University of Bergen

Master Thesis

Wikipedia based Query Expansion for Searching in Norwegian

Author:

Lars Petter Trydal Johnsen

Department of Information Science and Media Studies

1. June 2015

Table of Contents

Illustration Index

Index of Tables

Abstract

With the rapid increase of information available on the Internet, searching is becoming a daily operation. This thesis address the issue that arises when there is a difference between what the users are typing in the search engine, and what they actually are interested in. The method presented in this thesis is a Query expansion method using Wikipedia as a knowledge-base, with the goal of reducing the gap between what the users are formulating, and the actual information need behind the search. The method is based on analysis of hyperlinks between articles in Wikipedia. The research contribution of this thesis is a new method for query expansion designed for Norwegian language. The proposed method uses the knowledge-base to suggest words for the users that are relevant to the original query. The evaluation of the new method shows a marginal increase in quality, but highlights the need for a search engine designed for Norwegian language. The thesis concludes that a search option for Norwegian language will produce better results, and the system for suggestion of words might help users move from a broad search strategy to a more precise search strategy.

Acknowledgements

First and foremost I would like to thank my supervisor Weiqin Chen for guidance and motivation throughout the process of this thesis.

I would like to thank my mom and dad, Ingebjørg Trydal and Lasse Johnsen for the bearing with me, despite my constant nagging, helping with the thesis, and making sure that I was able to sustain my own life.

Finally, I would like to thank my colleagues at Room 637 for constant diversions, help with the thesis, Weekly quiz and maintaining caffeine intake. The five past years would not be the same without you.

1 Introduction

1.1 Background

Today we are in what we can call information era. On the Internet today there is so much information, that the question is no longer *if* it exists on the Internet, but *how* to find the existing information. The amount of information available on the Internet is so massive that it is not possible for the user to find the information they need on their own. This is the reason for the growth of search engines on the web. The search engines are designed to help users fill the information need the users have. The users have gotten a *new* way of finding the information they need.
Google.com is at the time of writing ranked as the most used page on the Internet according to the company Alexa.com. With over 35 million searches performed globally per day, we can safely assume that *searching* is a daily activity for many humans.

"*Whenever we seek out new knowledge – Whenever we turn to the ubiquitous search engines – we must grapple with the same fundamental paradox: how can one describe the unknown*?" (Milne, Witten, & Nichols, 2007)

The problem arises when the users need to formulate a query to find unknown information. How done one formulate an efficient search query towards unknown information?

One measure to help the user formulate a query is query expansion. Query expansion is a technique to increase the quality of the search results by adding terms or weight the different terms in the query (Manning & Raghavan, 2009). The main aim is to minimize the semantic gap between what the users actually want to find information about, and what they are actually writing as a query.

There are no open source search engines for Norwegian language as of now that support query expansion. There are some other commercial actors that offer intra-net search engines in Norwegian, but the engines are not publicly available.

Kvasir.no performs searching in Norwegian, but is based on a specially adapted service from Google, that increases the weighting of pages written in Norwegian. *SearchDaimon* is a Norwegian enterprise search engine, that was released as open source engine in 2013, but this open source engine do not include query expansion, or any other method for helping the user formulate the query.

1.2 Research Question

For this thesis the following research questions are formulated to help limit and refine the scope of the thesis. The first research questions I have formulated is:

"How to improve the search experience and results in Norwegian by query expansion using Wikipedia as a knowledge base?"

The knowledge base is modeled by Wikipedia, and uses link analysis between the articles to extract information. The amount of information that is implicitly stored in Wikipedia by link analysis are massive and therefore it is possible to increase the quality of searching by using this data as background knowledge. To improve the search experience the engine presents words relevant to the user input.
As a consequence of the research question it is of utmost importance to determine if it is possible to model relatedness between words using graph algorithms.

The aim for this thesis is to present a new search engine designed for Norwegian language. The features that set this search engine apart from other options are:

- The search engine use Norwegian Language algorithms, that allows for searching in Norwegian without impacting the quality of the search.
- The search engine is designed to enhance the search experience for Norwegian users, by supporting query expansion for Norwegian language.
- The query expansion method is using a knowledge base derived from the Norwegian Wikipedia to recommend search terms for the users. By using Wikipedia as a knowledge base it is

possible help the users to accurately and efficiently perform searching in their native language, by helping the users to formulate a query that covers their *information need*.

This project is justified by the lack of both commercial market participants and open source service providers. There are just one search engine that supports Norwegian language and no query expansion providers. Additionally, there is little to no research on the subject.
The amount of searching the Internet users has to perform each time they seek out new information are so massive that the lack of good alternatives to perform this operation might take a lot more time than the user wants to spend on searching.

The idea is to help the user formulate a query that reflects the user information need, using query expansion. The system extracts information from Wikipedia, by analyzing the links between articles and recommend terms to the user. The terms that the user selects to be relevant to the search are included in the expanded search. By performing this procedure we can help to user define a context for the search, and bridge the semantic gap between the users information need and the query the user has formulated. With this process the user can specify the query for the search, *before* performing the actual search, and thus avoid having to reformulate the query and perform multiple searches.

In the early days of search engines query expansion was a very expensive and difficult task to create and keep updated because of the high cost of manually producing and updating a knowledge base. The normal thesauri and dictionaries were not rich enough to cover the information needs for a search engine, for example the vocabularies in the scientific fields (Manning & Raghavan, 2009).

Since the birth of Information Retrieval (IR), Wikipedia, the online encyclopedia has been founded and become of the largest sites on the Internet. At the time of writing Wikipedia is the sixth most popular page on the Internet, with over thirty five million pages on the English Wikipedia page. Wikipedia also exits for many different languages including Norwegian. By using Wikipedia as a knowledge base we avoid the problem of manually creating and updating a knowledge base as the Wikipedia users updates the data.

1.3 Organization of the Thesis

This thesis has 6 chapters. Each chapter will present a different aspect or topic. The first chapter is focused on motivation and justification for the thesis. The second chapter givens an overview of the theoretical concepts and related research concerning the topics. The second chapter is also aimed at giving the reader an understanding of the theoretical concepts used in this thesis.

The third chapter is a presentation of the Design Science Research Framework. The framework is a scheme for research standards through development as a research strategy. The framework is used in this project to maintain research focus through system development.

The fourth chapter presents the development process and the implementation specific details meant to explain and present the system, and the process leading up to the finished prototype.

The fifth chapter is the evaluation chapter were the designed prototype is evaluated against existing constructs.

In the final conclusion chapter, the results from the evaluation will be discussed, focusing on future possibilities, and how the research covers the research questions formulated in chapter one.

2 Literature review and theoretical concepts

This chapter of the thesis will present the theoretical foundation for this project. It will explain the theoretical concepts used in the thesis, with concepts used in the system. It will present Query expansion, Language algorithms, and structural data from graphs. The last part of this chapter will present the traditional methods for evaluation of information retrieval applications.

2.1 The Search Process

When a user want to cover an information need through a search, the user need to perform a series of steps to determine the best possible way to achieve this result. This process was described by *Sutcliffe* and *Ennis (1998) as;*

- Problem identification
- Articulating the information need
- Formulate a query
- Evaluation the results.

This process assumes that the information need of the user is static and does not change through the process. The dynamic counterpart to this process assumes that the users information need can change through the process. The static model proposed by Sutcliffe and Ennis is regarded by many as the foundation for the more dynamic models.

One crucial part of the process is the articulation of the query. Natural language are not unambiguous, and there are multiple ways of describing something. Subsequently, a gap can occur a gap between what the users' information need actually are, and what the users formulate in their query. This is called a *semantic gap*. In the traditional search engines it is up to the user to determine how to formulate the query in the best possible way to cover the information need. By doing this we force the user to perform several *query reformulations.* For example, if the user has an information need regarding a

subject of which the user does not have much background information about, it will be very difficult for the user to formulate a query that covers the users need (Sutcliffe A, Ennis, 1998).

2.1.1 Information need

In web context the need behind the query is often not informational. Andrei Broder (2002) divided the web queries according to their intent into 3 different classes:

1. Navigational. The immediate intent is to reach a particular site.
2. Informational. The intent is to acquire some information assumed to be present on one or more web pages.
3. Transactional. The intent is to perform some web-mediated activity.

Navigational

Navigational queries are the wish to find a particular site, that the user has in mind. For example if a user wants to find the webpage for a train company, the user is likely to perform a search with a navigational intent.

One important note about navigational searches are that the user already knows what the desired target are. If the user wants the homepage for the local train company, that is what the user are searching for (Broder, 2002).

Informational

The purpose of informational queries are to find information assumed to be available on the web in a static form. The only expected interaction from the user is *reading*.

Transactional

Transactional queries are queries to reach a site where further interaction will happen. This interaction are transactions defining thees queries. For example finding the yellow pages, for looking up a phone number (Broder, 2002).

The search engine presented in this thesis does not include a crawler and is not primarily designed for usage on the web, but is mainly designed for finding documents in a large collection of documents exclusively in Norwegian language. Therefore, we fill mainly focus on the informational query need.

2.1.2 Query expansion

One measure to help the user formulate a query is query expansion. Query expansion is a technique to increase the search results by adding terms or weight the different terms in the query. The main aim is to minimize the semantic gap. By performing this process we can minimize the semantic gap between information need and query, and thus make the searching processes more efficient.

Traditionally query expansion has been used to eliminate spelling errors, or used logs from previous search to recommend new search terms. Query expansion are divided into two main classes, *local* or *global.* Local methods adapts the query to relatively fit better with the document that's the initial best match. Global methods will traditionally analyze what words the occurs together in a collection of documents (Xu & Croft, 1996).

Query expansion tends to improve the average retrieval performance, doing so by improving performance on some queries while reducing performance on others. Query expansion techniques are judged as effective in the cases that they help more than they hurt overall on a particular collection (Custis & Al-Kofahi, 2007).

2.1.3 Searching strategies

The goal of any searching process is to find a solution to a problem with minimal cost. The cost can be both in effort and in processing time. Different users have different strategies to formulate the initial query.

A regular user might have an information need regarding “Mona Lisa”. A web search with this query will yield a very large amount of hits covering many topics both loosely and tightly regarding the input query. A more experienced user might try to narrow the search by adding “painting” or “art” to the query to narrow down the search to the painting. A very experienced user might use knowledge the he or her has about the subject and include “Leonardo DaVinci” as a term in the query, further narrowing down the results (Yamin & Ramayah, 2001).

The strategies mentioned above are strategies known as *Breadth search queries.* The query formulated is general, wide and not focused on the domain. Another strategy for searching is the *Depth Search Query.*

Depth query strategy is a narrow usage of query, and can be divided into three main strategies: Boolean operator, the use of computer convention, and complex search.

The Boolean operator is the usage of Boolean operators, AND, OR, NOT, to refine the scope of the query. The use of computer convention is to use knowledge about computer to efficiently search. For example searching for query + .jpg if the user is interested in a picture. The last strategy, the complex search is a search directly at the subject. For example the Mona Lisa query might be formulated as "Mona Lisa Picture Louvre". This narrows down the results to the actual subject (Yamin & Ramayah, 2001).

Breadth First strategy	**Depth First Strategy**
Breadth	Boolean
Wide search definition	Computer convention
General knowledge search	Complex Search

Table 1: Different search strategies

All users of computers are different. Any successful search engine has to cater towards all the different search strategies. Different users will use different strategies. A user with little to none prior knowledge of the inner workings of a computer or search engine will very often use one of the breadth query strategy, while more experienced users will more likely use the depth query strategy.

2.2 Similar Work

Bunescu and Pasca suggested in 2006 to use Wikipedia data as knowledge base in a search engine to differentiate entities from each other. This article is the first credited use of Wikipedia as a knowledge base, and formed the foundation of further research in query expansion. In the article the authors demonstrated that it is possible to dramatically improve the results of a search by getting the user to define what entities that the user was looking for (Bunescu & Pasca, 2006).

Øyvind Døskeland wrote in 2012 a master thesis regarding using Wikipedia as a knowledge base. This thesis used structural data and usage statistics to suggest terms to the user. Døskeland presents results

that show major improvements of results by user testing, even beating Google on some measurements. Døskeland used artificial intelligence methods to augment queries based on Wikipedia (Døskeland, 2012). This thesis is the starting point for my thesis.

Arantxa Otegi and Xabier Arregi, proposed in 2011, a query expansion method starting in the original query terms and uses Wordnet to suggest related words the to the user improve the results (Otegi, 2011). The suggestion of the words are based on relatedness of the words. The equivalent to Wordnet in Norwegian *"Ordnett",* is not substantial enough to facilitate general, multi topic query expansion, but one of our hypothesis is that it is possible to replicate this by analyzing Wikipedia

Knowledge-based Query expansion to support scenario specific retrieval of medical free text from Chu, Zhenyu Liu and Wesley W. This article proposes a method for query expansion recommendations by creating domain knowledge. With domain knowledge it is possible to recommend specializations and generalizations of a query (Chu, 2007).

The THT system, a system proposed by Feil and Christensen is an article about a multi-language information retrieval system aimed at the Norwegian languages. The system performs retrieval across the different languages. This article is relevant for adapting the system to cope with Norwegian language (R. Feil, 2015).

In this thesis we will use graph data to recommend query terms to the user. The article "*Massive Query expansion by exploiting graph Knowledge bases"* by Guisado-Gamez used the same principles in their article. This project uses Wikipedia as a knowledge base and creates topics, called *concepts*, and lining between the concepts using link analysis and community detection. This method showed a significant increase in the quality of the results compared to the baseline. The experiments also show a correlation between the precision of the system and the quality of the knowledge base, that suggests that advances in more complete and customized knowledge bases will provide better search engines (Guisado-Gámez, Dominguez-Sal, & Larriba-Pey, 2013). This aspect is an interesting factor in this thesis, based on the fact that this is the first attempt at using the Norwegian Wikipedia as a knowledge base, compared to the English version, that is substantially larger, both in usage and in pages. This sparks an

interesting point, is the Norwegian version of Wikipedia large enough to effectively be used as a knowledge base for query expansion.

This thesis will use the concepts presented in the mentioned articles and apply the theoretical frameworks presented to create a search engine designed for Norwegian language. Query expansion using Wikipedia as a knowledge base have been done for English language using the English Wikipedia, but it has not been done for Norwegian language.

2.3 Norwegian Language

At the date of writing there are no open systems the supports query expansion in Norwegian. Norwegian language behaves differently than English and it's based on this it would be advantageous to get a search engine with query expansion that's designed for Norwegian languages.

This chapter will present the theoretical foundations for a search engine designed for Norwegian language. The implementation specifics will be presented in chapter 4 of the thesis, but in this chapter the theoretical concepts will be presented.

2.3.1 Tokenization

Tokenization is the process of splitting the text down to individual words. The tokenizer does often also perform character normalization. When inserting a document into a search engine, we perform an operation called an *indexing*. At the start the text from the document is one long string of text, with different case letters, punctuation and many elements. To be able to treat the different documents in a search engine we need the words to be in a form that is universal for all the occurrences of the word. The first process will be to tokenize the words.
To tokenize is to split character streams into tokens (Manning & Raghavan, 2009).

"A token is an instance of a sequence of characters in some particular document grouped together as a useful semantic unit for processing. A type is the class of all tokens containing the same character

sequence. A term is a (perhaps normalized) type that is included in the IR system's dictionary. "(Manning & Raghavan, 2009)

Typically, the tokens included in the dictionary are *normalized.*
When the documents are broken down into tokens, it should be easy to match a query to a document using matching. But the term "Jumping", does not match the word "jumping!". This is the reasoning behind normalization. In this process we remove punctuation, set all words to lowercase and remove special characters, such as hyphens, and exclamation marks. This creates equivalence classes of tokens. Tokens that were different before the process can be the same after normalization.

Norwegian language is a Germanic language and uses word merging. This means that words like *"Bokhylle"* is a merging of the two words bok (book) and hylle (shelf). The merging structure of the language is a feature that is hard to counter in the IR sense, and means that the users have to write correctly, and use the word merging correctly when performing a search (Vikør, 2010).

2.3.2 Stop words

In both the indexing process and the searching process the system is set up to remove words that carries little to no semantically important information. These words are known as *stop words.* Stop words have little to no value in the process of selecting documents that the user needs. Therefore, we exclude them from the process entirely (Manning & Raghavan, 2009).

In this project a list of Norwegian words, that consists of binding words with little to no meaning for the rest of the document has been used. This list is compiled by the creators of the snowball stemmer, and revised by Jan Bruusgaard (Bruusgaard, 2005)
The Snowball stemmer will be explained in section 2.3.3.

There are considerations that needs to be made when considering stop words. The user experience can be more challenging, because the user is not used to the fact that certain words in the input field are removed before the search is completed. Additionally, if you remove too many stop words it can affect the precision of the search process, but it makes the indexing process much cheaper to perform.

