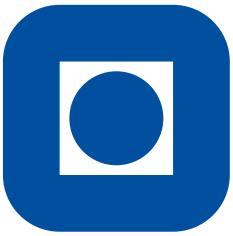
Christian Rasmussen

Finding Related Web Pages for Course Material Based on its Content

Specialization project, fall 2013

Information Systems Group
Department of Computer and Information Science
Faculty of Information Technology, Mathematics and Electrical Engineering



Abstract

This report is about finding related web pages for course material based on its contents. The course material comes from a wiki in the subject named "TDT4100 – Objektorientert programmering med Java" (TDT4100 – Object-oriented programming with Java).

By embedded related web pages into each page of the wiki, the students have easier access to useful information.

The goal of this project is implement a proof-of-concept that is able to find the related web pages (without embedding them into the wiki).

By using an information retrieval method called term frequency–inverse document frequency (*tf-idf*) I am able to find important words in a document. These words are used in a search query that searches the web.

The proof-of-concept is used to perform multiple experiments on a few selected documents. For each experiment the implementation outputs a list of links to web pages. These links are then evaluated by the author of this report as well as a committee.

The results indicate that the approach has some potential, but because the experiments were performed on well-chosen documents, the approach might not perform as well on other documents.

Preface

This is the project report for the subject "TDT4501 – Specialization project", fall 2013. The project is conducted by Christian Rasmussen who is studying Computer Science at Norwegian University of Science and Technology (NTNU).

This project was completed under the supervision of Associate Professor Hallvard Trætteberg.

A basic understanding of information retrieval is necessary to get the most out of this report.

Christian Rasmussen

Trondheim, December 20, 2013

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter describes the background and motivation for the project, followed by the project's goals, and lastly, how I will try to accomplish these goals.

1.1 Background and Motivation

This project is based on the course material from the subject "TDT4100 – Objektorientert programmering med Java" (TDT4100 – Object-oriented programming with Java). This subject has its own wiki[1], henceforth referred to as "the wiki". The contents of this wiki are written in Norwegian.

The wiki is a tool for students to learn the curriculum of the subject TDT4100. The information on the wiki is structured hierarchically and divided into four main categories:

- Object-oriented programming
- Java-programming
- Eclipse
- Procedure-oriented programming

To make the wiki more useful for the students, this project will look into ways to find related web pages for the various articles. By embedding the links in their respective article, the students get easy access to more information about the same topic. Ideally, the process of finding and embedding web pages should be done automatically by software and not manually by the authors of the wiki.

Even though this project is based on a specific wiki, the theory and methods could be applied to other wikis as well.

[1] The wiki is available at: `https://www.ntnu.no/wiki/display/tdt4100/Faginnhold`

1.2 Goals and Research Questions

As stated earlier, this project is about finding related web pages based on the content of a document.

The content on the wiki is written in Norwegian. My hypothesis is that there are more related and useful information in English. Thus, the search query has to be written in English.

Goal Implement a proof-of-concept that finds related web pages based on the content of a document.

Research question 1 Is it better to translate the text first and then find important words or the other way around?

Research question 2 Will it help to add words to the search query that are related to the whole document set?

1.3 Research Method

To research this topic I will look at common methods used in the information retrieval field.

Using these methods I will perform some coarse-grained experiments and evaluate the results myself. Later I will involve a committee to evaluate the results of the final experiments.

1.4 Thesis Structure

Chapter 2 describes the concepts that are needed to solve the problem. Chapter 3 describes the design choices and the web services/libraries used to implement a proof-of-concept. Chapter 4 describes the results from running various experiments. Chapter 5 discusses the results of the experiments.

Chapter 2

Background Theory

This chapter will cover the basic concepts and theory that are used throughout this report. The theory is based on the textbook "An Introduction to Information Retrieval" [1].

Because the problem is to find related web pages based on some text, the problem can be characterized as a search engine problem. One way to create a search engine is to use an information retrieval method called "term frequency–inverse document frequency". This is the main method that is used throughout this report.

