further remarks?

Appendix C: Changes in Fusion View Source Code

FUSION_CONFIG.js

```
appConfig = { //eslint-disable-line
  // If you don't know what you want for some configuration items,
  // leave them as-is and see what happens in UI.
  // You may need to clear browser history/cache before your changes
take affect.

  /**
   * Styles and colors
   *
   * In addition to the functional settings in this file,
   * you can edit the settings file in
client/assets/scss/_settings.scss
   *
   * There you can edit settings to change look and feel such as
colors, and other
   * basic style parameters.
   */

  /**
   * localhost is used here for same computer use only.
   * You will need to put a hostname or ip address here if you want to
go to
   * view this app from another machine.
   *
   * To use https set the https server key and certificate. And set
use_https to true.
   */
  host: 'http://localhost',
  port: '8764',

  proxy_allow_self_signed_cert: false, // Only turn on if you have a
self signed proxy in front of fusion.

  // The port from which View will be served. defaults to 3000.
  server_port: 3000,

  // Serve View via https.
  // use_https: true,
  // https: {
  //   key: 'path/to/your/server.key',
  //   cert: 'path/to/your/server.crt'
  // },

  /**
   * The name of the realm to connect with
   *   default: 'native'
   */
  connection_realm: 'native',
```

```
  /**
   * Anonymous access
   *
   * To allow anonymous access add a valid username and password here.
   *
   * SECURITY WARNING
   * It is recommended you use an account with the 'search' role
   * to use anonymous access.
   */
  anonymous_access: {
    username: 'search-user',
  //  password: 'search-user-password-here'
  },

  // The name of your collection - defaults to Fusion 3.0 default
collection
  collection: 'default',

  // Please specify a pipeline or profile that you want to leverage
with this UI.
  query_pipeline_id: 'default',
  query_profile_id: 'default',
  use_query_profile: false, // Force use of query-profile

  // Search UI Title
  // This title appears in a number of places in the app, including
page title.
  // In the header it is replaced by the logo if one is provided.
  search_app_title: 'UL Transaction Security',
  // Specify the path to your logo relative to the root app folder.
  // Or use an empty string if you don't want to use a logo.
  // This file is relative to the client folder of your app.
  logo_location: 'assets/img/logo/ul.png',

  /**
   * Document display
   * Fusion seed app is set up to get you started with the following
field types.
   * web, local file, jira, slack, and twitter.
   *
   * Customizing document display.
   * You can add your own document displays with Fusion Seed App. You
will have to
   * write an html template and add a new directive for your document
type.
   * @see https://github.com/lucidworks/lucidworks-
view/blob/master/docs/Customizing_Documents.md
   *
   * If you want to edit an existing template for a datasource you can
edit the html for that document type in the
   * client/assets/components/document folder.
   */

  /**
   * Default Document display
   *
```

```
   * This applies only to document displays that are not handled by the
handful of
   * document templates used above.
   *
   * These parameters change the fields that are displayed in the
fallback document display.
   * You can also add additional fields by editing the document
template.
   * Default Document template is located at:
   *
your_project_directory/client/assets/components/document/document_defau
lt/document_default.html
   */
  //In search results, for each doc, display this field as the head
field
  head_field: 'title_s',
  subhead_field: 'Author',
  description_field: 'body',
  //In search results, for each doc, use this field to generate link
value when a user clicks on head_field
  head_url_field: 'title_s',
  //In search results, display a image in each doc page (leave empty
for no image).
  image_field: 'image',

  // ADDING ADDITIONAL FIELDS TO DEFAULT DOCUMENTS
  //
  // There are 2 ways to add additional fields to the ui.
  // You can either use settings to create a simple list of values with
field
  // names or you can edit the html/css, which is far more robust and
allows
  // for more customization.
  //
  // SIMPLE CONFIG BASED FIELD DISPLAY
  //
  // This is the simpler option, but wont look as good.
  // It creates a list of field names next to field results
  // in the format of:
  // field label: field result
  //
  // In order to add items to the list you must add the fields to
  // fields_to_display. You can change the label of any field by adding
a
  // field mapping in field_display_labels. You can optionally use a
wildcard '*'
  // to display all fields.