The amount of words in the stop words list and what words the remove from the documents varies from search engine to search engine. The major search engines does often not remove stop words at all. This can be done because of the weighting of the words, means that the words doesn't have an effect on the search either way. (Manning & Raghavan, 2009)

Many of the larger search engines for searching the web, does not remove stop words at all. There are several reasons for this. The hardware that these companies has available to perform the retrieval operations on, are so powerful that the processing of stop words has no impact on the performance. Secondly the weighting methods that they use in these engines are tuned so that all stop words receive a very low score, thus effectively removing them from the retrieval search. By using this method you do not run the risk of removing to words that may affect the precision of the retrieval task

2.3.3 Stemming and Lemmatizing

In languages the same word often have different forms, *jumping, jumped.* When we search we want these words to be treated as the same, because their semantic meaning are the same. To be able to treat the words as the same we need to reduce the words down to the stem of the word.

Norwegian Snowball Stemming.

Norwegian Snowball stemming is a Norwegian algorithm variant of the Porter stemmer algorithm, *snowball stemmer. (Porter, 1979).* This stemming algorithm is written by Martin Porter and supports 12 different languages including Norwegian. This is the only algorithm for stemming in Norwegian that's publicly available.
There are many other stemming algorithms available, but there is no other than Porters algorithm that supports Norwegian language.

Porter stemming algorithm works by through several steps it reduces the word down to it's *stem,* the grammatical root form of the word. (Porter, 1979)

An Alternative may be to lemmatize the words using a more complex algorithm, and there exists some Lemmatizing algorithms that supports Norwegian, but the lemmatization process is very computationally expensive.
Lemmatizing has the same goal as stemming, to reduce inflectional and sometimes related forms of a word, to a common base word. While stemming is a crude, heuristic process for reducing the words, lemmatization uses vocabularies and morphological analysis, to achieve the same goal (Manning & Raghavan, 2009).

The techniques presented in this part of the thesis is the over theoretical concepts required to transform a set of documents in Norwegian to an index that it is possible to perform retrieval tasks on.

2.3.4 Character Encoding

Character encoding is used to represent characters in some form encoding system. In computers, different character sets are represented in different encoding. Norwegian language have more characters than English and are represented by a different encoding system. To represent Norwegian, the regular encoding is UTF-8. The normal encoding for representing English on the web is Unicode.

2.4 Structural Data

In large datasets like Wikipedia there are large amounts of data stored implicitly. We can analyze the articles and the connections between the articles. In small datasets this will provide a small amount of information, but in datasets like Wikipedia, it is possible to mine large amounts of data from the structure.

2.4.1 Pagerank

Pagerank is an algorithm developed by the creators of Google, to help sort the rankings from an online search. The main idea is to calculate the relative importance of a node, based on outgoing and ingoing links from a site. Google explains the main idea of the Algorithm as: “Pagerank works by counting the number and the quality of links to determine a rough estimate of how important the website is. The

underlying assumption is that more important websites are likely to receive more links from other websites (Manning & Raghavan, 2009).

In other words the algorithm analyzes the ingoing and outgoing links of a node, and uses the assumption that an importance website has many ingoing links and subsequently the links going out of an important site is helping the relative importance of the sites it's pointing to.

Let P be the number of outgoing links from a page *p,* and assume that page an is linked to one of the pages from p_1 to p_n.
Then the Pagerank value for page a will be given by the probability Pr(a):

$$PR(a) = (\frac{q}{T}) + (1 - q) \sum_{i=1}^{n} (\frac{PR(p_i)}{L(p_i)})$$

Where *T* is the total number of pages on the web graph, and *q* is a parameter set by the system(typically 0,15).

As an example; if *bbc.com* links to a smaller, more obscure site, then it's more probable that the smaller site is relatively more important than a site that gets five links in from small niche sites.
Pagerank is modeled by a series of random jump between sites on the Internet, controlled by a Markov-chain. A Markov chain is a discrete, time stochastic process that occurs in a series of random steps are performed (Manning & Raghavan, 2009). The parameter *q* in the definition is influenced by the Markov chain, and represents the probability for a user to switch to a totally different page at a random step in the process.

In this application i have primarily used Pagerank to weight the importance of nodes and in my graph each node is stored with a property value that contains the Pagerank value. This allowed for quick accessing of the property without having to perform the operation multiple times. On the other hand it increases the size of the database. The calculation was performed using the program called Mazerunner. Mazerunner allows for batch calculation of Pagerank values, and thus makes it possible to perform the operation without holding the whole graph in memory.

2.4.2 Pagerank as a measure of importance

In my project I use the Pagerank algorithm to measure the relative importance of nodes. This presents an issue because there are no obvious correlations between the Pagerank values of articles and the semantic meaning of the word. When the user is trying to formulate the search term, they are mainly focused on the semantic meaning of the words they are typing. Using Pagerank as a measurement of importance, is merely an approximation of the importance of the word, and an estimation of what the users information need are.
The figures below the figure are illustrations of the theoretical concepts.

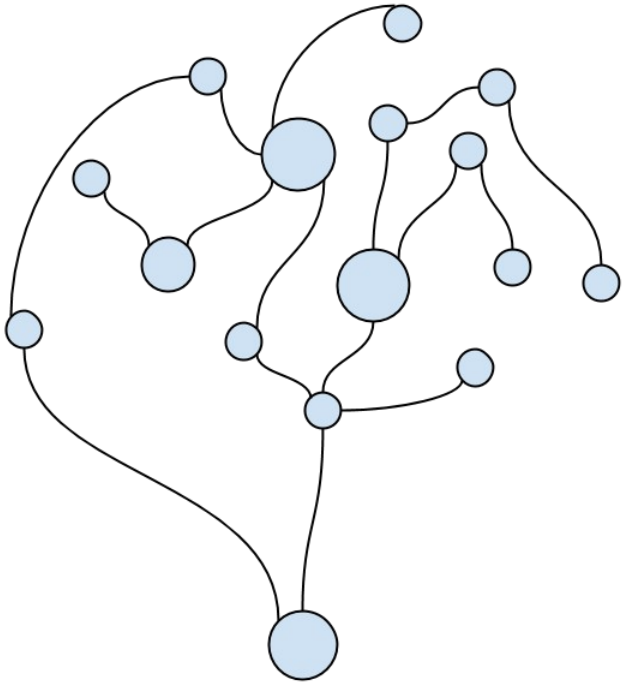

Illustration 1: Demonstration of importance in the graph

The size of the nodes represent the relative importance in the graph.

2.4.3 Community detection

Social networks of various kinds demonstrate a strong community effect. Actors in networks tend to form closely-knit groups (Tang & Liu, 2010).

We can use community structure to analyze what articles are influential on other articles.

In the article *"Exploring Local Community Structures in Large Networks"* by Feng Luo, James Z Wang and Eric Promislow, the authors proposed a method for community detection from a seed node.

The method proposed a method of finding a community in a given graph based on one node as an input. The proposed method has a complexity of $O(K^2d)$, where K is the number of nodes to be explored in the sub-graph and d is the average degree of all the nodes in the sub-graph.(Luo, Wang, & Promislow, 2006)

Many definitions of community or modules in graphs has been proposed, but the authors define a community or modules as a sub network that has more internal edges than external edges (Luo et al., 2006).

The degree of a node is the number of links pointing to a node in a graph. In a directed graph, the degree can be divided into *in-degree* and *out-degree.* In-degree is the number of edges to the node, and out-degree is the number of edges in to the node from other nodes.

The method is based on local modularity optimization. The modularity ***M*** of a sub-graph S in a given graph G is defined by the ratio of the *in-degree* ind(**S**), and the *out-degree* outd(S).

$$M = \frac{ind(S)}{outd(S)}$$

By using this definition of modularity the modularity will increase when subgraph S has more internal links than external.

Given a graph G, a subgraph S is a community if M > 1.This definition generally captures the general concept of a community.(Luo et al., 2006)

Given a graph G and a seed node N the algorithm will use to steps to find a sub-graph S that has the maximum local modularity: addition and deletion. The starting point for the algorithm is a sub-graph S that only has one node, the starting node. From the set of the Neighbors of the initial node, the algorithm iteratively adds one node at a time. The goal of this step is to maximize the modularity M of the sub-graph(S). The deletion step removes nodes from the sub-graph iteratively. To increase the modularity. Nodes adjacent to those newly added to S, will be included in the set of neighbors. This process will be repeated to no new nodes is added to the sub-graph S. (Luo et al., 2006).

The relatively low complexity of this algorithm makes it applicable for this project, and makes it possible to extract communities from subgraphs from each node and use this information to maximize the retrieval results.

One note on using this community detection algorithm; in many cases when calculating communities in a graph it's important to know the "*ground truth",* the facts about what is connected and what is not. This is not applicable to our system, because the system is designed to be generalized, to fit every purpose. Therefore, the estimation of a community is an approximation designed to find the best possible sub-graph.

As previously mentioned there have been proposed many methods for community detection. One of the major advantages of selecting this algorithm is the procedure for the community detection. The majority of community detection algorithms are network centric, and calculates communities for the graph as a whole. This algorithm calculates communities from a seed-node, that facilitates for changes in the graph over time.

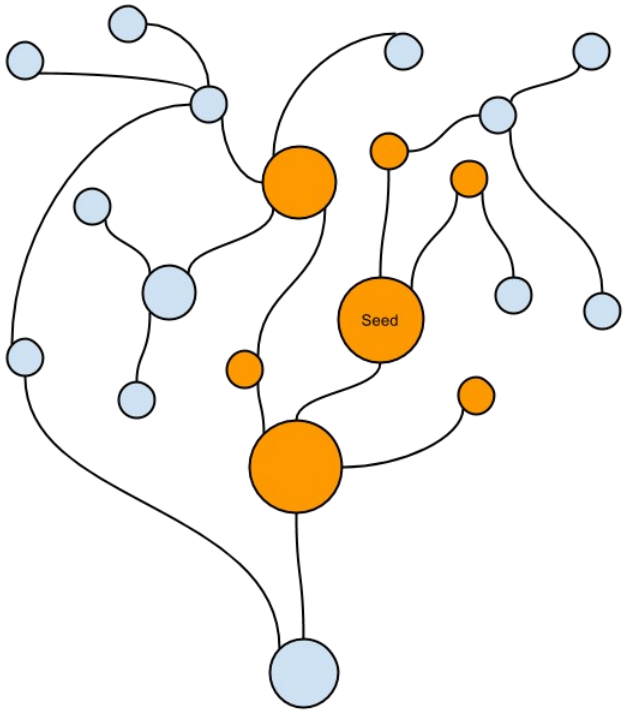

Illustration 2: Example of community detection

The nodes highlighted in orange are an example of a community returned by the algorithm.

2.5 Wikipedia

This chapter will present and discuss Wikipedia, and Wikipedia usage as a knowledge base.

Wikipedia, *the online encyclopedia,* was founded in 2001 by Jimmy Wales and Larry Sanger. Wikipedia was intended to complement Nupedia, an encyclopedia project where the articles where solely edited by experts. The idea behind Wikipedia is that everyone can freely create and edit articles about any given subject, and revisions are performed by other users. The Norwegian version of Wikipedia was founded in the same year as the English, but remained inactive for some time before it saw any use (Zahl, 2009).

At the time of writing, the Norwegian Wikipedia contains over 400,000 articles (Wikipedia Foundation, 2015).

2.5.1 The Wikipedia Process

As previously mentioned, Wikipedia is updated and edited by users, without expert involvement. This creates a constant cycle of improvement performed by users. This method for quality assurance are widely criticized in academic circles, because it means that Wikipedia will often contain errors, lack of citation and vandalism.

In 2008 published Robert Fjellstad a Master thesis regarding the quality of the Norwegian Wikipedia. In this thesis Fjellstad added errors to articles about natural sciences., to determine if they were found and corrected. He concludes that Wikipedias process for eliminating errors are focused on stopping vandalism, and factual errors, will most likely only be discovered if the original author have an ownership relationship with the article (Fjellstad, 2008).

Svenn Arne Nilsen wrote a Master thesis in 2009, regarding the quality of Norwegian Wikipedia articles compared to “Store norske leksikon”, a large encyclopedia written by experts. In 9 out of 10 articles regarding natural sciences the expertly written encyclopedia had higher quality than Wikipeda.

Snorre Zahls Master also wrote a thesis comparing the quality of Natural Science articles on Wikipedia about “Norwegian vertebrates” with the “Store Norske Leksikon”. The thesis present results where about half of the articles are best on Wikipedia, and the other half Store Norske Leksikon have the best articles (Zahl, 2009).

In this project the overall quality of Wikipedia are relevant for the quality of the database. The higher the quality of Wikipedia the better recommendation can the search engine produce, according to the article from Guisado-Gamez (Guisado-Gámez et al., 2013).

2.5.2 Distribution of article topics

Wikipedia is crowd sourced, meaning everyone and anyone interested in writing or edit an article freely can do so, and the quality control are also done by users. The motivation for users to write on Wikipedia are altruistic, there are no rewards for users writing or editing. Not all topics on Wikipedia have the same amount of articles as the other. What articles are written are dictated by what the users are interested in (Sucheck, Salah, Cheng, & Scharnhorst, 2012).

In 2011 Adam Brown published a study called "Wikipedia as a Data Source for Political Scientists: Accuracy and Completeness". This study examined political articles, and came to the conclusion that coverage are best on topics that are more recent or prominent (Brown, 2011).

This distribution of articles over topics are relevant to this thesis, based on the quality of the knowledge base are directly connected to the quality of the search recommendation produced from the system presented. The studies mentioned are performed on the English version of Wikipedia, but we will use an assumption that the Norwegian Wikipedia has the same distribution, based on the factors that leads to the difference in the English Wikipedia are still relevant on the Norwegian variant. If this assumption is incorrect the results from the search engine will be worse at some topics, but better at some.

2.6 Evaluation methods for Information Retrieval

The traditional method for measuring the effectiveness of information retrieval systems are generally made up of three main parts. The first is a *document collection.* The second is a set of *information needs*, set up as queries. The third part is a set of relevance judgements. The relevance judgements are traditionally an assessment of either *relevant* or *nonrelvant* for each query-document pair (Manning & Raghavan, 2009).

Manning et al. (2009) Argues that for quantitative testing, fifty information needs, *queries*, is a sufficient minimum for evaluating information retrieval applications (Manning & Raghavan, 2009).

The information needs has been relevant to a subject under consideration and appropriate for predicted usage of the application.

Manning et. al. also argues that the key utility measure for any information retrieval system is user happiness (Manning & Raghavan, 2009). User happiness is not only measured by the quality of the result. Some important factors are interface design, quick response, and performance. In this system, the amount of data that is being handled are immense, and with the available hardware it is not possible to achieve the required performance to be able to compete with traditional search engines. Therefore, this thesis will focus on the quality of the results, and helping the users to formulate a sensible query rather than focus on user happiness.

For English language it is possible to automate evaluation of Information retrieval systems. The Text Retrieval Conferences (TREC), a series of conferences that specializes in information retrieval, offers several data sets for testing and benchmarking of information retrieval systems. Many of these data sets contains relevance pairs, that consists of a query and relevance judgements for this query. This makes it possible to judge an IR application automatically (Baeza-Yates & Ribeiro-Neto, 1999). TREC offers some data sets, that contain more than English language, but there are no available data sets for Norwegian language.

2.6.1 Evaluation of query expansion method

The traditional methods for evaluating query expansion are to perform statistical analysis of the retrieval results, with and without the query expansion method enabled. The query expansion method is deemed effective if the results are better with the method enabled than without.

In this project testing will be performed in this manner, and in addition the system will be tested with other query expansions supplied with the Terrier Search engine, to determine if the method is more effective than query expansion methods purely based on statistical analysis of the corpus.

2.6.2 Precision and Recall

For statistical analysis and evaluation of information retrieval applications, precision and recall are simple methods for evaluating the results.
Precision P is the fraction of retrieved documents that are relevant (Manning & Raghavan, 2009).

$$Precision = \frac{\#\text{ of relevant items retrieved}}{\#\text{ of retrieved items}}$$

Recall is the fraction of the relevant documents that are retrieved (Manning & Raghavan, 2009).

$$Recall = \frac{\#\text{ of relevant items retrieved}}{\#\text{ of relevant items}}$$

If a document is retrieved and relevant it is classified as a true positive. If it is not relevant but retrieved the document is a false positive. Reversely if a document is relevant but not retrieved it is a false negative. If it is not relevant and not retrieved then the document is a true positive.

Both precision and recall is set-based methods for evaluation, and treats the result set as an unordered set. Users don't see a result as a set, but rather as a ranked list. It is important that the evaluation of the results are performed in a way that evaluates how the users perceives the results.

2.6.3 Precision at N

Precision at N (P@N) is a method for evaluation of a result as a ranked list. The method calculates the precision on the n-th document in the graph. The disadvantage with this method is that is the least stable of the commonly used methods for evaluation of IR applications (Manning & Raghavan, 2009).

2.6.4 BPREF

In data collections of trivial size it is not a problem establishing the recall values. In larger data collections this can be significant challenge, because all the documents has to be graded as relevant or non relevant against all the different query terms. In larger collections this is a problem because of the time consuming nature of manually matching queries against documents.

To be able to evaluate the rankings of a result *Mean average Precision (MAP)* is one of the most common methods. MAP requires that the relevance judgements are complete. If the Relevance judgements are incomplete, one method for evaluation of the rankings with incomplete relevance judgement is BPREF. The measure is called BPREF because the relevance judgements are binary.