2.1 Term Frequency

The term frequency (tf) refers to the number of times a word occur in a text document. A word that occur often are likely to be more important than words that occur less often.

2.2 Inverse Document Frequency

The inverse document frequency (idf) is used to penalize words that occur in many documents. Such words are likely to be less informative. To calculate the idf we must first find the document frequency (df), which is the number of documents a word occur in. The formula for calculating idf, where N is the total number of documents, is as follows:

$$idf = log_{10}\left(\frac{N}{df}\right)$$

The log_{10} function is used to dampen the effect of the penalty from idf.

2.3 TF-IDF

The term frequency–inverse document frequency (tf-idf) is calculated as follows:

$$tf\text{-}idf = tf \times idf$$

The *tf-idf* is used to give a score to every word in a document set. A high *tf-idf* indicate that the current word is important in the current document. This measure can then be used to find the most important words in a document.

2.4 Hierarchical Structure

The *tf-idf* method is used to find words that separate one documents from the other documents in a document set. By using only the *idf*, you are able to find the words that are common for the whole document set.

This idea could be extended onto a hierarchical structure, where each node is a document. By applying the method to a node and its child nodes you will find the words that separate one node from the other nodes, as well as the words that are common among them. The root node will include the words that are common for all nodes. Figure 2.1 shows the hierarchical structure of the wiki and the web. The figure also shows how the wiki is just any node on the web.

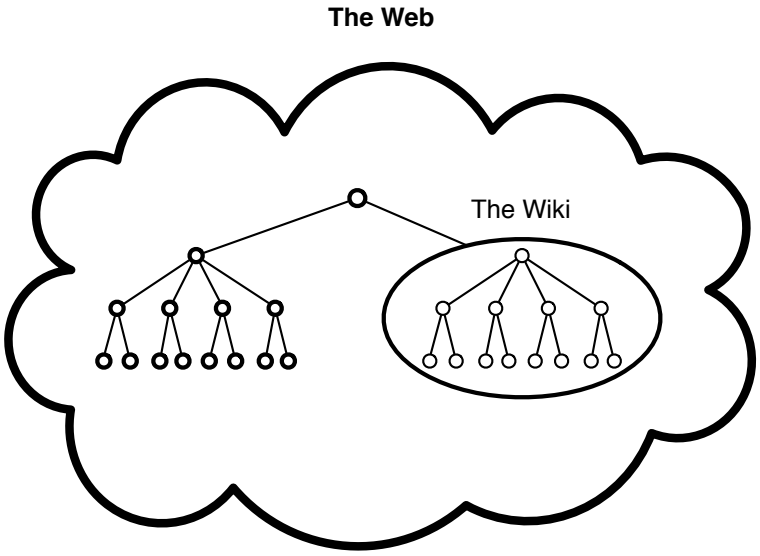


Figure 2.1: The hierarchical structure of the wiki and the web.

Note: In this report I consider the whole document set as a flat structure. Because of this I am only able find the words that are specific to a document and the words that are common for all documents.

Chapter 3

Implementation

The wiki uses Confluence as its platform which requires plugins to be written in Java, or at least a language that runs on the Java Virtual Machine (JVM) [2]. The implementation[1] is written in Clojure which runs on the JVM [3]. Because Clojure runs on the JVM, it is able to call Java-code and vice versa.

Figure 3.1 shows the steps involed in training the system. The implementation supports two different approaches to train the system; *translate-first* and *index-first*.

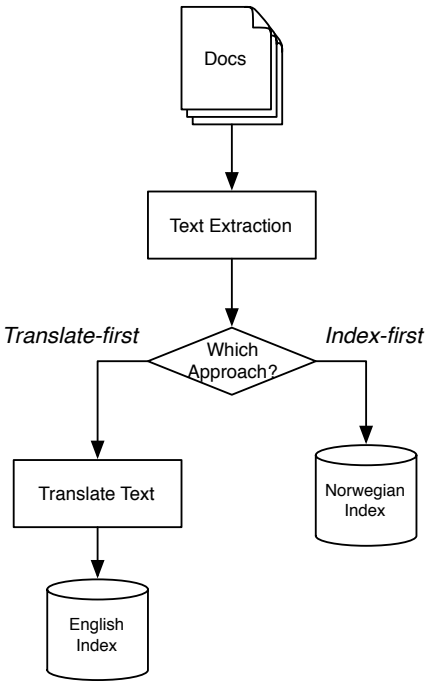


Figure 3.1: Stage 1 – Training the system.