  //
  // FLEXIBLE HTML FIELD DISPLAY
  //
  // For more advanced layouts edit the document template this provides
a great
  // deal of flexibility and allows you to add more complex logic to
your results.
  // You are able to use basic javascript to show hide, or alter the
display of
  // any or multiple results.
```

```
  //
  // The HTML/Angular template is located in the following directory:
  //
your_project_directory/client/assets/components/document/document.html
  fields_to_display:['title_s','body'],
  field_display_labels: {
    'title_s': 'Document Title',
    'body': 'Description',
    'Last_Author': 'Last Author',
    'extended_properties_Company': 'Company',
    'Application_Name': 'Application Type',
      'people_ss': 'Author',
      'year_s': 'Year',
      'country_ss': 'Country'
    // you can add as many lines of labels as you want
  },

  /**
   * Number of documents shown per page, if not defined will default to
10.
   */
  // docs_per_page: 10,

  /**
   * Landing pages
   *
   * Fusion allows mapping of specific queries links (or other data)
with its
   * landing pages QueryPipeline stage.
   *
   * Default: Do not redirect but show a list of urls that a user can
go to.
   */

  // If enabled and a landing page is triggered via a query, the app
will redirect
  // the user to the url provided.
  landing_page_redirect: true,

  /**
   * Sorts
   *
   * A list of field names to make available for users to sort their
results.
   *
   * NOTE: Only non multi-valued fields are able to be sortable.
   *
   * In order to sort on a multi-valued field you will have to fix the
schema
   * for that field and recrawl the data
   */
  sort_fields: ['year_s', 'title_s'],

  /**
   * Signals
   *
```

```
   * Allow the collection of data regarding search results. The most
typical use
   * case is to track click signals for a collection.
   */
  // Signal type for title click.
  signal_type: 'click',
  // This specifies the index pipeline that will be used to submit
signals.
  signals_pipeline: '_signals_ingest', // '_signals_ingest' is the
fusion default.
  // Should be a unique field per document in your collection.
  // used by signals as a reference to the main collection.
  signals_document_id: 'id',

  /**
   * Typeahead
   *
   * Typeahead or autocomplete shows you a number of suggested queries
as you
   * type in the search box.
   */
  typeahead_use_query_profile: false,
  typeahead_query_pipeline_id: 'default',
  typeahead_query_profile_id: 'default',
  typeahead_fields: ['title_s'],
  // The request handler defines how typeahead gets its results.
  // It is recommended to use suggest as it is more performant.
  // It will require some additional configuration.
  // @see https://lucidworks.com/blog/2016/02/04/fusion-plus-solr-
suggesters-search-less-typing/

  //typeahead_requesthandler: 'suggest', // recommended (requires
configuration)
  typeahead_requesthandler: 'select',

  /**
   * Default query
   *
   * If there is no query provided in the URL this query will be used.
It is in object form.