BPREF measures the number of times documents that are known to be non-relevant to an information need, are retrieved before relevant documents. The relevance judgements has to be performed manually by human experts. The BPREF measure is defined as:

$$bpref = \frac{1}{R}\sum 1 - \frac{|\text{ n ranked higher than r }|}{min(R, N)}$$

Where *R* is the number of judged relevant documents. *N* is the number of judged irrelevant documents, *r* is a relevant retrieved document, and *n* is a member of the first *R* irrelevant retrieved documents.

On average BPREF and MAP is very highly correlated when used on data sets with complete relevance judgements, but are constructed for use with incomplete judgements

3 Design Science Research

Research in information systems are concerned with people, organizations and technology (Hevner, March, Park, & Ram, 2004). Researches try to understand problems related to developing and successfully implement information systems in organizations

Development of information systems are often performed to help an organization to increase the efficiency and effectiveness. People, existing systems, development methodologies, and the capabilities of the information system are factors that will affect this process.

Hevner et. Al (2004) argues that there are two different paradigms that characterize this field, behavior science and design science. The first paradigm seeks to develop or justify theories that explain or predict human behavior. The latter is based in engineering and seeks to create innovations that effectively and efficiently solves problems for people and organizations (Hevner et al., 2004). This is the paradigm we will use in this thesis to maintain control over the development, and research results.

3.1 Guidelines for Design Science in Information Systems

Henver et at propose seven guidelines for design science and reasons that:

"The fundamental principle of design-science research from which our seven guidelines are derived is that knowledge and understanding of a design problem and it's solution are acquired in the building and application of an artifact."(Hevner et al., 2004)

The guidelines presented here are used as a framework for the project to maintain the research focus and valid results. After each guideline the usage of the guidelines in this project will be discussed.

Guideline 1: Design as an artifact

"The result from research in information systems is by definition a purposeful IT artifact created to address an important organizational problem. Design science research must produce a viable artifact in the form of a construct, a model, a method or an instantiation. "

The authors of the guidelines define IT artifact as a form of construct, model, a method or an instantiation. This is a broad definition of IT artifact, but in this definition people or elements of organizations are not included, to maintain focus on the innovative technologies element.

In this system the artifact is a prototype search engine, set up with Norwegian language algorithms, and a query expansion system set up using Wikipedia. As mentioned in the research question, the main focus of this project is the query expansion system. More details of the implementation will be discussed int the 4.4 chapter about development

Guideline 2: Problem Relevance

"The objective of design-science research is to develop technology based solutions to important and relevant business problem "

Design science relies on the artifact to construct a method for solving a problem. A problem is defined as "The difference between a goal state and the current state of the system (Hevner et al., 2004).
The general problem domain in this project is searching for Norwegian users. The problem relevance are as stated in the research questions, and will be further explained in the theory chapter, see section 2.3.

Guideline 3: Design Evaluation

"*The utility, quality and efficiency of a design artifact must be rigorously demonstrated via well executed evaluation methods"*

To ensure valid data the performance of the IT artifact has to be evaluated using proper methods. The evaluation of the artifact has to establish the proper metrics, gather data and analyze the appropriate data. The artifact can be evaluated with regard to: functionality, completeness, consistency, accuracy, performance., reliability, usability, fit with the organization or other relevant attributes (Hevner et al., 2004).

The methods for evaluation will be described in detail in the evaluation chapter, chapter 5.
The focus of the evaluation will be performance and accuracy as the search engine evaluations are generally focused on improving results of the search engine.

Guideline 4: Research contribution

"*Effective design science research must provide clear and verifiable contributions in the areas of design artifact, design foundations, and/or design methodologies.*"

This guideline focus on how the research project contributes to the knowledge base. The authors challenge the researchers of new projects to ask themselves: "*What are the new and interesting contributions" (Hevner et al., 2004)*

You can contribute through novelty, generality or significance of the new IT artifact.
In this project the contributions to the knowledge base is the artifact itself, and the results from the evaluations. One of the most interesting factors in this project is if it possible to use the general knowledge extracted from Wikipedia to improve searching for users in Norwegian Language.

Guideline 5: Research Rigor

"Design science research relies upon the application of rigorous methods both in the construction and evaluation of the design artifact."

This guideline is about the way the research is conducted and how the researcher perform the development and evaluation of the artifact. The design science research depends upon the researchers' ability the select the proper techniques when designing the artifact or theory. In addition, the researcher must select the proper methods for evaluate the artifact or theory.
The development process in this project was iterative. The iterative process facilitates frequent error correction, and makes it possible to test new a better solutions if they are discovered.
The theoretical models used for extracting information was based on works from others that was proved efficient in the assigned task, and will be further explained in the chapter about theory, chapter 2.

Guideline 6: Design as a search process

"*The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment. "*

In this context the guideline means that one should explore different implementations and different prototypes to determine the best possible outcome from the artifact, and test them against the

requirements for the project. To achieve this effect I have chosen to use an iterative system development method that allows for multiple iterations of development and testing.

Guideline 7: Communication of research

"Design-science research must be presented effectively both technology-oriented as well as management-oriented audiences. "

The seventh guideline urges researches to communicate the research on different levels of technicality to broaden the target audience and make the thesis understandable for readers.

The primary communication of this research will be this thesis. The thesis will be aimed at an academic audience.

4 Design and Development

In this chapter, the focus is on the development of the system. My aim is to develop a query expansion system for Norwegian language using Wikipedia as a knowledge base. The system is designed to analyze the links between articles in Wikipedia to recommend additional terms that can be added to a search, and hopefully help the user to discover in what domain or context the user information need are in, and reduce the semantic gap between the user's information need and the query that the user are formulating.

By modeling the links between the articles we can estimate what articles are important and most influential. We can use this estimation when we are recommending terms to add to the query and therefore improve the users experience. This method, to estimate relevance between query terms, was successful by Guisado-Gámez (2013), and Døskeland (2014), and the research contribution of this thesis is to determine if this method is viable for Norwegian language, and using the Norwegian knowledge base.

4.1 Prototyping

The system presented in this thesis is not a completely fully implemented system. The system presented is a *prototype* of the system, with focus on the functional requirement for the system. This means that the non-functional requirements that would need to be implemented before the system can be used in *the real world.* Examples of non-functional requirements that needs to be taken care of can be security and performance.

Prototyping can be split into two different kinds of prototypes. Low fidelity or high fidelity. Low fidelity prototypes are prototypes that are cheap and easy to perform, and do not require much investment either financially or time wise. For example, a paper prototype of a computer system is a low fidelity prototype. Low fidelity prototypes are generally limited function, and limited interaction prototyping efforts (Rudd, Stern, & Isensee, 1996).

High-fidelity prototypes are prototypes that requires more in the way in investment and are completely functional, and interactive. High-fidelity prototypes allows the users to experience to look and feel of the product and can be used for marketing and sales of the product. The disadvantages of high fidelity prototypes is the time consuming process of development, and the high financial cost.

The application presented in this thesis is a high fidelity prototype, meant to demonstrate the possibilities from the application. This means that the focus of the prototypes is the functional requirements and very little attention has been paid to the non-functional requirements. Although the non-functional requirements were not the main focus, some consideration had to be made regarding performance. To be able to perform the operations that the system is designed to do, some limitations had to be made. The limitations are further explained in section 4.4.

4.2 Requirements

Before we build an information system, we need to know what it is supposed to do. Many information system development projects have been delayed or failed completely, because the development starts before the involved members know how the system should behave as a finished project.

The solution to this problem is to take time to gather and verify the *software requirements* – Documentation that completely describes the behavior that is required for the software, *before* the system is designed, built and tested. (Stellman, Green, 2005).

To process of determining the requirements vary from development methods to development methods, but in this project rapid application development was used and therefore the natural way to determine requirements was to write *user stories*.

A user story describes functionality that will be valuable to either a user or a purchaser of the system or software. User stories has three components:

- A written description of the story used for planning and as a reminder
- Conversations about the story that serve to flesh out the details of the story
- Tests that convey and document details and that can be used to determine when a story is complete.

This definition is from Mike Cohn's book *User Stories Applied,* published in 2004. One main issue when defining user task is to only write tasks that are applicable to the different user roles. For example: “The system will be written in C++”, is not a good user story because the user don't care about the technical aspects of the system, only about the functionality.

When defining user stories, it is important to care about the different user roles. For a large Enterprise Resource Planning (ERP), it's important to think of the different needs the different users might have. For example a HR consultant might have different needs from the system then an accountant. (Cohn, 2004)

The user stories in this project was described as a top level functionally requirement, and then divided into smaller concrete subtasks. All the subtasks was directly related to a user story that described the functionality. For example; "*As a user I would like to be able to search for keywords".* This user story will be broken down into different *concrete* tasks. For example: "Create an index that supports Norwegian language".

This technique makes the stack from a high levels user story to a specific task very readable and before starting the development it is possible to predict the behavior of the system.

Functional requirements are requirements that are a requirement for a behavior for the system. As previously mentioned user stories was being used for determining the functional requirements of this system.

The usage of the functional requirements for the development of this system will be further explained in section 4.3.

Users will often have *implicit* requirements from a system. This will often be requirements regarding usability, how easy it is to use the system, performance and security. It is very important for the users that when they press a button, something happens, and they don't have to wait for several minutes.

4.3 Development Method

Development of information systems are notoriously difficult to predict and control. To be able to exercise control over the development progress it's common to use a development method.

In the development system the choice for development method fell on *Rapid Application Development (RAD)*, with elements from the Scrum development methodology. The iterative nature of this

development method is important to keep the project within the guidelines presented in the design science research.

RAD is centered around minimal planning, but rapid prototyping. After each sprint a prototype is developed. The elements borrowed from Scrum are the short planning phase in the start of the sprint and the retrospective consideration of the sprint after the sprint is performed. In accordance with RAD, in the retrospective, the prototype is evaluated. The evaluation is centered around if the prototype is performing as intended.

Critics of the RAD development method argues that the method makes the planning of larger development processes difficult to control, and argues that communication between stakeholders is key to succeeding in a RAD development process (Gerber, Merwe, & Alberts, 2007).

In the planning phase the key objective is to analyze what needs to be done, and what is critical to get done. The retrospective allows for identification of mistakes and failures for further improvement of the development. The constant building of a backlog of tasks allows for iterative adding of tasks and allows for testing of new ideas, to improve the final result.

4.3.1 Physical Artifacts in the Development Process

In this project a scrum board was used to keep track of the different tasks. Each task consists of a short description of what needs to be done, with an estimation of how time consuming the task is. The backlog consisted of tasks in prioritized order based on importance of task. The physical artifact of a scrum board helps with keeping track of progress, while maintaining control over progress. The scrum board consists of a white board with four different columns. *To do, In progress, Review,* and *done.* For each sprint, all the columns were updated to make sure the board was easily readable.

The review column was before the final acceptance test was performed and the task moved to done. For each sprint all the tasks were taken down, and new tasks was defined from the prioritized backlog.

4.4 Query Expansion implementation

The system presented in this thesis is as previously mentioned an extension for the Terrier search engine. In this part the additions to the search engine will be presented. The necessary components for

evaluation of information retrieval application will also be presented in this chapter, as they are essential to maintain the iterative nature of the research, and they are critical for evaluation of the proposed system.

4.4.1 Corpus

Corpus or the document collection are the total collection of documents to perform the retrieval on (Manning & Raghavan, 2009). This collection is referred to as *Corpus (*body of text). To be able to perform retrieval operations a corpus is required for testing and evaluation

There is no universal *minimal* size requirement for information retrieval applications but the collection needs to be large enough to be able to represent the users different information needs.

The document collection in this project will be divided into two different parts. The first part is a collection of documents to search in. This collection has to be in Norwegian language exclusively for the user to search in them. For it to be possible it also needs to cover as many different topics as possible, to effectively reflect a user's information need.

The corpus implemented and tested for this project is a corpus of approximately 200,000 newspaper articles made publicly available in 2012 and 2013. This corpus was downloaded from the *Norwegian National Library.* The access to this corpus was granted by *avis.uib.no*

This part of the corpus is indexed as a normal index by Terrier the search engine, with the exception of the stemming algorithm. The stemming algorithm used was a Norwegian Snowball Stemmer, as mentioned in section 2.3.3 Stemming was chosen over the more correct, but computationally expensive lemmatization process.

The second part of the corpus is a hidden indexed version of Wikipedia corpus. When the user performs a search, the engine first performs a hidden search against the Wikipedia corpus. This results in a starting node in the Neo4j graph database, which allows us to perform estimations on what nodes to include in the search. The reasoning behind this hidden index is to as fast as possible determine a viable and high quality starting node for the system to start from without guessing what usage of the query the user is after.

4.4.2 Creation of the knowledge base.

To be able to perform intelligent recommendations for query expansion we have to have a knowledge base. In my thesis I have chosen to use the Norwegian Wikipedia as a knowledge base. To be able to extract the information from Wikipedia to a graph database I downloaded an XML dump of Wikipedia, and used a heavily modified version of a program called Graphipedia to translate this into a format readable for Neo4j. Graphipedia will be further presented in section 4.5.2.
This setup allowed me to use Neo4js own query language to query the graph directly.

The size of the knowledge base is massive. The graph has 850,000 Nodes with over 8 Million links between the nodes. In the process of extracting sensible data from the knowledge base I tried various approaches. The final method for extracting data from the graph will be presented in section 4.4.4.

4.4.3 Construction of index

To create the index of the Norwegian newspaper articles, the Terrier index class called *SimpleXMLCollection* was used. This class creates an index over xml files with the tags supplied. All the documents in the document collection are delimited by xml tags called *Document.*

For this indexing process Terrier was configured to use a tokenizer with support for UTF-8 encoding of words, so that the engine could support usage of the Norwegian letters, æ, ø, å, as mentioned in section 2.3.1 and section 2.3.4. This made sure that the engine did not exclude words containing the letters and maintain support for Norwegian language.

The Terrier built in indexer system includes an automatic file parsing system that detects and analyzes the documents to read, and then selects an appropriate indexer. In this case the documents are in XML format, so the system automatically selects the *simpleXMLparser* class to index the data. This process guaranties that all the files in the corpus is parsed correctly. If the system tries to read a file and cannot determine the file type or cannot read the document properly, then the system casts an exception and notifies the user.

As previously mentioned in section 2.3.2, the stop words are removed from the documents using a list of stop word gathered from the web page of tartarus.org (Bruusgaard, 2005).

The methods presented here created an index that allowed for retrieval task using Norwegian language, using methods designed for optimizing the results for the language.

4.4.4 Retrieval

NeoExpansion is the starting method for the query expansion. The method is designed to take a word as input and returns a ArrayList of pages that contains results relevant to the initial word. This method is being run first for the whole query as a sentence, then on the individual words. By doing this we first try to determine what the whole query is about, and then we try the individual words.

When are word or a sentence is fed into the expansion method, the first thing that happens is that the system tries to match the word to a Wikipedia article, by performing a matching operation with the word to a hidden index of Wikipedia, using the Okapi BM25 method.

Okapi BM25 is a ranking function for information retrieval, based on probabilistic retrieval framework (Baeza-Yates & Ribeiro-Neto, 1999).

This results in a list of 10 different Wikipedia articles that are relevant to the word. The number ten is an arbitrary number, but it reflects the top ten matching articles from an IR perspective.

From the list of ten different articles, we perform a method *importantNeighbour. These* methods connects to the Neo4j graph database and returns the most important neighbor of the starting node. We determine the importance of the nodes using the Pagerank algorithm.

If this node is over a threshold, it is added into the queue that is to be presented to the user, where the user can try to determine in what context the query formulation is used. The Threshold for the neighbor node, is if the Pagerank value are lower than the Pagerank value of the initial node times 20. The purpose of this limitation is to stop nodes regarding articles that gets an unnatural high amount of links to it. For example, articles regarding large nations. This method for skewing the weighting of nodes became visible during the optimization phase of the development, see section 4.7.

The different threshold levels, and the limiting values were evaluated during the development. The values are arbitrary numbers, but the results are improved as a result, and the guidelines presented in section 3.1, supports this iterative process of method of research.

In addition, the method takes the starting nodes and performs a community detection algorithm. The algorithm is presented in section 2.4.3. This algorithm returns a set of articles that naturally belongs together from a graph theory view. The nodes retrieved in this set is added to a total result list.

From the result list, the top fifteen results based on the Pagerank algorithm, explained in section 2.4.1, are presented to the user.

The user can select one or more of the suggestions presented. Some query expansion models, for example *Divergence from randomness,* will not present the selected expansion terms to the user, but rather just include them in the search. The selection of terms allows for users to find related terms and refine the scope of the search before it is performed. The reason for making the user choose is to support the different searching strategies, as mentioned in section 2.1.3.