[1]Implementation available at: `https://github.com/chrrasmussen/NTNU-Pre-Project`

Figure 3.2 shows the steps involved in retrieving the related web pages for a specific document.

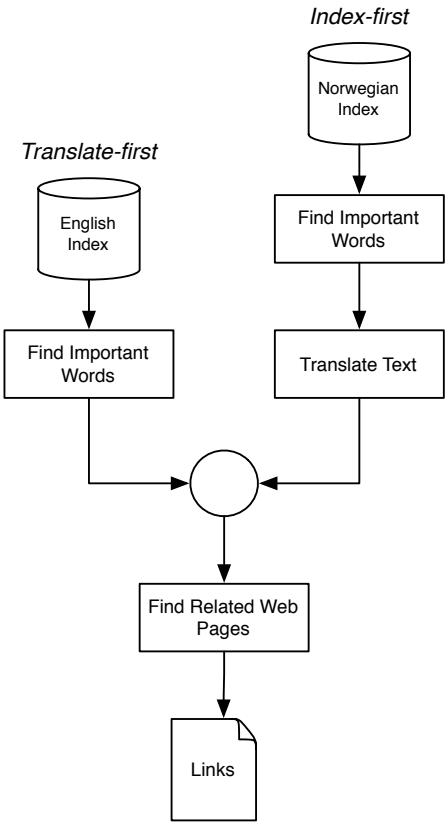


Figure 3.2: Stage 2 – Retrieve related web pages.

3.1 Text Extraction

Because the source code for the pages on the wiki are structured as XML, I must perform text extraction to get only the main content from the pages. Some pages embeds examples and charts that should be removed. These examples and charts are placed inside an element similar to: `<ac:macro ac:name="code"></ac:maro>`.

My first attempt to solve the text extraction step, was to use a Java library called "Apache Tika". It promises to be able to "extract metadata and structured text content from various documents using existing parser libraries" [4].

Unfortunately, Tika turned out to be too inflexible for the task. It could only be configured to remove specific elements by name, e.g. `ac:macro`. The source code for the wiki pages also include macros that should not be filtered, e.g. `<ac:macro ac:name="excerpt"></ac:macro>`.

To solve the problem I turned to another Java library called "jsoup". This library is specialized in manipulating and extracting data from HTML documents [5].

3.2 Translation

The next step is to translate the text from Norwegian to English. The implementation uses a web service called Google Translate API [6]. To use Google Translate you need to register a Google account.

The web service does not require any configurations.

3.3 Finding Important Words

One way to find the most important words in a document is to calculate the *tf-idf* for every word in every document. Apache Lucene is a Java library that is able to do this. However, I elected to use Apache Solr instead. Solr is a web service built on top of Lucene [7].

Solr includes some example projects and it is easy to get up and running. This project used the bundled `schema.xml` file as the starting point. To enable the calculations of *tf-idf* for the indexed text, I added `termVectors="true"` to the field named "text".

3.4 Finding Related Web Pages

The last step is to find related web pages based on a query. The implementation uses a web service called Google Custom Search API [8]. To use Google Custom Search you need to register a Google account.

To use the web service you also need to configure a Custom Search Engine (CSE) for your account[2]. The CSE should be configured to search the whole web and not just some specific web pages.

[2]Control panel to add Custom Search Engines is available at: `https://www.google.com/cse/`

Chapter 4

Experiments and Results

This chapter describes the experiments that was performed in the course of this project. The results will be discussed in more details in chapter 5.

4.1 Experimental Plan

The experiments are structured to help answer the research questions (See section 1.2).