   */
  default_query: {q:'*'}
};
```

document_file.html

```
<div id="documentList">
  <div ng-repeat="doc in vm.docs" ng-switch="::vm.getDocType(doc)">
    <!-- <document-example ng-switch-
when="string_matching_data_connector" doc="doc"
highlight="vm.highlighting"></document-example> -->
    <document-file ng-switch-when="lucid.anda/file" doc="doc"
position="vm.getDocPosition(doc,vm.docs)"
highlight="vm.highlighting"></document-file>
    <document-jira ng-switch-when="lucid.anda/jira" doc="doc"
position="vm.getDocPosition(doc,vm.docs)"
highlight="vm.highlighting"></document-jira>
```

```
    <document-slack ng-switch-when="lucid.slack/slack" doc="doc"
position="vm.getDocPosition(doc,vm.docs)"
highlight="vm.highlighting"></document-slack>
    <document-twitter ng-switch-
when="lucid.twitter.search/twitter_search" doc="doc"
position="vm.getDocPosition(doc,vm.docs)"
highlight="vm.highlighting"></document-twitter>
    <document-twitter ng-switch-
when="lucid.twitter.stream/twitter_stream" doc="doc"
position="vm.getDocPosition(doc,vm.docs)"
highlight="vm.highlighting"></document-twitter>
    <document-web ng-switch-when="lucid.anda/web" doc="doc"
position="vm.getDocPosition(doc,vm.docs)"
highlight="vm.highlighting"></document-web>
    <document-default ng-switch-default doc="doc"
position="vm.getDocPosition(doc,vm.docs)"
highlight="vm.highlighting"></document-default>
  </div>
<!-- Regular grouped results -->
  <div ng-if="vm.groupedResults" ng-repeat="(key, val) in
vm.groupedResults">
    <h1>{{key}}</h1>
    <div ng-repeat="group in val.groups">
      <h3 ng-if="group.groupValue" ng-
click="vm.toggleGroupedResults(group.groupValue)">{{group.groupValue}}<
/h3>
      <!-- TODO refactor this -->
      <div ng-repeat="doc in group.doclist.docs" ng-
switch="::vm.getDocType(doc)" ng-
show="vm.showGroupedResults[group.groupValue] ||
vm.showGroupedResults['noGroupedValue']">

        <!-- <document-example ng-switch-
when="string_matching_data_connector" doc="doc"
highlight="vm.highlighting"></document-example> -->
        <document-file ng-switch-when="lucid.anda/file" doc="doc"
position="vm.getDocPosition(doc,vm.docs)"
highlight="vm.highlighting"></document-file>
        <document-jira ng-switch-when="lucid.anda/jira" doc="doc"
position="vm.getDocPosition(doc,vm.docs)"
highlight="vm.highlighting"></document-jira>
        <document-slack ng-switch-when="lucid.slack/slack" doc="doc"
position="vm.getDocPosition(doc,vm.docs)"
highlight="vm.highlighting"></document-slack>
        <document-twitter ng-switch-
when="lucid.twitter.search/twitter_search"
position="vm.getDocPosition(doc,vm.docs)" doc="doc"
highlight="vm.highlighting"></document-twitter>
        <document-twitter ng-switch-
when="lucid.twitter.stream/twitter_stream"
position="vm.getDocPosition(doc,vm.docs)" doc="doc"
highlight="vm.highlighting"></document-twitter>
        <document-web ng-switch-when="lucid.anda/web" doc="doc"
position="vm.getDocPosition(doc,vm.docs)"
highlight="vm.highlighting"></document-web>
```

```
        <document-default  ng-switch-default doc="doc"
position="vm.getDocPosition(doc,vm.docs)" highlight="vm.highlighting"
></document-default>
      </div>
    </div>
    <!--Simple Grouped Results -->
    <div ng-repeat="doc in vm.groupedDocs" ng-
switch="::vm.getDocType(doc)" ng-
show="vm.showGroupedResults['simpleGrouped']";>
      <!-- <document-example ng-switch-
when="string_matching_data_connector" doc="doc"
highlight="vm.highlighting"></document-example> -->
      <document-file ng-switch-when="lucid.anda/file" doc="doc"
position="vm.getDocPosition(doc,vm.docs)"
highlight="vm.highlighting"></document-file>
      <document-jira ng-switch-when="lucid.anda/jira" doc="doc"
position="vm.getDocPosition(doc,vm.docs)"
highlight="vm.highlighting"></document-jira>
      <document-slack ng-switch-when="lucid.slack/slack" doc="doc"
position="vm.getDocPosition(doc,vm.docs)"
highlight="vm.highlighting"></document-slack>
      <document-twitter ng-switch-
when="lucid.twitter.search/twitter_search"
position="vm.getDocPosition(doc,vm.docs)" doc="doc"
highlight="vm.highlighting"></document-twitter>
      <document-twitter ng-switch-
when="lucid.twitter.stream/twitter_stream"
position="vm.getDocPosition(doc,vm.docs)" doc="doc"
highlight="vm.highlighting"></document-twitter>
      <document-web ng-switch-when="lucid.anda/web" doc="doc"
position="vm.getDocPosition(doc,vm.docs)"