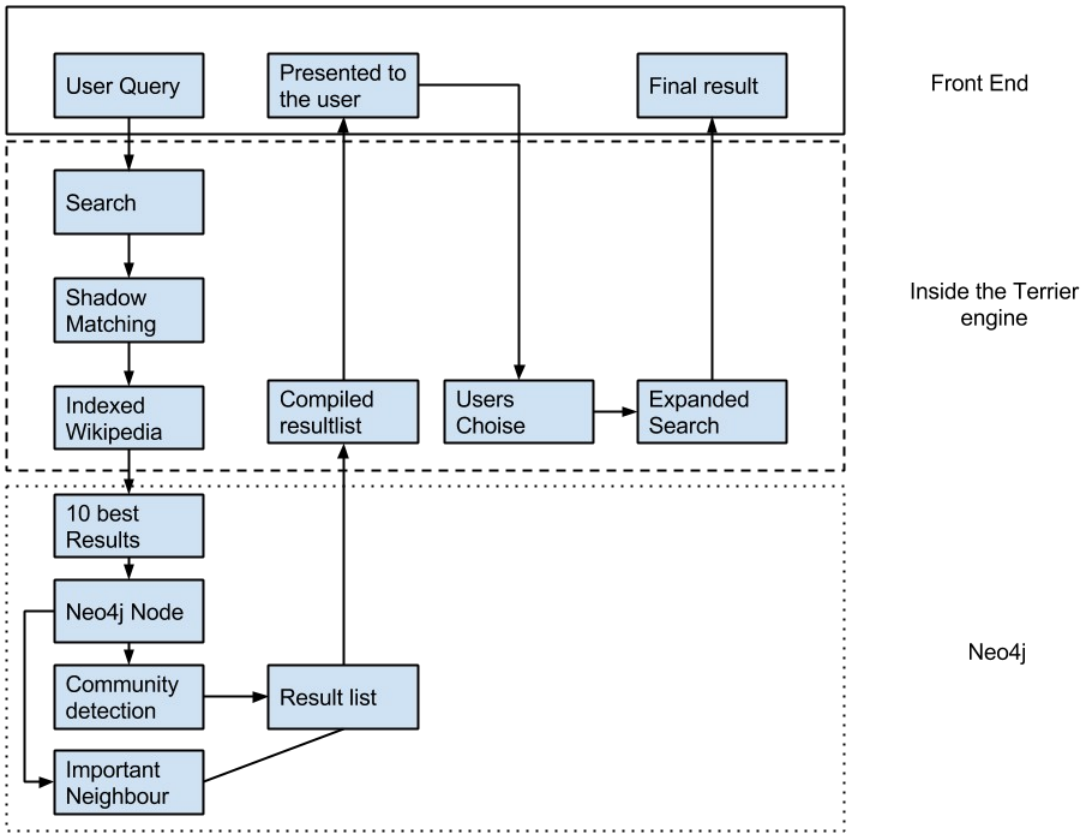

Illustration 3: Overview of data flow

4.4.5 Community Detection Algorithm

The algorithm mentioned in section 2.4.3 has been implemented in this system. The Java implementation is written by the authors of the algorithm. In this implementation the input for the algorithm is the starting node that best matches the query from a lexical retrieval. From this starting node, the system uses the Neo4j graph database to extract the neighboring nodes. The starting node and

the adjacent nodes are then inserted into a Java Jung graph iteratively, up until the Jung Graph contains 1000 different nodes. This limiting value of 1000 nodes is added due to memory constraints.

Java Jung is a universal Network/Graph framework, for traversal and calculation of graph properties. (Joshua O ' Madadhain, Fisher, Nelson, White, & Boey, 2005) The Jung graph is then inserted to the algorithm. The result set is the sub-graph with locally optimized modularity.

This community is a set of nodes that naturally are related to each other based on the graph property (Tang & Liu, 2010).The community detection algorithm returns a set of nodes that are estimated to be relevant for the original query.

This system is as mentioned based on a graph of Wikipedia. Wikipedia is under constant evolvement, with users updating, and adding articles constantly. By using a seed-based community detection the algorithm can quickly respond to changes in the structure of Wikipedia, without needing to calculate the communities for the graph as a whole every time the structure of Wikipedia changes.

The suggestion of words presented to the user is based upon estimations. If the estimations leading up to the suggestions of words are incorrect, the user can still choose to ignore the suggestion. This is to support graceful degrade.

4.5 Tools

In this part I will describe the different tools I used for the development of the system. Both the hardware and the software that was used.

4.5.1 Hardware

The indexing process is a computationally expensive process. The first tests of the indexing process is done on a laptop with mediocre specifications. The tests will be repeated on a computer with better specification.

Processor: Intel i5 CPU M 520 2,4GHz
Memory: 4 GB
Hard drive; 256 GB memory upgraded during the development process to 128 GB SSD.

Computer Specification of desktop PC
Processor: Intel i7 3770k 3,7GHz
Memory: 16gb DDR3
Hard drive: 128 GB SSD + 1 TB Had

Several of the operations performed in this project was so memory intensive, that the operations had to be performed on the desktop.
The system can run without using too much computational power but the indexing of the corpus and knowledge-base are so computational expensive that more processing power are required to perform the operations.

The operations needed to create this project was mainly memory intensive. The choke points for the graph calculations and the indexing was memory constraints.

4.5.2 Software

In this section I will present the software used in the development of my system. All the software used are within licensing for free use.

Terrier Search Engine

The Terrier search engine, Terrabyte retriever, is a project that was initiated at the University of Glasgow in 2000. The aim of the project is to provide a publicly available test-bed for rapid development of IR applications. (Ounis et al, 2006)

This search engine is a highly modular and scalable search engine that ships with a selection of modern and common statistical retrieval models such as; TF-IDF, BM25, and Language modeling (Iadh Ounis, Vassilis Plachouras, Ben He, Craigh Macdonald, 2006).

This Open source search engine suits this project based on the modular design and implementation of a wide variety of retrieval model. The indexing system included in Terrier allows very fast indexing of large amounts of data, without sacrificing the customizability. A comparison of open source search engine shows that the Terrier search engine is on par with the other open source search engine

(Middleton & Baeza-Yates, 2007). Terrier is Java based, and based on my previous programming experience Terrier was the engine best suited for this project. The downside with Terrier compared to the other search engines, is the linear growth in memory usage.

Java

The Terrier search engine is Java based, and the result of this project is an extension to this engine. Therefore, this project is based on the programming language *Java.* I have chosen this because this is the programming language that I am most familiar with, and the extensive libraries and documentation available.

Ubuntu

This project was developed using Ubuntu as operating system. Ubuntu is a Linux based operating system. I have chosen Ubuntu based on availability and the other used tools are easier to control with the Unix based command line available in Linux.

NoSQL databases

NoSQL is a term used to describe a collection of database technologies that were developed from the rise of Web 2.0 applications. NoSQL data stores are designed to scale simple OLTP style application loads over many servers. The systems are designed to handle millions of users performing updates and reads, in contrast to traditional database systems. (Cattell, 2001)

Graph Database

“A Graph is a collection of vertices and egdes, or nodes and the relationships connecting the nodes. Graphs represents entities as nodes and the ways in which those entities relate to the world as relationships. This general-purpose, expressive structure allows us to model all kinds of different scenarios, from the construction of a space rocket, to a system of roads, to the supply-chain or provenance foodstuff, to medical history for populations and beyond. “ (Robinson, Webber, & Eifrem, 2013)

I have chosen to use a graph database to utilize the transactional capabilities of the database to extract data from the link structure.

Neo4j

A Graph database management system is an online database management system with create, read, update and delete methods (CRUD) that exposes a graph data model. (Robinson et al., 2013) The graph databases are often optimized for transactional performance, and engineered with transactional integrity and operational availability in mind.

Robinson argues that there are two properties that are important to determine when investigating graph database technologies:

The underlying storage

Some graph databases uses *native graph storage* that is optimized and designed for storing and managing graphs. Not all graph database technologies use native graph storage. Some serialize the graph into a relational database, an object oriented database or some other general purpose data store.

The processing engine

Some definition requires that the graph database use *index-free adjacency,* meaning that connected nodes physically "point" to each other in the database. Robinson argues that the processing engine are a graph database if the database *behaves* like a graph database from a user perspective. (Robinson et al., 2013)

To store my knowledge extracted from Wikipedia, I have chosen to store the data in a graph database, with the articles as nodes and hyperlinks between the articles as vertices. This allowed us to create a highly scalable database that can store the knowledge. In a way that represents the extracted information correctly.

I have chosen Neo4j as the graph database for this project. There exists many graph databases, but Neo4j's high scalability and transactional capabilities
Neo4j satisfies Robinsons definition of a graph engine with both native graph storage and native graph processing engine.
The Neo4j Querying language *Cypher* allows for very fast transactional queries, adding further support for the project's engine. In addition, Neo4j in combination with the graph analytics tool Mazerunner

allows for graph calculations without holding the graph in memory as a whole. Mazerunner will be presented in this chapter. Neo4j is a NoSQL graph database.

Graphipedia

The operation of inserting the Wikipedia dump in xml format to a Neo4j graph database is an extremely computationally expensive operation. There are ways to avoid this issue, the most common is to perform a batch insertion operation, that circumvents the built in transactional safeties. The transactional limitations are a safety feature against corruption of the database if something goes wrong when performing database operations. To perform a batch insertion into Neo4j I used a customized version of a program called Graphipedia. Graphipeda converts xml dumps of Wikipedia into neo4j databases, and removes pages that are not articles. For example redirects, portals, categories. In this project we need as much data as possible, so the customized version do not remove these links, as they represent a connection that we can analyze. Graphipedia is created by Mirko Nasato (Nasato, 2015).

Mazerunner

To be able to efficiently calculate the Pagerank value for the nodes in my Neo4j Graph i have used a graph analytics tool called Mazerunner. Mazerunner was created by Kenny Bastani, and is still in its beta. This tool uses Apache Spark and GraphX to perform ETL graph analysis on graphs and subgraphs exported from Neo4j, and stores the properties as properties on the individual nodes in the Neo4j graph. This means that the data supplied from Mazerunner is easy to query on using Neo4j's query language, *Cypher*.

The system also facilitates Triangle Counting, Connected Components and Strongly connected components, for further analysis of graphs (Bastani, 2015).

The Mazerunner tool is licensed under the Apache License. The license allowed the user to use modify distribute without concern for royalties.

4.6 Graphical User Interface

This project is aimed at improving the retrieval method, and the focus on graphical user interface (GUI) is minimal. For testing purposes the built-in desktop application from the Terrier search engine was used to construct a method for presenting the relevant term to the user.

The main window is a box with a search field, output logs and return field. When the user types in a query in the window a performs the retrieval process, the result field gets populated by a ranked list with the best matching results.

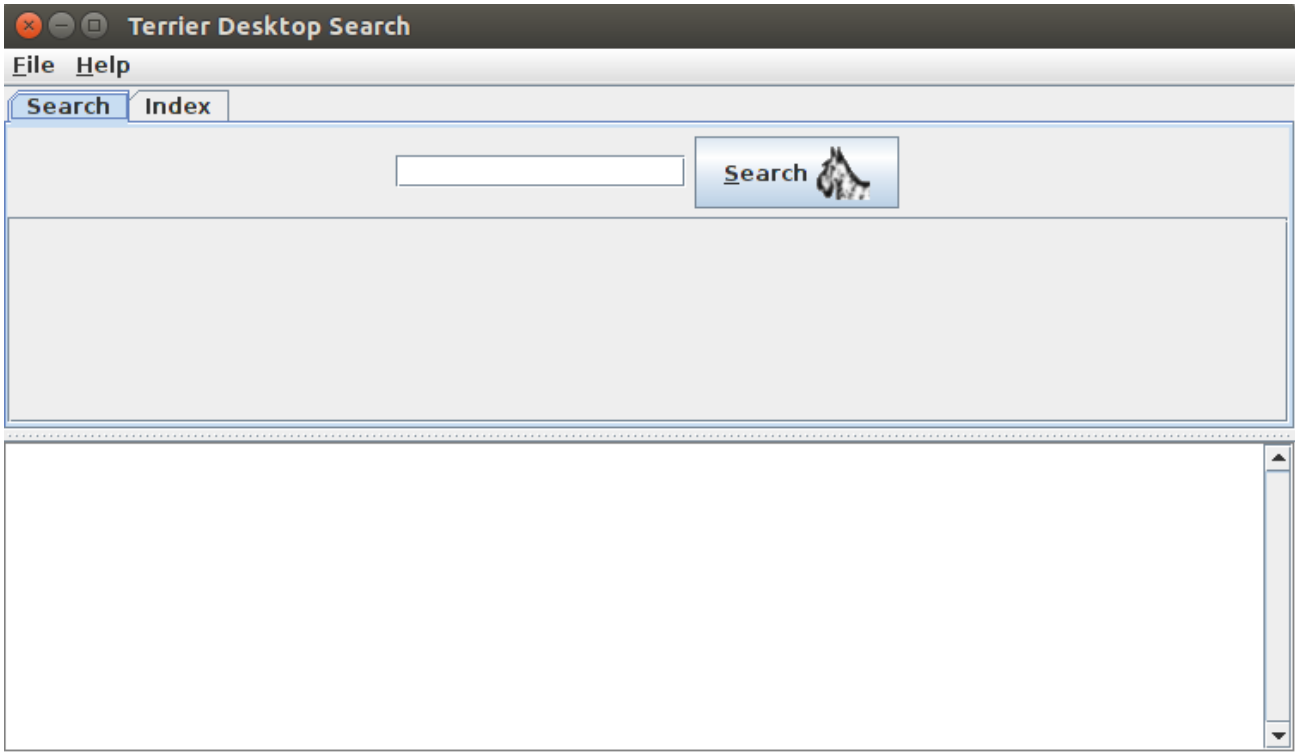

Illustration 4: Graphical User Interface

When the user perform a query a pop-up box will appear where the user can select what terms to add to the query.

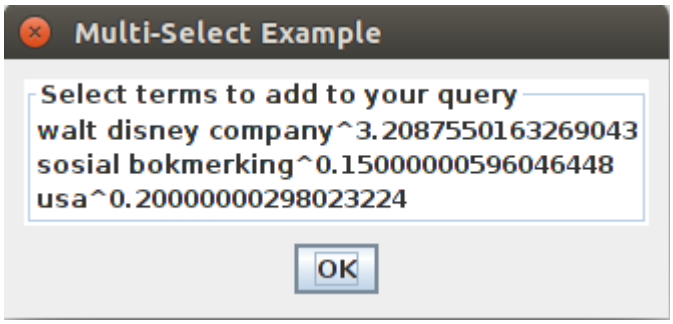

Illustration 5: Graphical User Interface Suggestions

The GUI presented here is sufficient for testing the retrieval process, but for implementation and usage for users a complete GUI, preferably a web-server will have to be developed. This will be further discussed in section 6.2, about future work.

4.7 Iterations

In this part, the different iterations during the development process will be presented. During development process of this project multiple approaches and methods were attempted. The testing and failing of different methods are in line with the design science research guidelines discussed in section 3.1.

4.7.1 First

The first iteration consisted of setup of the computer environment. This included installation of the software for development, documentation and backup solutions. In this iteration I started gathering the data for the search engine. I started by creating a crawler designed to find documents in Norwegian and download sufficient amount of documents to perform retrieval tasks. The main challenge was creating a large enough collection of documents consisting exclusively of documents written in Norwegian. This task was aborted when I got granted access to the current document collection. In the start of the project the plan was to set up this system as a web server, completely or partially, so some effort was put into researching the possibilities for hosting of Neo4j and Java applications. This idea was scrapped after determining the cost of maintaining or renting such as server.

4.7.2 Second

In the second iteration the focus was getting The terrier search engine up and running and customized for Norwegian language. This includes Norwegian stemmer, and finding Norwegian stop words. This process included setup and testing of both indexing and retrieving of Norwegian documents.

4.7.3 Third

In this phase I encountered some major difficulties, and the progress of the project halted massively. The issues were to determine a way to define the measurement of importance, in a way that both represent the importance of a node, and can be performed within reasonable time, in other words, so fast that the user will not notice it.
To determine this I tried many methods of measuring the performance. I used the nodes extracted from the Neo4j database and tried to determine the important nodes from this data. Most of *network-centric* methods of calculations can be in the worst case scenario be O(n3). Even with the reduced dataset extracted from the Neo4J graph this was far to ineffective.
This lead me to the conclusion that the importance measurement needs to be preprocessed and stored in the graph as fields on each article. This will allow the program to effectively and quickly find the importance of each nodes.

This iteration gained momentum when I was recommended the Mazerunner system. This system allows for Pagerank calculation on the graph without holding the graph as a whole in the computer memory.

4.7.4 Fourth

In the fourth iteration, the goal was to extend the Terrier search engine. This proved to be a much more complicated task than anticipated. The process of extending and determine the necessary components has been very time consuming. This process requires knowledge of Ant, and has proven to be difficult to extend the search engine and then build it correctly to include the added functionality. This has proven to be a more difficult task than expected.

After working with the node centrality algorithms in third iteration, it became clear that the main problem to deal with, was to split the graph in to smaller, more manageable subgraphs that naturally belong together. This thought leaded the process towards the implementation of the community detection algorithm mentioned in section 2.4.3. The objective for a community detection algorithm is to split a graph based on nodes that naturally belongs together, so community detection was the most efficient method of making the graph more manageable.

4.7.5 Fifth

When the basic understanding and setup of Terrier was completed I started on extending the search engine. This included removal of stop words, Norwegian stemming algorithm and a separate hidden index. This hidden index is an indexed version of the Norwegian Wikipedia. This hidden index allows performs a hidden search against Wikipedia, and returns the nodes with the best match in the Wikipedia graph. This node is a starting point for the link analysis algorithms. Different method for calculating importance was tried and tested with and without success.