The procedure will be as follows:

1. Select a few documents as sampling tests.
2. For each document:
    - Get the top 5 words from the Lucene index.
    - Apply the search queries using Google Search.
    - Evaluate the resulting web pages.

For each iteration, two simultaneous experiments will be performed using separate Lucene indexes:

1. *Translate-first* – Documents are translated into English and then inserted into the Lucene index. Then the important words are retrieved directly from the Lucene index.
2. *Index-first* – The Norwegian documents are inserted into the Lucene index and only the important words are translated into English.

The first itereation will be evaluated only by the author, while the second iteration will be evaluated by both the author and a committee[1]. Each link should be assigned a rating depending on how useful the web page is from the perspective of a student. The ratings are as follows:

- 0 – Not useful
- 1 – Somewhat useful

[1] The committee consists of Hallvard Trætteberg.

- 2 – Useful
- 3 – Very useful

4.2 Experimental Setup

The following items are required to conduct the experiments:

- The source code for this project.[2]
- The documents from the wiki.
- A Google Account to get access to Google Custom Search API and Google Translate API.
- An installation of Apache Solr.

All experiments are based on 80 documents from the wiki, collected at 2013-11-28. About 30 of these documents are more or less empty (they are placeholders for future content). These documents are still included in the experiments but should not make any significant difference.

4.3 Experimental Results – Iteration 1

To increase the likelihood for getting good results, I will choose documents that have a high average *tf-idf*. To make it easier to compare the results I will use the same five documents for every type of experiment. Thus I should choose documents that have a high average *tf-idf* in both the *translate-first* document set and the *index-first* document set.

Table 4.1 lists the top 10 documents from the *translate-first* document set, sorted by average *tf-idf* value.

Document	Avg. tf-idf	Top Word (tf-idf)	Word Count
JavaFX	16.70	node (33.00)	1146
Tråder med java	6.20	taxi (9.50)	516
Swing	4.84	jframe (7.00)	705
Tall og beregninger	4.80	infinity (7.00)	747
Koding av valideringsmetoder	4.37	validation (13.00)	395
Swing Timer	4.20	timer (10.00)	371
Import av kode med lim inn-funksjonen	4.00	paste (4.00)	305
Bruk av debuggeren i Eclipse	3.40	debug (5.00)	575
Klasser i java	3.31	song (4.33)	803
Anonymeklasser	3.20	inline (5.00)	383

Table 4.1: Top 10 documents (by avg. tf-idf) from the *translate-first* document set.

Table 4.2 lists the top 10 documents from the *index-first* document set, sorted by average *tf-idf* value.

[2]Implementation available at: `https://github.com/chrrasmussen/NTNU-Pre-Project`

Document	Avg. tf-idf	Top Word (tf-idf)	Word Count
JavaFX	12.00	node (15.00)	1108
Memory-eksempel versjon 1	4.80	hovedprogram (5.00)	1035
Swing	4.80	jpanel (6.00)	670
Tråder med java	4.70	tråd (5.00)	555
Kontrollstrukturer	4.40	løkker (5.00)	701
Bruk av debuggeren i Eclipse	4.40	step (8.00)	574
Swing Timer	4.00	timeren (8.00)	371
Synlighetsmodifikatorer	3.90	ressursen (6.00)	302
Anonymeklasser	3.80	anonym (6.00)	380
Tall og beregninger	3.80	infinity (5.00)	732

Table 4.2: Top 10 documents (by avg. tf-idf) from the *index-first* document set.

The documents with the highest average *tf-idf* in both table 4.1 and table 4.2 are:

- JavaFX
- Tråder med java
- Swing
- Bruk av debuggeren i Eclipse
- Tall og beregninger

4.3.1 Translate-first Experiments

Finding Related Web Pages for Document "JavaFX"

Table 4.3 lists the top 5 words.

Word	tf-idf
node	33.00
scene	20.00
nodes	14.00
graph	9.50
style	7.00

Table 4.3: Top 5 words from the document "JavaFX" in the *translate-first* document set.