highlight="vm.highlighting"></document-web>
      <document-default  ng-switch-default doc="doc"
position="vm.getDocPosition(doc,vm.docs)" highlight="vm.highlighting"
></document-default>
    </div>
  </div>
</div>
```

document_file.js

```
(function () {
  'use strict';

  angular
    .module('lucidworksView.components.documentList',
['lucidworksView.services.config',
      'ngOrwell', 'lucidworksView.services.landingPage'
    ])
    .directive('documentList', documentList);

  function documentList() {
    'ngInject';
    return {
      restrict: 'EA',
      templateUrl: 'assets/components/documentList/documentList.html',
      controller: Controller,
      controllerAs: 'vm',
      bindToController: {},
```

```
      scope: true,
      replace: true
    };

  }

  function Controller($sce, $log, $anchorScroll, Orwell) {
    'ngInject';
    var vm = this;
    vm.docs = [];
    vm.highlighting = {};
    vm.getDocType = getDocType;
    vm.groupedResults = false;
    vm.toggleGroupedResults = toggleGroupedResults;
    vm.showGroupedResults = {};
    vm.getDocPosition = getDocPosition;

    activate();

    ////////

    function activate() {
      var resultsObservable = Orwell.getObservable('queryResults');
      resultsObservable.addObserver(function (data) {
        vm.docs = parseDocuments(data);
        vm.highlighting = parseHighlighting(data);
        vm.getDoctype = getDocType;
        $anchorScroll('topOfMainContent');
      });
    }

    /**
     * Get the document type for the document.
     * @param  {object} doc Document object
     * @return {string}     Type of document
     */
    function getDocType(doc){
      // Change to your collection datasource type name
      // if(doc['_lw_data_source_s'] === 'MyDatasource-default'){
      //   return doc['_lw_data_source_s'];
      // }
      return doc['_lw_data_source_type_s'];
    }

    /**
     * Decorates the document object before sending to the document
directive.
     * @param  {object} doc Document object
     * @return {object}     Document object
     */

    function isNotGrouped(data){
      return _.has(data, 'response');
    }
    function isGrouped(data){
      return _.has(data, 'grouped');
    }
```

```
    /**
     * Get the documents from
     * @param  {object} data The result data.
     * @return {array}       The documents returned
     */
    function parseDocuments(data){
      var docs = [];
      if (isNotGrouped(data)) {
        docs = data.response.docs;
      }
      else if(isGrouped(data)){
        vm.groupedResults = data.grouped;
        parseGrouping(vm.groupedResults);
      }
      return docs;
    }


    function toggleGroupedResults(toggle){
      vm.showGroupedResults[toggle] = !vm.showGroupedResults[toggle];
    }

    function parseGrouping(results){
      _.each(results, function(item){
        if (_.has(item, 'groups')){
          _.each(item.groups, function(group){
            if(_.has(group, 'groupValue') && group.groupValue !==
null){
              vm.showGroupedResults[group.groupValue] = false;
            }
            else{
              vm.showGroupedResults['noGroupedValue'] = true;
            }
          });
        }
        else{
          vm.groupedDocs = item.doclist.docs;
          vm.showGroupedResults['simpleGrouped'] = true;
        }
      });
    }

    /**
     * Get highlighting from a document.
     * @param  {object} data The result data.
     * @return {object}      The highlighting results.
     */
    function parseHighlighting(data) {
      if (data.hasOwnProperty('highlighting')){
        _.each(data.highlighting, function(value, key){
          var vals = {};
          if (value) {
            _.each(Object.keys(value), function (key) {
              var val = value[key];
              _.each(val, function(high){
                vals[key] = $sce.trustAsHtml(high);
              });
```

```
            });
            vm.highlighting[key] = vals;
          }
        });
      }
      else{
        vm.highlighting = {};
      }
      return vm.highlighting;
    }

    /**
     * Get index of the doc in the returned documentList
     * @param  {object} doc  Doc of which the index is required
     * @param  {object} docs List of returned documents
     * @return {number}      The index of the document in the
documentList
     */
    function getDocPosition(doc, docs){
      return _.findIndex(docs, doc);
    }
  }
})();
```