4.7.6 Sixth – Optimization phase

In this iteration the focus was on optimizing the search engine. This meant testing and optimizing the threshold values for when a node is added as a result, and should be presented to the user. In start of this process, I had a suspicion that the community detection algorithm returned words that was strictly worse than using only the most important neighbor, calculated by Pagerank as mentioned in section 2.4.1. To verify or disprove this theory, both the different algorithm was tested, and this resulted in being false, and no further testing was performed with only the closest neighbor.

In the early part of this evaluation phase it became obvious that articles regarding years and dates was getting abnormally high importance scores, while there are very few information needs regarding them. So a batch operation that removed them from the graph was performed.

The same pattern was observed regarding geographical entities, as they are very easy to create links to. Almost all user terms, resulted in a suggestion for a geographical entity. The solution for this problem was to limit the upper Pagerank values of the recommended nodes, as the articles regarding large geographical entities often holds massive values. Users will rarely search for years and dates, but geographical entities are often subject for searching operations. This means that it's not a viable solution to remove the articles from the database. The solution was to limit the maximum value of the candidate node, to the value of the starting node times 20. This ensures that the node selected are relevant but not subjected to the rich get richer effect that the network displays.

This is meant as an example of the evaluations and considerations performed in this iteration. The threshold numbers are arbitrary numbers, but reflects the results with the highest quality.

5 Evaluation

In this chapter the process for evaluation and the results from the evaluation will be presented. The goal of the evaluation is to gather data and use this data to answer the research questions introduced in chapter 1.

5.1 Evaluation During Development

During the development cycle of this project we worked as previously mentioned on specific tasks derived directly from user stories. One of the main advantages of this development method is that it requires the task to pass an *acceptance test*, in order to be declared as finished as mentioned in section 4.3. This leads to a constant testing of results to make sure the system actually performs what it is designed to do.

In the last development iteration, called the development phase, the focus was on optimizing the set of words presented to the users. This was an early evaluation of the results. The first few rounds of evaluation was an intuitive reason. By changing the different threshold levels, it is possible to intuitively evaluate the results that the system recommends to the user.

For example, articles regarding years was very highly rated. This happens because there are a lot of links to articles regarding years. These articles are very unlikely to be relevant in a search, therefore they were removed completely from the database. The processes described here are performed to maintain the iterative searching nature of the research process described in section 3.1.

5.2 Quantitative evaluation

5.2.1 Goal

The goal for this experiment is to determine if the search engines results are of higher quality than other engines and specifically with regard to the query expansion method in Norwegian. This is because

there are no providers of this service in Norwegian, and the research contributions of this thesis is to evaluate if it is viable in Norwegian language.

The evaluations perform the standard method for testing query expansion, and comparing the results from one run with community detection enabled, one run without community detection, and one run without query expansion enabled. The goal for this test is to determine if the query expansion makes the overall results better or worse.

As mentioned is section 2.6 there are no automated test methods for information retrieval in Norwegian, so the evaluation of the search engine are performed manually, by checking the relevance of a query towards the results.

5.2.2 Procedure

As mentioned in section 2.6 evaluation of information retrieval applications requires a set of terms for evaluation, a document collection and a set of relevance judgement. In this part the different components will be presented, and the metrics for evaluation of the system.

As mentioned in section 4.4.1, the corpus used for this search engine is a collection of news articles, published in Norway in 2012 and 2013. This means that the retrieval results will be focused about topics and events that were discussed in the media, and the corpus cannot be considered a general purpose corpus, and will not reflect all possible information need, as a large search engine for the web.

As discussed in section 2.6, relevance judgements are essential in the evaluation of information retrieval applications. In this evaluation, the articles that are focused on the query term are relevant. This means that if the word is mentioned in an article, but not central to the text itself, then the document will be regarded as nonrelevant. All the relevance judgements are performed manually.

The topic distribution in the knowledge base is also affecting the quality of the results. As mentioned in section 2.5.2 about Wikipedia topic distribution. This is because articles regarding recent topics are more updated and better linked from one article to another. This is directly affecting the database, so

that queries concerning new topics will result in better suggestions from the query expansion system, than subjects regarding more dated topic.

Terms

The queries are designed to test different ways a formulation can be formulated. For example finding specializations and generalizations, finding syn sets of words and determining the context of an ambiguous word.

As mentioned in section 2.6 evaluation of Information retrieval requires a minimum of queries to achieve valid results. For evaluation of this engine, fifty different information needs have been developed, and will be tested with and without the query expansion method enabled. See appendix A.

The terms selected for the evaluation is a set compiled from different sources. Some of the terms were collected from Døskelands thesis (2012). As mentioned in section 2.6, the terms for evaluation should be germane to the documents in the test collection.

To adapt the terms to be relevant to the data set, some other terms are used to focus the testing towards topics that have been covered in the media the last years to get enough data to compare the results from older topics with newer topics.

The different information needs have also been divided into *topics.* The topics are information needs that are regarding the same overall type of entities. For example; geographical entities, spatial events, theoretical concepts. The reasoning for this is the possibility to analyze where the engine is better or worse.

The set of queries are designed to reflect the different possible search strategies that the user might use in the formulation of the query. As previously mentioned in section 2.1.3 there are many strategies and the evaluation needs to reflect the different methods. The queries are designed to test different ways a formulation can be formulated. For example finding specializations and generalizations, finding synset of words and determining the context of an ambiguous word.

Data gathering procedure

The evaluation was performed by matching the queries to the test set, and comparing the results of the first 20 results against the query and declaring them relevant or not relevant. This method was performed four times. The first time with community detection algorithm enabled, the second with only the closest neighbor added, and no community detection, and the third time without query expansion

enabled. The fourth time the retrieval process was performed without any Norwegian algorithms enabled. In the fourth evaluation the language specific algorithms was the standard terrier values, that are customized for English language. The only difference is the tokenizer, as Norwegian language requires UTF-8 encoding, to be able to represent language specific characters.

5.2.3 Data Analysis

To analyze the data gathered, the traditional methods for evaluation of query expansion methods were used, as mentioned in section 2.6.

For all the queries' precision at n, were calculated, as explained in section 2.6.2 The precision were calculated for the 5, 10 and 20 level of articles as mentioned in section 2.6.3. To be able to determine changes within the different topics, the changes for the topics were calculated.

The BPREF evaluation method presented in section 2.6.4 was applied to the first ten results, both with the Query expansion method enabled and with no Query expansion.

5.2.4 Results

Using the community detection algorithm, the system shows a 2.4 percent improvement at the precision at the tenth article, and 3.6 percent improvement at article number twenty, compared to using no query expansion, using precision at N, see section 2.6.3.
Without the community detection algorithm, the precision at the tenth article is of 1% higher quality, and 1.4 percent improvement at 20. As previously mentioned the queries were assigned a *topic,* that reflected what the query is about. This list is not exclusive, and one query can be assigned to one or more topics.

With query expansion the results whereas follows:

Topic	Difference in P@10	Difference in P@20
Organization	- 14.3 %	- 4.2 %
Concept	1,6 %	7.5 %
Geography	5.7 %	8.5 %
Generalization	0	- 2.5 %
Persons	3.7 %	0
Abbreviations	0	1.6 %
Specializations	3.3%	13.3%
Franchise	5 %	3.7 %
Physical entity	13 %	6.6 %
Incident	5%	2.5 %
Total	2.4%	3.6%

Table 2: Precision results

The percentage in difference is the difference between the results from using query expansion, and the results without query expansion. The negative values means that the results were better when using no query expansion compared to using it.

Overall the query results improved 2.6% compared to no query expansion, using the query expansion method with community detection enabled as presented in section 4.4.

The topic *organization* results declined with 14.2 % at the tenth article. The reason for this decline in quality is that the term selected from the query expansion method is incorrectly read by the stemming engine and thus is matched incorrectly with the index.

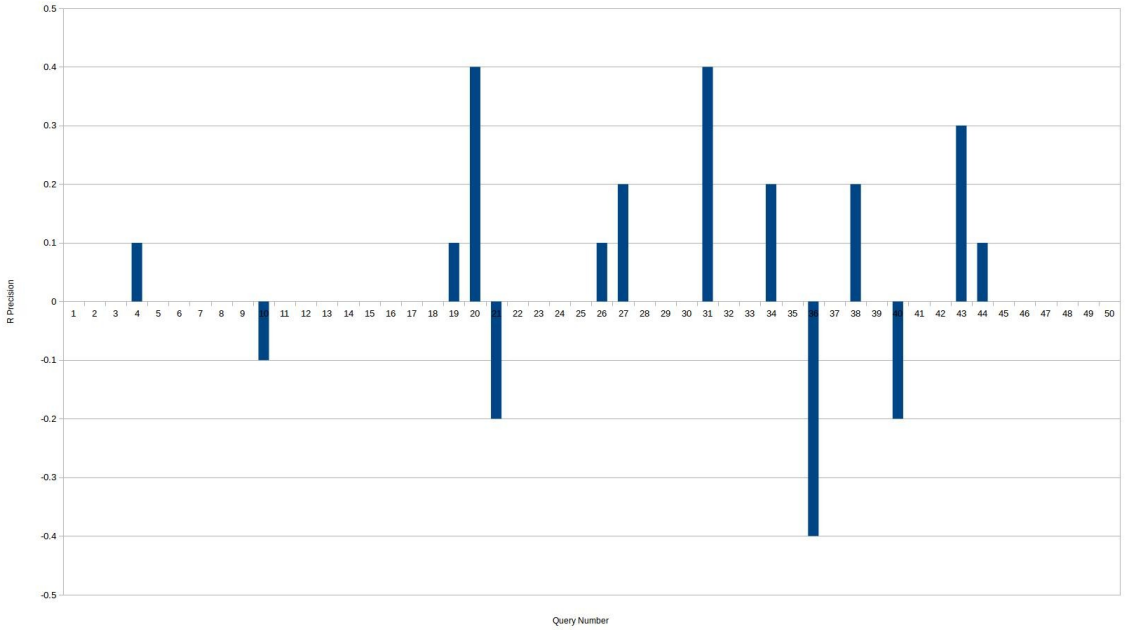

Illustration 6: Precision Histogram at 20

The histogram shows the difference between the results when using the query expansion algorithm and without using query expansion. The numbers are positive when the query expansion method is better, and negative when the query expansion method is worse. The complete results from precision can be found in appendix G.

The BPREF method for evaluation of the results shows no change in the quality of the rankings, with only 0.004 change in the BPREF results. The BPREF results can be found in appendix H and I.

Without Norwegian algorithms the engine shows a 13.1% decrease in the quality of the results. This result is interesting as there very limited options for users that wish to search in Norwegian. If every retrieval operation performed in the users native tongue is 13 % percent worse than retrieval operations performed in English, this may affect the users experience, and it can force the user to reformulate the query multiple times to cover the information need. The precision for Norwegian terms in an English search engine can be found in appendix J.

Naturally queries regarding names and organizations are not affected by the lack of language algorithms, but the language specific queries shows an increase in results.

5.3 Qualitative Evaluation

5.3.1 Goal

In this part the suggestions of new terms to add to the query will be evaluated. The suggestion of the query terms is an essential step in helping the users formulate a query. The goal for this evaluation is to determine if the suggested words can relate to the possible information needs from a user generated query.

The goal of this evaluation is to determine if the words suggested from the system is related to the query term formulated by the user. If the suggested word is related it would make the user to think about the query formulated and add the relevant suggested words to the query.

5.3.2 Procedure

To evaluate if the suggestions covers the different information needs, the process starts by gathering terms to test.

To gather the data for the experiment, interview of experts were conducted.

The definition of experts are unclear and discussed. In this thesis the definition of an expert is a person with domain knowledge.

This method for data gathering is an online form of interview. Though this method is debated, but based on the availability of subjects with the required knowledge this method provides a very efficient method for data gathering (Oates, 2006).

The interviews can be found In appendices F through J, and are written in Norwegian. The interviews themselves was conducted in Norwegian, to make sure that no information was lost in the translation, and as the subject of the interviews was in Norwegian.

Six experts from different disciplines were asked to come up with ten terms from their respective disciplines. The set of words from the experts was then tested using the system, and the suggestions from the application were collected. Then the suggestions from the system was sent back to the expert. The experts then commented on the set as a whole and each term at a time, to evaluate if the words from the suggestion were related to the original term, and how it can help users formulate queries.

The interviews of the experts was conducted over email. First, the experts were asked if they wanted to participate. Some of the experts was curious about other words in their respective disciplines and the terms suggested by the experts was also included. The interview method used was a structured interview.

5.3.3 Results

Law

The law experts comments that many of the suggestions are related concepts regarding the same law, or same theory. For example in one of the terms, *lex superior*, one of the recommended terms are lex posterior. The experts notes that lex posterior is one of the other laws regarding the same branch of principles.

The experts also notes that one of the recommended terms are not related to the term, from a legal point of view. For example the term culpa, some of the recommended terms are psykedelisk musikk("Psychedelic music"), which may not be relevant in this sense of the word.

Some suggestions return a person's name. In the example of *Avsavnsrente (A legal form of economic interest*), one of the suggested terms is *Viggo Hagstrøm.* The connection between the terms may not obvious to the naked eye, but the person was a professor of law, and wrote academic works on the subject, and can subsequently be interpreted as relevant as a suggestion. For example if a user has information needs in regard to a textbook on the subject this would be a relevant addition to the users query.

Four of the expert generated queries did not return in any result. This is based directly on the quality of the knowledge base, and if there are no entries regarding the subjects then there are no starting point for the system.

For example the word *rettshåndhevelsesarrest (*A form of arrest, legal if there are concerns for safety and justice). It is a legal term, but there are no Wikipedia entries on the subject. This is an example of lack of content in the knowledge base.

Another term *Presumpsjonsprinsippet(A legal form of presumption)* is a term used within the discipline but is not an actual law term.

The experts notes that only two of the results are completely unrelated.

Economics

The economics experts comments that there are quite a few terms that does not match the indented information need. For example the query *Taylor-regelen(Taylor rule).* None of the returned terms was considered relevant. This is because the tokenizer removes any special character, in this case the hyphen, and replaces it with a dash. This leads to two different words, one that is a common name, and one very wide. Even further this leads to suggestion terms that have little or no relevance to the original query term.

The same effect in the query *adaptive forventninger (*adaptive expectations). It splits a theoretical concepts into to nonrelevant terms, and the meaning gets lost in the tokenizer. Concepts with less common names, such as *likevekt* (equilibrium) returns relevant queries.

The query *Balansert vekstbane (steady state)* results in some concepts are relevant, and some of the concepts are not. This query suffers from the same as the previously mentioned queries, were the multi-word query gets split up into two separate terms, and sub sequentially leads to worse results.

A few of the concepts are very broad and general. The term *samfunnsøkonomi (social economics)* is a broad subject, but the search engine returns no result from this search, and this problem impacts the results of the application. *Finans (Finance), Budsjettering (Budgeting),* are both very broad and general terms with results of varying quality. A variety of the results are relevant to the discipline and some of the results are not relevant at all.

The user query *monopol (monopoly)* returns results that are relevant both from the discipline and the board game with the same name. This result is positive as the user can separate the information need from ambiguous terms, and formulate a more precise query. The experts comment's that the broad terms returned no results or non-relevant results contrary to what was excepted.

Archeology

The archeology experts comments that the results are mostly relevant to the original query term. Some of the concepts are related, for example *terminus, ante quem* and *post quem.* The term *ante quem* returns *post quem* and vice versa. The cross suggestion of these terms is positive and shows that the terms are related and the engines' ability to identify related terms. Some of the terms are broader terms, that the experts notes that more related terms could be returned.

Bipolar (kjerne) and *Korteks (Cortex)* are shared terms between different disciplines and the other disciplines dominate the result.

Overall the expert note that the relatedness of the suggested terms are reasonable.

The experts notes that for the majority of the terms the recommendations are relevant, but not complete. The suggestions are often related concepts, but the experts noted that the expected terms for suggestions often is generalizations or higher level topics regarding a word. As noted in section 2.4.2, the suggestion is based on an estimation of the most important terms from a mathematical view.

The lack of relevant suggestions from the search engine comes down from a lack of data in the knowledge base. In 2010, the student driven newspaper Studvest performed a non-academic experiment of different disciplines within Wikipedia. In this study, The economic discipline was reported as lacking, both in quantity and quality. The quality of the results were tested by experts within the respective disciplines (Matre & Tjeldflåt, 2010). The results from this experiment is not directly applicable to this thesis, but the indication of lower than average quality on the articles within the discipline is present.

As previously mentioned the goal of this evaluation is to use experts to determine if the suggested terms are relevant. The usage of a search engine, such as the system presented, are general applications and not contained only within the disciplines represented by the experts in this evaluation. This evaluation is designed to determine if the suggested terms are relevant within a field, as it is not possible to exhaustively test all possible subjects, without further optimization of the engine. This topic will be further discussed in section 6.2.