The search results based on the query "node scene nodes graph style":

1. `Node (JavaFX 2.2)`
   `http://docs.oracle.com/javafx/2/api/javafx/scene/Node.html`
2. `Scene graph - Wikipedia, the free encyclopedia`
   `http://en.wikipedia.org/wiki/Scene_graph`
3. `javafx.scene (JavaFX 2.2)`
   `http://docs.oracle.com/javafx/2/api/javafx/scene/package-summary.html`

4. Chapter 3. Nodes and Groups
   http://techpubs.sgi.com/library/dynaweb_docs/0620/SGI_Developer/books/Inv_Mentor/sgi_html/ch03.html
5. Support/Tutorials/FindingNodes { osg
   http://www.openscenegraph.org/projects/osg/wiki/Support/Tutorials/FindingNodes

Table 4.4 shows the evaluation of each link.

#	Rating (by Author)
1	2 – Useful
2	1 – Somewhat useful
3	2 – Useful
4	1 – Somewhat useful
5	1 – Somewhat useful
Average	1.4

Table 4.4: Ratings of the links for the document "JavaFX" in the *translate-first* document set.

Finding Related Web Pages for Document "Tråder med java"

Table 4.5 lists the top 5 words.

Word	tf-idf
taxi	9.50
thread	7.00
process	5.50
driver	5.00
client	4.00

Table 4.5: Top 5 words from the document "Tråder med java" in the *translate-first* document set.

The search results based on the query "taxi thread process driver client":

1. Shadow Market For Taxi Permits Lucrative For Some, Hardship For ...
   http://www.kpbs.org/news/2013/jun/10/shadow-market-taxi-permits-lucrative-some-hardship/
2. Uber phone app may be shifting incentives for taxi and livery drivers ...
   http://www.wbez.org/news/uber-car-service-app-makes-winners-and-losers-104544
3. Boston Taxis: Why Some Drivers Say Credit Cards Machines Are ...
   http://bostinno.streetwise.co/2013/01/28/boston-taxi-credit-card-machine-broken/
4. Drivers: Uber Is Skimming Our Tips | Mother Jones
   http://www.motherjones.com/politics/2013/03/uber-drivers-strike-tips-wages-class-action
5. HMRC targets taxi drivers...again | AccountingWEB
   http://www.accountingweb.co.uk/anyanswers/question/hmrc-targets-taxi-driversagain

Table 4.6 shows the evaluation of each link.

#	Rating (by Author)
1	0 – Not useful
2	0 – Not useful
3	0 – Not useful
4	0 – Not useful
5	0 – Not useful
Average	0.0

Table 4.6: Ratings of the links for the document "Tråder med java" in the *translate-first* document set.

Finding Related Web Pages for Document "Swing"

Table 4.7 lists the top 5 words.

Word	tf-idf
jframe	7.00
jpanel	6.00
component	5.00
panel	3.20
listener	3.00

Table 4.7: Top 5 words from the document "Swing" in the *translate-first* document set.

The search results based on the query "jframe jpanel component panel listener":

1. `How to Write a Component Listener (The Java™ Tutorials ...`
   `http://docs.oracle.com/javase/tutorial/uiswing/events/componentlistener.html`

2. `Java JPanel mouse listener doesn't work over its components`
   `http://stackoverflow.com/questions/9794765/java-jpanel-mouse-listener-doesnt-work-over-its-components`

3. `Javanotes 6.0, Section 6.1 -- The Basic GUI Application`
   `http://math.hws.edu/javanotes/c6/s1.html`

4. `java - Listening/Handling JPanel events - Stack Overflow`
   `http://stackoverflow.com/questions/10051176/listening-handling-jpanel-events`

5. `How to Write a Component Listener`
   `http://www.math.uni-hamburg.de/doc/java/tutorial/uiswing/events/componentlistener.html`

Table 4.8 shows the evaluation of each link.