6 Conclusion and Future Work

Before starting the work with this thesis, it became apparent that there was a lack of options for users to perform searching operations using Norwegian language. In this thesis a system for helping users formulate a precise search query has been presented. By exploiting implicit data from Wikipedia the system is designed to help the users formulate queries, and the users are supported by a search engine customized for Norwegian language. The goal for the evaluation is to gather data to help answer the research questions formulated in section 1.2.

The research question states:

"How to improve the search experience and results in Norwegian by query expansion using Wikipedia as a knowledge base?"

To answer this question some subquestions needs to be answered. Firstly can the system improve the search experience? The goal within this formulation is to determine if the system can help the users formulate a query by suggesting relevant words. The expert evaluation shows that the terms can be considered related in many cases, but not all. The sample size of experts is too small to be able to conclude that the suggestions are related, but the evaluation gives an indication that the words are related. The presentation of related words can help the user to formulate the query, and transform a broad searching strategy to a more precise strategy, for example the complex search strategy, as mentioned in section 2.1.3. The complex search strategy requires prior knowledge, and the method can in some cases help this transformation, even with users without domain knowledge.

The second question that needs to be answered in order to answer the research question, is can the system increase the quality of the results?
The method presented in this thesis shows that it is possible to increase the quality of the results using Norwegian algorithms, and query expansion method designed for Norwegian language.

By using a search engine customized for Norwegian language, the result quality increased by 13 %. There is a limited number of participants, commercial or others, that offers retrieval operations configured for Norwegian language. This demonstrates how important it is for users to have an option that caters searching in the Norwegian language.

Compared with using no query expansion algorithm the engine yields a 2.4 % increase in precision at the tenth article, and a 3.6 % increase in precision at the twentieth article.

Compared to searching in Norwegian in a search engine configured for English language, the thirteen percent increase in quality, combined with the suggested terms from the system, the system has a potential to help the users move from the broad search strategy to the more advanced strategies

6.1 Design and Development Process

In this section I discuss the design and development process, and highlight aspects that did or didn't work as intended. The first part is regarding the design of the system, and the second part is about the process leading up to the design.

6.1.1 Design

In the evaluation chapter, one of the results was that the results from concepts with a name consisting of two terms, polynyms, scores low, as the engine splits the words down to two separate terms. In this process the term loses its meaning and the results follow from this. This is an aspect where further work can improve the results. With proper optimization, both on the hardware side and on the software side, it could be fixed by expand the terms as a whole before the splitting, but the performance considerations took presence in this project.

Guisado-Gamez argued that the quality of the knowledge base directly affected the quality of query expansion (Guisado-Gámez et al., 2013). This trend is observable in this system. The Norwegian Wikipedia is much smaller than the English version, and therefore the results are affected by this.

Earlier studies that concluded that using Wikipedia as a knowledge base leads to improved results, but using the Norwegian Wikipedia, the results are slightly improved compared to not using a query expansion algorithm. This claim is based on the results from the evaluation. One reason for this effect is the size of the Norwegian Wikipeda compared to the English Wikipeda. The size difference reflects the amount of articles and how well they are linked together. This system is dependent on the connections between articles to recommend useful terms to the user.

The computational power required for this system is quite large and a search might take between five or fifteen seconds to perform. If this system was to be optimized in regard to performance, a more extensive user study on the system is important.

If this system were set up as a web server with proper hardware, and optimized hardware it would lead to possibilities regard further research on the subject, and hopefully succeed as a standalone search engine designed for Norwegian language.

6.1.2 Development

The design and development of the system, was as previously mentioned performed iteratively.

Different ideas and possible tracks for the system were tested during the development phase before settling on the final prototype presented in this paper. The testing of different methods are in line with the guidelines presented in section 3.1.

In the optimization phase, presented in section 4.7.6 different approaches was tested. In this phase the system was tested with multiple threshold levels, and different implementations of centrality, and with community detection algorithm. The iterative nature of this development style was highly effective as it allowed for constant improvement.

The extending of the Terrier search engine was a process where the development halted progress. A lack of understanding of the concepts essential to extending was missing when the progress of extending was started. This had a major impact on the development, as the time consumed for this part took much longer than anticipated in the planning of the development.

6.1.3 Evaluation Methods

As previously mentioned in section 2.6, Manning et. al. argues that the key utility for any information retrieval application is user happiness. To sufficiently evaluate the user happiness for an information retrieval application, user testing should be performed. Unfortunately, this system is very computationally demanding and the hardware available for evaluation purposes were not sufficient to achieve the level of performance to be able to compete with other search engines. Hoxmeier and DiCesare (2000) argues that the user satisfaction decreases over time, within a 12 second window (Hoxmeier & DiCesare, 2000). This system performs on the lower edge of this 12 second time frame, and the user satisfaction would not be a viable evaluation of the system, as the user would judge the system as worse based on the performance of the system.

The aforementioned TREC collections with automated evaluation of Information Retrieval applications, makes it possible for researchers to efficiently evaluate the results. The lack of such possibilities makes the evaluation of Information Retrieval applications in Norwegian an issue. The relevance between results and user generated queries must be generated manually. For this thesis only over 7000 results was manually considered as relevant or non-relevant regarding query terms. This is a time consuming process and the result of this is for example the lack of recall in the statistical analysis. As mentioned in section 2.6.2, recall requires that all the documents are judged as relevant or non-relevant for a query and the time required for this operation is not viable for a project of this scope. In the TREC conferences the data set is also of a nontrivial size, so the usual method for calculating recall manually is not viable. The TREC collection use a technique known as pooling to find the relative truth. This method was attempted in this project to estimate the recall levels, but the time required for performing this operation made it not viable. Pooling using a subset of the query terms was considered, but the data from the subset is not sufficient to be able to use the data to as arguments towards answering the research questions.

6.2 Future Work

In this part future improvements of the thesis is presented. The focus will be both on further work with the system as a research artifact, and as a functional product.

The data available in Wikipedia extends further than just the link analysis. If the engine were extended to include usage of the actual data from Wikipedia, it could further help users to formulate the query.

In this project, the links between articles in Wikipedia was treated as equal. There are possibilities for future improvements by treating different links with different weighting. As an example, Wikipedia Redirects are currently considered as an article. If the redirect-articles are instead considered as a link with much higher weighting than the normal link

6.2.1 Wikipedias quality

The usage of Wikipedia as a knowledge base, aided by algorithms has been proven efficient using both artificial intelligence (Døskeland, 2012) and using graph theory (Guisado-Gámez et al., 2013). These previous attempts have been performed on the English Wikipedia, a much larger knowledge base. The Norwegian Wikipedia is growing but the size is not substantial compared to the English Wikipeda.

This project resulted in marginally better results with query expansion than without. This comparison would be interesting to perform at a later date in time. As Wikipedia expands and improves over time, it would be interesting to observe if the quality of the results from the search engine would improve as a side effect.

6.2.2 Optimization of Performance

As previously mentioned, the system is very computationally expensive to run. If the system was optimized with both proper hardware and optimization of the system, there would be possibilities for more exact suggestions of words, and better user experience.

In the current state, the community detection algorithm is limited to a subgraph of 1000 nodes. If the threshold level for the number of nodes was raised, the community detection algorithm could be more precise. It is possible to perform the community detection algorithm in advance, as a preprocessed property of on the graph, but the size of the graph would increase massively, and the Wikipedia graph is not a static graph, and the chosen method responds better to changes.

6.2.3 Evaluation

This section will present evaluation methods that could be performed as future work.

As mentioned in section 2.6.1 Manning et. al (2009) argues that the key utility for any information retrieval application is user happiness. For this system to be able to compete in a user study, focusing on user happiness the system needs optimization as mentioned in section 6.2.2. If the system were developed with interaction design in mind, and optimized, it would be possible to for the system to compete in a more extensive user evaluation. By doing so would be possible to determine if this form for suggestions would help the user formulate queries using the more efficient searching strategies without background knowledge of the domain, and if the user actually wants this to happen, or if they want to iteratively refine the formulation until the search query matches the information need.

7 References

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. *New York* (Vol. 463, p. 513). doi:10.1080/14735789709366603

Bastani, K. (2015). Mazerunner. Retrieved from https://github.com/kbastani/neo4j-mazerunner

Broder, A. (2002). A Taxonomy of web search. Retrieved March 05, 2015, from http://doi.acm.org/10.1145/792550.792552

Brown, A. R. (2011). Wikipedia as a Data Source for Political Scientists: Accuracy and Completeness of Coverage. Retrieved March 11, 2015, from http://adambrown.info/docs/research/brown-2011-wikipedia-as-a-data-source.pdf

Bruusgaard, J. (2005). Norwegian Stop words. Retrieved from http://snowball.tartarus.org/algorithms/norwegian/stop.txt

Bunescu, R., & Pasca, M. (2006). Using Encyclopedic Knowledge for Named Entity Disambiguation. Retrieved March 07, 2014, from http://www.cs.utexas.edu/~ml/papers/encyc-eacl-06.pdf

Cattell, R. (2001). Scalable SQL and NoSQL data stores. Retrieved April 16, 2015, from http://doi.acm.org/10.1145/1978915.1978919

Chu, Z. L. ·. W. W. (2007). Knowledge-based query expansion to support scenario-specific retrieval of medical free text. Retrieved February 25, 2014, from http://dx.doi.org/10.1007/s10791-006-9020-6

Cohn, M. (2004). *User Stories Applied* (p. 291). Redwood City, CA: Addison Wesley Longman Publishing Co., Inc. Retrieved from http://www.trilemon.com/wp-content/uploads/2012/05/User-Stories-Applied-For-Agile-Software-Development.pdf

Custis, T., & Al-Kofahi, K. (2007). A New Approach for Evaluating Query Expansion: Query-Document Term Mismatch. Retrieved March 05, 2015, from http://doi.acm.org/10.1145/1277741.1277840

Døskeland, Ø. (2012). Use of Wikipedia content, structural connections and usage statistics to generate context aware query augmentation in a topical search engine. Retrieved March 05, 2014, from https://bora.uib.no/bitstream/handle/1956/6394/103699701.pdf?sequence=1

Fjellstad, R. (2008). Wiki - Prosessen, en god nok kvalitetssikring? Retrieved March 10, 2015, from http://brage.bibsys.no/xmlui/bitstream/handle/11250/145774/RFjellstadMaster.pdf?sequence=3&isAllowed=y

Gerber, A., Merwe, A. van der, & Alberts, R. (2007). Practical Implications of Rapid Development Methodologies. Retrieved May 25, 2015, from http://www.academia.edu/1052370/Practical_Implications_of_Rapid_Development_Methodologies

Guisado-Gámez, J., Dominguez-Sal, D., & Larriba-Pey, J.-Ll. (2013, October 21). Massive Query Expansion by Exploiting Graph Knowledge Bases [Information Retrieval]. Retrieved February 20, 2014, from http://arxiv.org/abs/1310.5698

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. Retrieved April 17, 2015, from http://em.wtu.edu.cn/mis/jxkz/sjkx.pdf

Hoxmeier, J., & DiCesare, C. (2000). System Response Time and User Satisfaction : An Experimental Study of Browser-based Applications. *Proceedings of the Association of Information …*, 1–26. doi:10.1.1.99.2770

Iadh Ounis, Vassilis Plachouras, Ben He, Craigh Macdonald, C. L. (2006). Terrier: A High Performance and Scalable Information Retrieval Platf orm. Retrieved February 17, 2015, from http://terrierteam.dcs.gla.ac.uk/publications/ounis06terrier-osir.pdf

Joshua O ' Madadhain, Fisher, D., Nelson, T., White, S., & Boey, Y.-B. (2005). The Java Universal Network/Graph Framework (JUNG): A Brief T our. Retrieved from http://jung.sourceforge.net/presentations/JUNG_M2K.pdf

Luo, F., Wang, J. Z., & Promislow, E. (2006). Exploring Local Community Structures in Large Networks. Retrieved March 25, 2015, from http://dx.doi.org/10.1109/WI.2006.72

Manning, C. D., & Raghavan, P. (2009). An Introduction to Information Retrieval. (A. C.-B. E. Salas, Ed.)*Online*. Cambridge University Press. doi:10.1109/LPT.2009.2020494

Matre, A. K. F., & Tjeldflåt, G. M. (2010). Wikipedia på godt og vondt. *Studvest*, p. 1. Bergen.

Middleton, C., & Baeza-Yates, R. (2007). A Comparison of Open Source Search Engines. Retrieved May 12, 2015, from http://wrg.upf.edu/WRG/dctos/Middleton-Baeza.pdf

Milne, D. N., Witten, I. H., & Nichols, D. M. (2007). A knowledge-based search engine powered by wikipedia. doi:http://dx.doi.org/10.1145/1321440.1321504

Nasato, M. (2015). Graphipedia. https://github.com/mirkonasato/graphipedia.

Oates, B. J. (2006). *Researching Information Systems and Computing*. *Inorganic Chemistry* (Vol. 37, p. 341). doi:10.1016/j.ijinfomgt.2006.07.009

Otegi, A. (2011). Query Expansion for IR using Knowledge-Based Relatedness. Retrieved March 03, 2014, from http://aclweb.org/anthology//I/I11/I11-1175.pdf

Porter, M. (1979). The Porter Stemming Algorithm. Retrieved March 11, 2015, from http://tartarus.org/~martin/PorterStemmer/index-old.html

R. Feil, L. W. C. (2015). The ThT System - A Multilingu al Nordic Search Interface NorNa. Retrieved February 26, 2014, from http://static.sdu.dk/mediafiles/Files/Om_SDU/Institutter/Ifki/Norna _pdf/The_ThT_System_paper.pdf

Robinson, I., Webber, J., & Eifrem, E. (2013). *Graph Databases*. *O'Reilly Media, Inc.* Retrieved from http://info.neotechnology.com/rs/neotechnology/images/GraphDatabases.pdf

Rudd, J., Stern, K., & Isensee, S. (1996). Low Vs High. Retrieved March 09, 2015, from http://doi.acm.org/10.1145/223500.223514

Sucheck, K., Salah, alkim almila akdag, Cheng, G., & Scharnhorst, A. (2012). Evolution of Wikipedias Category Structure. *World Scientific Publishing Company*, 19. Retrieved from http://arxiv.org/pdf/1203.0788.pdf

Sutcliffe A, Ennis, M. (1998). Towards a cognitive theory of information retrieval. *Interacting with Computers*, *10*.

Tang, L., & Liu, H. (2010). *Community Detection and Mining in Social Media*. Morgan & Claypool.

Vikør, L. S. (2010). Norsk. In *Store Norske Leksikon*. Retrieved from https://snl.no/norsk

Wikipedia Foundation. (2015). Wikipedia Statistikk. Retrieved from http://no.wikipedia.org/wiki/Wikipedia:Statistikk

Xu, J., & Croft, W. B. (1996). Query Expansion Using Local and Global Document Analysis. Retrieved March 03, 2015, from http://doi.acm.org/10.1145/243199.243202

Yamin, F. M., & Ramayah, T. (2001). User Web Search Behavior on Query Formulation. *International Conference on Semantic Technology and Information Retrieval*. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5995786&tag=1

Zahl, S. (2009). *Komparativ undersøkelse på naturfaglige artikler, med fokus på faglig kvalitet. Wikipedia vs. Store Norske Leksikon*. Høyskolen i Nesna. Retrieved from http://brage.bibsys.no/xmlui/bitstream/handle/11250/145766/SnorreMasterEks.pdf?sequence=1&isAllowed=y

Appendix A Terms For evaluation

Reddit (Sosialt nettsed)

Tottenham (Football team)

Facebook (Social network site)

Irland (Geographical area)

Vin (alchoholic beverage)

SAS (Norwegian Airline)

Fotball (Sport)

Radioresepsjonen(Radioshow)

Klubb (Team)

Agder (Geographical Area)

Instrumenter (Musical instruments)

Hitler (Adolf Hitler

Gevær (Rifle)

Sitrus (Citrus Fruits)

Væske (Liquid)

Frukt (Fruits)

Magnus Carlsen (Chess Player)

India (Geographical Entity)

Scandinavian Star (Accident)

Gitar (Musical instrument)

Kollektivtransport (public transport)

Iphone (Mobile Phone)

Eurasia (Geographical Area)

Depresjon (Psychological illness)

Kalashnikov (Weapon)

Sushi (Food)

Star Wars (Movie Franchise)

Deepwater Horizon (Accident in the gulf of Mexico)

Führer (German title for leader, often assoicated with Hitler)

007 (James Bond)

USSR (Older name for Sovjet Union or Russia)

Newton (Sir Isaac Newton)

Napoli (Geographical Area)

Røde Kors (Red Cross organization)

Steve Jobs (Founder of Apple Inc)

Toyota (Car maker)

Tiger Woods (Professional golf player)

Kate Middleton (Wife of prince Williams)

Vuvuzela (Instrument)

Wikileaks (Organization)

Aung San Suu Kyi (Nobel peace prize winner)

Pyongyang (Capitol)

Aspergers (Psychologil illness)

Richard Nixon (Earlier president of USA)

Spionasje (Espionage – concept)

SS (Shutstaffel – famous part of German Army during WW2)

NATO (Alliance of nations)

EU (European union)

Uaktsomhet

The Simpsons (TV-franchise)

The Beatles (Band)

Appendix B Economic expert 1

Skalaavkastning - det stemmer at det er et begrep innen konsumentteori (mikroøkonomi)

Ricardiansk ekvivalens - David Ricardo, og ikke Ricardi

Balansert vekstbane (steady state) - funksjon og komposisjon er forsåvidt greie nok, vet ikke helt med "bankkonto" og frekvensmodulasjon derviat har jeg aldri hørt om, men tipper det er noe innenfor finans - og det kan godt være det er noe innen finans som også heter balansert vekstbane. Balanse passer veldig godt. Ellers ville det vært naturlig at "Solow" (han som har funnet opp modellen) dukket opp?

Taylor-regelen - aldri hørt om noen av tingene...

Adaptive forventninger - ser ikke helts sammenhengen her heller.

Likevekt - mye ting jeg ikke har hørt om, men som helt sikkert har en sammenheng. Spillteori og balanse er bra i alle fall.

Nash - samme som likevekt. Samfunnsøkonomi og spillteori passer godt, de andre vet jeg ikke, men har sikkert en eller annen sammenheng.

Alt i alt, syns ikke det passet så veldig bra? Men nå kan det hende det var veldig dårlige begreper fra min side da.

Appendix C Economic expert 2

Samfunnsøkonomi:
”No result”

Samfunnsøkonomi er et ganske bredt tema, og som således kan relateres til veldig mange ulike aspekter.

Komparative fortrinn:
Komparative fortrinn er en term knyttet til internasjonal handel, og motoren treffer ganske bra med søkeresultatene ”EFTA” og ”handel”. Motoren lister også opp ulike land, noe som er positivt da termen ofte knyttes opp mot nasjoner. ”Derivat” er en finansterm som ikke kan knyttes direkte til komparative fortrinn.

Finans:
Motoren treffer godt med resultatene ”finansloven” og ”dagens næringsliv”, i tillegg til at den relaterer termen mot Dow Jones. På den andre siden er ikke resultatene ”Sambandsflokkurin”, ”Norges

historie", "Færøyene", "Danmark" og "1970-tallet" relevant innfor fagtermen finans. Dette er en svært bred term, og det er tydelig at motoren har hengt seg opp i historie for Færøyene.

Budsjettering:
Denne fagtermen er også ganske bred, og resultatene "Norges Handelshøyskole" og "Stortinget" er relevante. "Tv-program", "representativt demokrati" og "Mauritius" knyttes ikke direkte opp mot fagtermen.

Finanskrise:
Motoren treffer veldig godt på resultatet "Finanskrisen 2007-2010". De andre resultatene er ikke relevante.

Pengepolitikk:
Denne fagtermen er i hovedsak relatert til makroøkonomi og politikk, og resultatene knyttet opp mot land og "Arbeiderpartiet" ansees som positivt. Jeg vil heller ikke påstå at motoren er på villspor når den foreslår "Statistisk sentralbyrå" og "Oslo", selv om disse resultatene i seg selv ikke er direkte relevante.

Personlig økonomi:
"No result"

Jeg tror det er vanskelig å finne gode resultater på dette søkermålet, og synes det er forståelig at motoren ikke har kommet med noen resultater.

Monopol:
Motoren gir gode resultater knyttet til dette søkermålet. Resultatene er delt mellom fagtermen, men også brettspillet Monopol. Den klarer å knytte fagtermen opp mot "Konkurransetilsynet" og "pris", noe jeg mener er gode resultater. I tillegg vil jeg si at "monopol (brettspill)" og "Torggata Oslo" er relevante resultater knyttet opp mot brettspillet. Resten av resultatene kan ikke direkte knyttes til fagtermen.

Aksjer:
Motoren treffer bra med "Alfagruppen" og "emisjon". "Internett" er et positivt resultat, selv om det ikke kan knyttes direkte til termen. "Byggma" er uinteressant.

Regnskap:
Motoren gir ett resultat, og treffer bra på dette; "lønnsavstemning".

Appendix D Law expert 1

Lex superior er et rettsprinsipp, som går ut på at rettsregler av høyere rang går foran regler av lavere rang ved motstrid mellom reglene.

- *Lex posterior* er en av de andre lex-reglene, i tillegg til lex specialis, og er dermed relevant

- Derogasjon: Det er et juridisk begrep, men er ikke relevant i denne sammenhengen.
- *Forvaltningsrett*: Lex superior-prinsippet er styrende for mye av forvaltningsrettens område og er dermed relevant.
- *Rettsvitenskapsforslaget* sier seg selv at er relevant. Lex superior er et av rettsvitenskapens grunnleggende prinsipper. Det samme er *rettskildelære* som egentlig betyr det samme som rettsvitenskap.
- *Tysklands politiske system, boplikt og swati* har ikke noe med lex superior å gjøre.

Presumsjonsprinsippet: Ingen resultat.

Culpa: Culparegelen er en erstatningsrettslig regel om at den som ved forsett eller uaktsomhet er skyld i at en annen lider økonomisk tap, har plikt til å erstatte dette.

- *Culpa levissima*: Culpa levissima er en særlig lett form for uaktsomhet, som også betyr den letteste uaktsomhet. Culpa levissima kommer til anvendelse der man begår en handling, men det likevel skjer en uforsettlig følge av handlingen (følgen var ikke tilsiktet), som fører til strengere straff. Dette er med andre ord en annen grad for culpa-skyld, og har en klar sammenheng med culpa.
- I og med at dette er en forsett/uakstomhetsregel er også forslagene *uaktsomt* og *uaktsomhet* relevant.
- *Lars Martin Myhre, psykedelisk musikk, womens professional soccer,* og *gud fader* er ikke av interesse i denne sammenheng.

Deklaratorisk: Ingen resultat.

Avsavnsrente: Avsavnsrente er en rente som skal dekke det tap som en person/kreditor har hatt ved at han ikke selv har hatt mulighet til å disponere over pengene sine.

- *Rente:* Forslaget er relevant ettersom det her er snakk om renter.
- *Mora:* Morarente er det samme som forsinkelsesrente – man kan få forsinkelsesrente både av den opprinnelige summen og avsavnsrentene, slik at dette forslaget i høyeste grad må anses som relevant.
- *Viggo Hagstrøm:* Var professor i jus og spesialiserte seg særlig innenfor privatrettens område – han har skrevet lærebøker og oppslagsverk om kjøpsrett og obligasjonsrett, hvor avsavnsrente er behandlet, slik at dette er av aktuell karakter.

Servitutt: En servitutt er en rettighet man kan ha på en annens grunneiendom, som begrenser denne grunneierens bruk av egen eiendom. Som oftest en avtale.

- *Servitutt* og *servituttlova:* Høyst relevant. Servituttloven er den loven som omhandler servitutter.
- Jeg anser ikke *liste over rettsområder* som relevant da dette ikke direkte har noe med servitutter å gjøre. Man kan på den andre side inngå servitutter på mange ulike rettsområder slik at man kan anse dette som relevant dersom man tenker seg en liste over de områdene der det finnes servitutter.

Eiendomshevd: Dreier seg om at hevderen kan, dersom vilkårene er oppfylt, hevde eiendomsrett til eiendom eller løsøre og dermed ekstingvere den opprinnelige eieres rettigheter til eiendommen/tingen.

- *Hevd* er relevant ettersom eiendomshevd er en del av hovedregelen, hevd. Forskjellen er utelukkende at eiendomshevd er mer spesifisert.

Skjevdeling: Ingen resultat.

Rettshåndhevelsesarrest: Ingen resultat.

Uskyldspresumsjonen: Dette er et prinsipp innen strafferetten som tilsier at tvil om de faktiske sider i saken skal komme tiltalte til gode.

- *Stavanger tingrett:* Stavanger tingrett har tatt stilling til en rekke skyldspørsmål, og kan dermed være relevant, men de er bare en av få. Stavanger tingrett er ikke mer relevant enn andre domstoler i Norge.
- *Norsk tidsskriftforening*: Ikke relevant slik jeg kan se det.
- *Moen-saken*: Denne saken dreide seg om anklage om dobbelt justismord, altså var uskyldspresumsjonen her sentral og dommen er dermed relevant.
- *Den europeiske menneskerettighetskonvensjon:* EMK art. 6.2 fastsetter dette prinsippet – og er dermed i høyeste grad aktuelt.
- *Jon Ausonius*: Etter litt googling har jeg kommet frem til at Jon Ausonius er en drapsdømt svensk seriemorder og bankraner. Under hans sak var selvfølgelig uskyldspresumsjonen aktuell ettersom man skal anses som uskyldig til det motsatte er bevist. I dette tilfellet var det ikke vanskelig å finne bevis for skyld og han ble dømt.
- *Kriminalomsorgen*: Relevant ettersom det er kriminalomsorgen som i stor grad tar seg av varetektsfengsling, samt skal sørge for at straffereaksjoner blir gjennomført på en skikkelig måte. Kriminalomsorgens daglige oppgaver er behandling av skyldige, noe som gjør at uskyldspresumsjonen er relevant også for dem.
- *Strasbourg*: Den europeiske menneskerettighetsdomstol ligger i Strasbourg, og der er den som dømmer i saker som omhandler mulige brudd på EMK – dermed har dette nær sammenheng med uskyldspresumsjonen.

Appendix E Law expert 2

Lex Superior: - Lex Posterior, derogasjon, forvaltningsrett, rettsvitenskap, er fagutrykk som har en viss relevans i forhold til søkeordet.

Presumsjonsprinsippet: - dette er et uttrykk/begrep som anvendes innefor faget, men som antakelig ikke kan ses på som noe fagutrykk.

Culpa: Uaktsomt og culpa levissima, er relevante., Psykedelisk musikk og Womans professional hører nok til andre fagfelt.

Deklaratorisk: Ordet betyr fravikelig. (Motsatt; - preseptorisk (ufravikelig)). Sentralt juridisk begrep. Dersom det begrepet finnes i artikler på W, er det svakhet ved programmet at dette ikke er angitt.

Avsavnsrente: Rente + mora er relevante. Viggo Hagstrøm; jusproffesor som har skrevet om bl.a. avsanvensrente.

Eiendomshevd: Hevd i form av vedlikehold (holde i hevd) er feil. Juridisk begrep hevd er noe annet. NB! Som søkeord bør du ikke bruke eiendomshevd, men kun ”hevd”. (Dette omfatter hevd av både eiendomsrett + hevd av bruksrett).

Skjevdeling: Her burde programmet funnet artikler dersom det eksisterer artikler hvor begrepet er anvendt.

Rettshåndhevelsesarrest: - OK med ingen treff. Begrepet er svært spesielt.
Uskyldspresumsjon: Treffene er relevante, og da spesielt den europeiske menneskrettighetskommisjon. Stavanger tingrett antas å ha kommet med fordi det er artikler hvor begrepet er anvendt og som refererer seg til Stavanger tingrett.

Generelt: På hele listen på 10 søkeord er det 2 av svarene som ikke har faglig relevans overhode. Dette gjelder ”psykedelisk musikk” og ”Womans professional”. Øvrige forslag har en eller annen faglig relevans. Jeg forstår det slik at der hvor det er angitt ”no result” kan bero på mangel på artikler hvor aktuelle søkeord er anvendt, og i så fall er resultatet korrekt.

Appendix F Archeology expert

Cortex: Heter korteks på norsk så kan hende dette er årsaken til null treff.
proximalende: skulle stått *Proksimalende. (ser forøvrig at det er en term som brukesa av andre disipliner, F.eks Medesin.*

Mesolitikun: ga gode treff: Eneste der er at Det heter Traktbegerkulturen
strategrafisk: stratigrafisk og ikke strategrafisk. Antar dette er grunnen for så få treff.
harrismatrix: heter hvist *Harris matrise på norsk og ikke Harrismatrix (skal forsåvidt deles opp til harris matrix),*
Antar dette er grunnen for så få treff
kokegrop: Relativt gode treff, da jeg ser for meg at denne popper opp i mange feltraporter (derav Europavei, nøtterøy og borheim). Kokegrop er en struktur så den er bra. Usikker på billedkunst og Schussler.
Flekke: ingen gode treff etter hva jeg kan se. kan hende denne ville hatt bedre suksess om du brukte mikroflekke i stedet. Dette er kanskej endå mere sereget for arkeologi.
bipolar (kjerne): Dropp parantesene. bipolar alene blir nok delt med for mange andre disipliner.
singel context: Skulle selvsagt vært single context. (dyslexia 101)
jernalder: Greie treff, men ville trodd det skulle være flere her.
kulturlag: Meget passnede treff.
Terminus: burde antaklig stått terminus post quem. men ser at dette iukke blir brukt så mye. så det kan også være grunnen til at trffene ikke er så gode.
Antequem og postquem: begge skulle vært delt opp, ante quem, post quem. i tillegg henger de sammen med forrige term, så kanskje du skulle vurdert å bytte ut 2 av dem.
langhaug: deter funnet langhauger på årstad og i sunnfjord, så dette er greie treff.
Klyngetun: Denne er også grei.

Hjerne
Occipitallapp

Denne deles med medesin, så ser at den ikke ga noen rellevante svar. (har du flint med i søket kommer det nok flere treff)

Appendix G Precision

Queries	Topic	With QE No Community detection		With QE No Community detection		With QE Community detection		With QE Community detection		No QE		No QE	
		Presicion @ 10		Presicion @ 20		Presicion @ 10		Presicion @ 20		Presicion @ 10		Presicion @ 20	
Reddit (Sosialt nettsted)	Organization	0.4		0.4		0.4		0.4		0.4		0.25	
Tottenham (Football team)	Concept	1		1		1		1		1		1	
Facebook (Social network site)	Organization	1		0.95		1		0.95		1		0.65	
Irland (Geographical area)	Geography	0.9		0.9		0.9		0.9		0.8		0.65	
Vin (alcoholic beverage)	Generalization	0.9		0.95		1		1		1		1	
SAS (Norwegian Airline)	Organization	1		1		1		1		1		1	
Radio (Concept)	Concept	1		0.8		0.9		0.8		0.9		0.65	
Radiorespsjonen(Radioshow)	Franchise	0.1		0.05		0.1		0.05		0.1		0.05	
Klubb (Team)	Organization	0.5		0.45		0.5		0.45		0.5		0.45	
Agder (Geographical Area)	Geography	0.9		0.65		0.6		0.65		0.7		0.5	
Instrumenter (Musical instruments)	Generalization	0.7		0.65		0.5		0.35		0.5		0.5	
Hitler (Adolf Hitler	Person / Historical Figure	0.6		0.5		0.7		0.6		0.7		0.65	
Gener (Noe annet)	Generalization	0.5		0.5		0.3		0.4		0.7		0.55	
Situs (Citrus Fruits)	Specialization	0.3		0.15		0.3		0.3		0.3		0.15	
Veske (Liquid)	Generalization	0.9		0.6		0.9		0.6		0.9		0.55	
Frukt (Fruits)	Generalization	0.1		0.2		0.1		0.2		0.1		0.2	
Magnus Carlsen (Chess Player)	Person	1		1		1		1		1		1	
Nedbør (Geographical Area)	Geography	1		0.9		0.9		0.85		0.9		0.95	
Sosialisme (Social Ideology)	Specialization	1		1		1		1		0.9		0.95	
Gitar (Musical instrument)	Physical Entity	0.5		0.3		0.5		0.3		0.1		0.15	
Kollektivtransport (public transport)	Concept	0.7		0.6		0.7		0.6		0.9		0.6	
Iphone (Mobile Phone)	Physical Entity	0.8		0.65		1		0.95		1		0.9	
Eurasia (Geographical Area)	Geography	1		0.95		0		0		0		0	
Alzheimer (Psychological Illness)	Concept	1		0.65		1		0.65		0		0.65	
Kokeboker (Wikipedia)	Specialization	0.4		Not enough data		0.3		Not enough data		0.3		Not enough data	
Sushi (Food)	Specialization	0.7		0.75		0.7		0.75		0.6		0.5	
Star Wars (Movie Franchise)	Franchise	1		0.75		1		0.75		0.8		0.6	
Deepwater Horizon (Accident in the gulf of Mexico)	Incident	1		0.95		1		1		1		1	
Führer (German title for leader, often associated with Hitler)	Title	0.1		0.05		0.1		0.05		0.1		0.05	
Norge (Norway)	Person	0.5		0.3		0.5		0.3		0.5		0.5	
USSR (Older name for Soviet Union or Russia)	Geography/Nation	0.4		0.2		0.4		0.2		0		0	
Newton (Sir Isaac Newton)	Person	0		0		0		0		0		0	
Napoli (Geographical Area)	Geography	0		0		0		0		0		0	
Røde Kors (Red Cross organization)	Organization	0		0		0.9		0.75		0.7		0.6	
Steve Jobs (Founder of Apple Inc)	Person	0.9		0.6		0.9		0.65		0.9		0.6	
Tiger Woods (Professional golf player)	Person	0.5		0.45		0.6		0.45		1		1	
Tiger Woods (Professional golf player)	Person	1		1		1		1		1		1	
Kate Middleton (Wife of prince William)	Person	0.5		0.05		0.5		0.45		0.3		0.25	
Vinyasela (Instrument)	Physical Entity	0		0		0		0		0		0	
Wikileaks (Organization)	Organization	0.3		0.3		0.3		0.3		0.5		0.45	
Aung San Suu Kyi (Nobel peace prize winner)	Person	1		0.95		1		0.95		1		0.95	
Bergen (Geographical Area)	Geography	0.2		0.1		0.2		0.1		0.2		0.1	
Asperger's (Psychological illness)	Concept	0.8		0.5		0.8		0.6		0.5		0.3	
Richard Nixon (Earlier president of USA)	Person	0.1		0.05		0.1		0.05		0		0.05	
Spionasje (Espionage – concept)	Concept	0.9		0.95		1		0.9		1		0.9	
SS (Shutzstaffel – famous part of German Army during WW2)	Abbreviation	0.5		0.3		0.5		0.3		0.5		0.3	
NATO (Alliance of nations)	Abbreviation	1		0.95		1		1		1		1	
EU (European union)	Abbreviation	1		0.9		1		1		1		0.95	
Uaktsomhet	Concept												
The Simpsons (TV-franchise)	Franchise	0.3		0.15		0.3		0.15		0.3		0.15	
The Beatles (Band)	Franchise	0.7		0.6		0.7		0.6		0.7		0.6	
COUNT	0												
	0												
Gjennomsnitt		0.616		0.528		0.63		0.55		0.608		0.514	