#	Rating (by Author)
1	3 – Very useful
2	2 – Useful
3	3 – Very useful
4	2 – Useful
5	2 – Useful
Average	2.4

Table 4.8: Ratings of the links for the document "Swing" in the *translate-first* document set.

Finding Related Web Pages for Document "Bruk av debuggeren i Eclipse"

Table 4.9 lists the top 5 words.

Word	tf-idf
debug	5.00
breakpoint	4.00
stopped	3.00
step	3.00
perspective	2.00

Table 4.9: Top 5 words from the document "Bruk av debuggeren i Eclipse" in the *translate-first* document set.

The search results based on the query "debug breakpoint stopped step perspective":

1. `Java Debugging with Eclipse - Tutorial`
   `http://www.vogella.com/articles/EclipseDebugging/article.html`

2. `Debugger`
   `http://pydev.org/manual_adv_debugger.html`

3. `Comp310: Debugging in Eclipse`
   `http://www.clear.rice.edu/comp310/Eclipse/debugging.html`

4. `Effective Java Debugging with Eclipse ≪ EclipseSource Blog`
   `http://eclipsesource.com/blogs/2013/01/08/effective-java-debugging-with-eclipse/`

5. `About the Debug perspective`
   `http://livedocs.adobe.com/coldfusion/8/usingdebugger_5.html`

Table 4.10 shows the evaluation of each link.

#	Rating (by Author)
1	3 – Very useful
2	1 – Somewhat useful
3	3 – Very useful
4	3 – Very useful
5	0 – Not useful
Average	2.0

Table 4.10: Ratings of the links for the document "Bruk av debuggeren i Eclipse" in the *translate-first* document set.

Finding Related Web Pages for Document "Tall og beregninger"

Table 4.11 lists the top 5 words.

Word	tf-idf
infinity	7.00
range	5.00
wrapper	5.00
numeric	3.50
floating	3.50

Table 4.11: Top 5 words from the document "Tall og beregninger" in the *translate-first* document set.

The search results based on the query " infinity range wrapper numeric floating":

1. `Numeric Datatypes (XML Schema)`
   `http://docstore.mik.ua/orelly/xml/schema/ch04_04.htm`

2. `2.2 Floating point values`
   `http://www.cs.rit.edu/~ats/java-2005-2/2.2.html`

3. `How to implement infinity in Java? - Stack Overflow`
   `http://stackoverflow.com/questions/12952024/how-to-implement-infinity-in-java`

4. `Chapter 5. Conversions and Promotions`
   `http://docs.oracle.com/javase/specs/jls/se7/html/jls-5.html`

5. `JavaScript Number Object`
   `http://www.w3schools.com/jsref/jsref_obj_number.asp`

Table 4.12 shows the evaluation of each link.

#	Rating (by Author)
1	1 – Somewhat useful
2	2 – Useful
3	1 – Somewhat useful
4	3 – Very useful
5	0 – Not useful
Average	1.4

Table 4.12: Ratings of the links for the document "Tall og beregninger" in the *translate-first* document set.

4.3.2 Index-first Experiments

Finding Related Web Pages for Document "JavaFX"

Table 4.13 lists the top 5 words.

Word	tf-idf
node	15.00
scene	14.00
noder	11.00
graph	11.00
id	9.00

Table 4.13: Top 5 words from the document "JavaFX" in the *index-first* document set.

The search results based on the query "node scene nodes graph id":

1. `Node (JavaFX 2.2)`
   `http://docs.oracle.com/javafx/2/api/javafx/scene/Node.html`

2. `Irrlicht 3D Engine: irr::scene::ISceneManager Class Reference`
   `http://irrlicht.sourceforge.net/docu/classirr_1_1scene_1_1_i_scene_manager.html`

3. `Ogre::SceneNode Class Reference - OGRE Documentation`
   `http://www.ogre3d.org/docs/api/html/classOgre_1_1SceneNode.html`

4. `Irrlicht 3D Engine: Tutorial 3: Custom SceneNode`
   `http://irrlicht.sourceforge.net/docu/example003.html`

5. `Lesson 2: The Scene Graph and Nodes`
   `http://docs.autodesk.com/3DSMAX/15/ENU/3ds-Max-SDK-Programmer-Guide/files/GUID-DCD280C9-36B4-4053-9A7F-DF03CA4F41F0.htm`

Table 4.14 shows the evaluation of each link.

#	Rating (by Author)
1	2 – Useful
2	0 – Not useful
3	0 – Not useful
4	0 – Not useful
5	0 – Not useful
Average	0.4

Table 4.14: Ratings of the links for the document "JavaFX" in the *index-first* document set.

Finding Related Web Pages for Document "Tråder med java"

Table 4.15 lists the top 5 words.

Word	tf-idf
tråd	5.00
taxisjåføren	5.00
tråden	5.00
taxi	4.50
kunder	4.00

Table 4.15: Top 5 words from the document "Tråder med java" in the *index-first* document set.

The search results based on the query "Thread taxi driver thread taxi customers":

1. `/vp/ - Pokémon ≫ Thread #14770771`
   `https://archive.foolz.us/vp/thread/14770771`

2. `I am a 26 year old NYC yellow cab driver. Ask me anything : IAmA`
   `http://www.reddit.com/r/IAmA/comments/19g9mi/i_am_a_26_year_old_nyc_yellow_cab_driver_ask_me/`

3. `Boston should thread rightly in changing taxi system - Boston.com`
   `http://www.boston.com/2013/08/28/taxipodium/rkvcjky4As9Gl4nqXUvGmM/story.html`

4. `Reddit Thread Of The Week: 'Scarface,' 'Taxi Driver,' And 'Harry ...`
   `http://news.moviefone.com/2012/03/30/reddit-thread-of-the-week/`

5. `taxi driver's monopolies - WordReference Forums`
   `http://forum.wordreference.com/showthread.php?t=2741683`

Table 4.16 shows the evaluation of each link.

#	Rating (by Author)
1	0 – Not useful
2	0 – Not useful
3	0 – Not useful
4	0 – Not useful
5	0 – Not useful
Average	0.0

Table 4.16: Ratings of the links for the document "Tråder med java" in the *index-first* document set.

Finding Related Web Pages for Document "Swing"

Table 4.17 lists the top 5 words.

Word	tf-idf
jpanel	6.00
jframe	6.00
lytteren	4.00
hendelsen	4.00
fyller	4.00

Table 4.17: Top 5 words from the document "Swing" in the *index-first* document set.

The search results based on the query "JPanel JFrame listener event fill":

1. `java - JButtons inside JPanels, fill up the whole panel - Stack Overflow`
   `http://stackoverflow.com/questions/14550255/jbuttons-inside-jpanels-fill-up-the-whole-panel`

2. `Javanotes 6.0, Section 6.1 -- The Basic GUI Application`
   `http://math.hws.edu/javanotes/c6/s1.html`

3. `java - Parent JPanel - How to listen to the events generated by ...`
   `http://stackoverflow.com/questions/15753536/parent-jpanel-how-to-listen-to-the-events-generated-by-components-of-a-child-j`

4. `Draw a filled in circle when the mouse is clicked | DaniWeb`
   `http://www.daniweb.com/software-development/java/threads/200752/draw-a-filled-in-circle-when-the-mouse-is-clicked`

5. `java - Combining JFrame and JPanel - Stack Overflow`
   `http://stackoverflow.com/questions/15975438/combining-jframe-and-jpanel`

Table 4.18 shows the evaluation of each link.

#	Rating (by Author)
1	1 – Somewhat useful
2	3 – Very useful
3	2 – Useful
4	2 – Useful
5	2 – Useful
Average	2.0

Table 4.18: Ratings of the links for the document "Swing" in the *index-first* document set.

Finding Related Web Pages for Document "Bruk av debuggeren i Eclipse"

Table 4.19 lists the top 5 words.

Word	tf-idf
step	8.00
breakpoint	4.00
fortsette	4.00
stoppet	3.00
breakpoints	3.00

Table 4.19: Top 5 words from the document "Bruk av debuggeren i Eclipse" in the *index-first* document set.