Appendix H BPREF With QE

WITH QE COMDEC

Queries	Document Number										Average Precision	Number of Relevant		
	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10				
Reddit (Sosialt nettsed)	Ja	Nei	Ja	Ja	Nei	Nei	Nei	Nei	Nei	Ja	0.7041666667	4	0.5	
Tottenham (Football team)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1	10	1	
Facebook (Social network site)	Ja	ja	ja	Ja	Ja	Ja	Ja	Ja	ja	Ja	1	10	1	
Irland (Geographical area)	ja	ja	JA	Ja	Ja	Ja	Ja	Nei	Ja	Ja	9.6638888889	9	0.975308642	
Vin (alchoholic beverage)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1	10	1	
SAS (Norwegian Airline)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1	10	1	
Fotball (Sport)	ja	ja	ja	ja	ja	ja	ja	ja	Nei	ja	0.9888888889	9	0.987654321	
Radioresepsjonen(Radioshow)	Ja	Nei	nei	nei	nei	nei	nei	nei	nei	nei	1	1	1	
Klubb (Team)	Ja	Ja	ja	Ja	ja	nei	nei	nei	nei	nei	1	3	1	
Agder (Geographical Area)	Ja	nei	ja	nei	nei	nei	ja	ja	ja	ja	0.6251322751	6	0.6388888889	
Instrumenter (Musical instruments)	ja	ja	ja	ja	ja	nei	nei	nei	nei	ja	0.9333333333	6	0.8888888889	
Hitler (Adolf Hitler	ja	ja	ja	ja	nei	ja	nei	nei	ja	ja	0.8857142857	7	0.8571428571	
Gevær (Rifle)	ja	ja	ja	ja	ja	nei	nei	ja	ja	nei	0.9325396825	7	0.9183673469	
Sitrus (Citrus Fruits)	nei	nei	nei	ja	ja	nei	nei	nei	nei	ja	0.3166666667	3	0.2222222222	
Væske (Liquid)	ja	ja	ja	ja	ja	ja	ja	ja	nei	ja	0.9888888889	9	0.975308642	
Frukt (Fruits)	nei	nei	ja	nei	nei	nei	nei	nei	nei	nei	0.3333333333	1	-1	
Magnus Carlsen (Chess Player)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1	10	1	
India (Geographical Entity)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1	10	1	
Scandinavian Star (Accident)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1	10	1	
Gitar (Musical instrument)	nei	nei	ja	ja	ja	nei	nei	nei	nei	ja	0.4583333333	4	0.25	
Kollektivtransport (public transport)	ja	ja	ja	nei	ja	ja	ja	nei	ja	nei	0.8954648526	7	0.8979591837	
Iphone (Mobile Phone)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1	10	1	
Eurasia (Geographical Area)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0	0	0	
Depresjon (Psychological illness)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1	10	1	
Kalashnikov (Weapon)	ja	nei	nei	ja	nei	nei	ja	nei	nei	nei	0.6428571429	3	0.5277777778	
Sushi (Food)	ja	ja	ja	ja	nei	ja	ja	ja	nei	nei	0.8363095238	7	0.9387755102	
Star Wars (Movie Franchise)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1	10	1	
Deepwater Horizon (Accident in the gulf of Mexico)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1	10	1	
Führer (German title for leader, often assoicated with Hitler)	nei	nei	nei	ja	nei	nei	nei	nei	nei	nei	0.25	1	0.25	
007 (James Bond)	ja	ja	ja	ja	nei	nei	nei	ja	nei	nei	0.578125	5	0.88	
USSR (Older name for Sovjet Union or Russia)	nei	nei	nei	ja	nei	ja	ja	ja	nei	nei	0.3779761905	4	0.0625	
Newton (Sir Isaac Newton)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0	0	0	
Napoli (Geographical Area)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0	0	0	
Røde Kors (Red Cross organization)	ja	ja	ja	ja	ja	ja	ja	nei	ja	ja	1	9	1	
Steve Jobs (Founder of Apple Inc)	ja	ja	ja	ja	ja	ja	ja	ja	ja	nei	1	9	1	
Toyota (Car maker)	ja	ja	nei	nei	ja	nei	ja	ja	nei	ja	0.7327380952	6	0.7777777778	
Tiger Woods (Professional golf player)	ja	ja	ja	ja	ja	ja	ja	nei	ja	ja	0.9765432099	9	0.975308642	
Kate Middleton (Wife of prince Williams)	ja	ja	nei	nei	nei	nei	ja	ja	nei	ja	0.6857142857	5	0.48	
Vuvuzela (Instrument)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0	0	0	
Wikileaks (Organization)	ja	nei	nei	nei	nei	ja	nei	nei	nei	nei	0.6666666667	2	0.6	
Aung San Suu Kyi (Nobel peace prize winner)	ja	ja	ja	ja	ja	ja	ja	nei	ja	ja	1	9	0.975308642	
Pyongyang (Capitol)	nei	ja	nei	ja	nei	nei	nei	nei	nei	nei	0.5	2	0.7	
Aspergers (Psychologil illness)	ja	ja	ja	ja	ja	ja	nei	ja	ja	nei	0.9861111111	8	0.96875	
Richard Nixon (Earlier president of USA)	nei	nei	nei	ja	nei	nei	nei	nei	nei	nei	0.25	1	0.4	
Spionasje (Espionage – concept)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1	10	1	
SS (Shutstaffel – famous part of German Army during WW2)	ja	ja	ja	nei	ja	nei	ja	nei	nei	nei	0.8523809524	5	0.88	
NATO (Alliance of nations)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1	10	1	
EU (European union)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1	10	1	
												0		
The Simpsons (TV-franchise)	ja	ja	nei	ja	nei	nei	nei	nei	nei	nei	0.9166666667	3	0.8888888889	
The Beatles (Band)	nei	ja	ja	ja	ja	nei	ja	ja	ja	nei	0.7083900227	7	0.7959183673	
													36.2127465986	
											Count		50	
												BPREF	0.724254932	

Appendix I BPREF no Query Expansion

Queries	Document Number										Average Precision	Number of Relevant		
	1	2	3	4	5	6	7	8	9	10				
Reddit (Sosialt nettsed)	ja	nei	nei	nei	ja	nei	nei	ja	nei	ja	0.54375		4	0.125
Tottenham (Football team)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1		10	1
Facebook (Social network site)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1		10	1
Irland (Geographical area)	ja	ja	ja	ja	ja	ja	ja	Nei	ja	ja	0.9765432099		9	0.987654321
Vin (alchoholic beverage)	ja	ja	ja	ja	ja	nei	ja	ja	ja	ja	0.9467813051		9	0.950617284
SAS (Norwegian Airline)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1		10	1
Fotball (Sport)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1		10	1
Radioresepsjonen(Radioshow)	ja	nei	nei	nei	nei	nei	nei	nei	nei	nei	1		1	1
Klubb (Team)	Ja	Ja	ja	Ja	ja	nei	nei	nei	nei	nei	1		3	1
Agder (Geographical Area)	ja	nei	ja	ja	ja	ja	ja	ja	ja	ja	0.7424603175		9	0.9012345679
Instrumenter (Musical instruments)	ja	ja	ja	ja	nei	nei	nei	nei	nei	ja	0.9		5	0.84
Hitler (Adolf Hitler	ja	ja	ja	ja	ja	nei	nei	nei	ja	ja	0.9095238095		7	0.8775510204
Gevær (Rifle)	ja	ja	ja	ja	ja	nei	nei	nei	ja	ja	0.9095238095		7	0.8775510204
Sitrus (Citrus Fruits)	nei	nei	nei	ja	ja	ja	nei	nei	nei	nei	0.3833333333		3	0
Væske (Liquid)	ja	ja	ja	ja	ja	ja	ja	ja	nei	ja	0.9888888889		9	0.987654321
Frukt (Fruits)	nei	nei	ja	nei	nei	nei	nei	nei	nei	nei	0.3333333333		1	0.5
Magnus Carlsen (Chess Player)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1		10	1
India (Geographical Entity)	ja	ja	ja	nei	ja	ja	ja	ja	ja	ja	0.9060405644		9	0.9259259259
Scandinavian Star (Accident)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1		10	1
Gitar (Musical instrument)	nei	nei	ja	nei	nei	nei	nei	nei	nei	nei	0.3333333333		1	0.6
Kollektivtransport (public transport)	ja	ja	ja	ja	ja	ja	ja	ja	nei	ja	0.9888888889		9	0.987654321
Iphone (Mobile Phone)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1		10	1
Eurasia (Geographical Area)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0		0	0
Depresjon (Psychological illness)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1		10	1
Kalashnikov (Weapon)	ja	nei	ja	ja	nei	nei	nei	nei	nei	nei	0.8055555556		3	0.8055555556
Sushi (Food)	ja	ja	nei	ja	ja	nei	ja	ja	nei	nei	0.7801587302		6	0.8333333333
Star Wars (Movie Franchise)	ja	ja	ja	ja	ja	nei	ja	ja	ja	nei	0.9526289683		8	0.953125
Deepwater Horizon (Accident in the gulf of Mexico)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1		10	1
Führer (German title for leader, often assoicated with Hitler)	nei	nei	nei	ja	nei	nei	nei	nei	nei	nei	0.25		1	0.25
007 (James Bond)	ja	ja	ja	ja	nei	nei	nei	ja	nei	nei	0.925		5	0.88
USSR (Older name for Sovjet Union or Russia)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0		0	0
Newton (Sir Isaac Newton)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0		0	0
Napoli (Geographical Area)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0		0	0
Røde Kors (Red Cross organization)	ja	ja	ja	nei	ja	ja	ja	ja	nei	nei	0.9093537415		7	0.9183673469
Steve Jobs (Founder of Apple Inc)	ja	ja	nei	ja	ja	ja	ja	ja	ja	ja	0.8782627866		9	0.9135802469
Toyota (Car maker)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1		10	1
Tiger Woods (Professional golf player)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1		10	1
Kate Middleton (Wife of prince Williams)	ja	nei	nei	nei	nei	nei	nei	ja	ja	nei	0.5277777778		3	-0.3333333333
Vuvuzela (Instrument)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0		0	0
Wikileaks (Organization)	nei	ja	ja	nei	nei	ja	ja	ja	nei	nei	0.5726190476		5	0.56
Aung San Suu Kyi (Nobel peace prize winner)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1		10	1
Pyongyang (Capitol)	nei	ja	ja	nei	nei	nei	nei	nei	nei	nei	0.5833333333		2	0.8
Aspergers (Psychologil illness)	ja	ja	nei	nei	ja	ja	nei	nei	ja	nei	0.7644444444		5	0.68
Richard Nixon (Earlier president of USA)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0		0	0
Spionasje (Espionage – concept)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1		10	1
SS (Shutstaffel – famous part of German Army during WW2)	ja	ja	ja	ja	nei	nei	ja	nei	nei	nei	0.9428571429		5	0.92
NATO (Alliance of nations)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1		10	1
EU (European union)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1		10	1
													0	
The Simpsons (TV-franchise)	ja	ja	nei	ja	nei	nei	nei	nei	nei	nei	0.9166666667		3	0.9166666667
The Beatles (Band)	ja	nei	nei	ja	ja	ja	ja	ja	ja	nei	0.7155328798		7	0.7551020408
										SUM	37.3865918682			36.4132396384
										Count	50			50
										MAP	**0.7477318374**	**BPREF**		0.7282647928
														0.0040098608

Appendix J Precision Norwegian search in English Engine

Queries	Document Number										P@10
	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	
Reddit (Sosialt nettsed)	ja	nei	nei	nei	nei	nei	ja	nei	nei	nei	0.2
Tottenham (Football team)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1
Facebook (Social network site)	nei	ja	ja	ja	ja	ja	ja	ja	ja	ja	0.9
Irland (Geographical area)	ja	ja	ja	ja	ja	ja	nei	ja	ja	ja	0.9
Vin (alchoholic beverage)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1
SAS (Norwegian Airline)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1
Fotball (Sport)	ja	ja	ja	ja	ja	ja	nei	nei	ja	ja	0.8
Radioresepsjonen(Radioshow)	ja	nei	nei	nei	nei	nei	nei	nei	nei	nei	0.1
Klubb (Team)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0
Agder (Geographical Area)	ja	nei	ja	nei	ja	ja	nei	nei	nei	nei	0.4
Instrumenter (Musical instruments)	ja	ja	nei	nei	nei	nei	nei	nei	nei	nei	0.2
Hitler (Adolf Hitler	nei	nei	ja	ja	nei	nei	nei	nei	nei	ja	0.3
Gevær (Rifle)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0
Sitrus (Citrus Fruits)	nei	nei	nei	nei	nei	ja	ja	nei	nei	nei	0.2
Væske (Liquid)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0
Frukt (Fruits)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0
Magnus Carlsen (Chess Player)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1
India (Geographical Entity)	ja	ja	ja	ja	nei	ja	ja	ja	ja	nei	0.8
Scandinavian Star (Accident)	ja	ja	ja	ja	ja	ja	ja	ja	ja	nei	0.9
Gitar (Musical instrument)	nei	nei	ja	nei	nei	ja	nei	nei	nei	nei	0.2
Kollektivtransport (public transport)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1
Iphone (Mobile Phone)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1
Eurasia (Geographical Area)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0
Depresjon (Psychological illness)	ja	ja	ja	ja	ja	ja	nei	nei	nei	nei	0.6
Kalashnikov (Weapon)	nei	nei	nei	nei	ja	ja	nei	nei	nei	nei	0.2
Sushi (Food)	ja	ja	nei	ja	ja	nei	ja	nei	nei	nei	0.5
Star Wars (Movie Franchise)	ja	ja	ja	ja	nei	ja	nei	nei	ja	nei	0.6
Deepwater Horizon (Accident in the gulf of Mexico)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1
Führer (German title for leader, often assoicated with Hitler)	ja	nei	ja	nei	nei	nei	nei	nei	nei	nei	0.3
007 (James Bond)	ja	ja	ja	ja	nei	nei	nei	nei	nei	nei	0.4
USSR (Older name for Sovjet Union or Russia)	nei	nei	nei	nei	nei	nei	ja	nei	nei	nei	0.1
Newton (Sir Isaac Newton)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0
Napoli (Geographical Area)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0
Røde Kors (Red Cross organization)	ja	ja	ja	nei	ja	ja	ja	nei	nei	ja	0.7
Steve Jobs (Founder of Apple Inc)	ja	ja	ja	ja	ja	ja	ja	ja	nei	ja	0.9
Toyota (Car maker)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1
Tiger Woods (Professional golf player)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1
Kate Middleton (Wife of prince Williams)	ja	ja	ja	nei	nei	nei	nei	nei	ja	nei	0.5
Vuvuzela (Instrument)	nei	ja	ja	nei	nei	nei	nei	nei	nei	nei	0.2
Wikileaks (Organization)	nei	nei	nei	ja	nei	nei	ja	nei	nei	nei	0.2
Aung San Suu Kyi (Nobel peace prize winner)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1
Pyongyang (Capitol)	nei	nei	nei	ja	nei	nei	nei	nei	nei	nei	0.1
Aspergers (Psychologil illness)	nei	ja	ja	nei	ja	ja	nei	nei	nei	nei	0.4
Richard Nixon (Earlier president of USA)	nei	nei	nei	ja	nei	nei	nei	nei	nei	nei	0.1
Spionasje (Espionage – concept)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1
SS (Shutstaffel – famous part of German Army during WW2)	nei	nei	nei	nei	nei	nei	nei	nei	nei	nei	0
NATO (Alliance of nations)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1
EU (European union)	ja	ja	ja	ja	ja	ja	ja	ja	ja	ja	1
Uaktsomhet	nei	ja	ja	nei	nei	nei	nei	nei	nei	nei	0.2
The Simpsons (TV-franchise)	nei	ja	nei	nei	nei	nei	nei	nei	ja	nei	0.2
The Beatles (Band)	nei	ja	nei	ja	nei	ja	nei	nei	nei	nei	0.3

Count 0

Average **0.49804**

0.1319607843