The search results based on the query "step breakpoint continue stop Breakpoints":

1. `Debugging with GDB - Stopping and Continuing`
   `http://www.chemie.fu-berlin.de/chemnet/use/info/gdb/gdb_6.html`

2. `Debugging JavaScript - Chrome DevTools | Google Developers`
   `https://developers.google.com/chrome-developer-tools/docs/javascript-debugging`

3. `Continuing and Stepping - Debugging with GDB`
   `http://sourceware.org/gdb/onlinedocs/gdb/Continuing-and-Stepping.html`

4. `26.2. pdb - The Python Debugger - Python v2.7.6 documentation`
   `http://docs.python.org/library/pdb.html`

5. `Debugging Process and Features - MATLAB & Simulink`
   `http://www.mathworks.com/help/matlab/matlab_prog/debugging-process-and-features.html`

Table 4.20 shows the evaluation of each link.

#	Rating (by Author)
1	1 – Somewhat useful
2	0 – Not useful
3	0 – Not useful
4	0 – Not useful
5	0 – Not useful
Average	0.2

Table 4.20: Ratings of the links for the document "Bruk av debuggeren i Eclipse" in the *index-first* document set.

Finding Related Web Pages for Document "Tall og beregninger"

Table 4.21 lists the top 5 words.

Word	tf-idf
infinity	5.00
verdiområdet	4.00
talltyper	4.00
wrapper	3.00
flyttall	3.00

Table 4.21: Top 5 words from the document "Tall og beregninger" in the *index-first* document set.

The search results based on the query " infinity range numeric types wrapper floating point":

1. `Primitive Data Types (Java in a Nutshell)`
   `http://docstore.mik.ua/orelly/java-ent/jnut/ch02_04.htm`

2. `Chapter 5. Conversions and Promotions`
   `http://docs.oracle.com/javase/specs/jls/se7/html/jls-5.html`

3. `Infinity - Rosetta Code`
   `http://rosettacode.org/wiki/Infinity`

4. `Java: Floating-point`
   `http://www.leepoint.net/notes-java/data/basic_types/22floatingpoint.html`

5. `Floating point math issues - Geos-chem`
   `http://wiki.seas.harvard.edu/geos-chem/index.php/Floating_point_math_issues`

Table 4.22 shows the evaluation of each link.

#	Rating (by Author)
1	3 – Very useful
2	3 – Very useful
3	1 – Somewhat useful
4	1 – Somewhat useful
5	0 – Not useful
Average	1.6

Table 4.22: Ratings of the links for the document "Tall og beregninger" in the *index-first* document set.

4.4 Experimental Results – Iteration 2

As described in section 2.4, this wiki is just any node on the web. Because of this, it could be helpful to find the words that differentiates the wiki from the rest of the web.

Because it is not feasible to create an index that contains the whole web and the wiki, I need to do an approximation. One way is to find the most common words in the current document set and filter out all the stop words.

Table 4.23 lists the top 50 mosts commons words in *translate-first* document set. The document set contains 2374 distinct words.

x10	1	2	3	4	5	6	7	8	9	10
0	and	the	to	in	of	is	as	that	a	can
1	for	be	with	are	this	it	or	an	if	code
2	not	will	one	quot	class	use	you	example	method	on
3	used	but	also	have	when	by	methods	see	more	has
4	object	what	about	java	so	which	using	classes	from	there

Table 4.23: Top 50 most common words in the *translate-first* document set.

Table 4.24 lists the top 50 mosts commons words in *index-first* document set. The document set contains 3577 distinct words.

x10	1	2	3	4	5	6	7	8	9	10
0	er	og	som	en	i	å	for	det	til	på
1	av	kan	med	et	at	dette	har	eller	om	den
2	ikke	vil	skal	de	denne	også	være	men	brukes	ved
3	man	så	når	fra	dersom	koden	se	kode	java	slik
4	vi	over	f.eks	mer	metoder	alle	eksempel	gjøre	disse	objekter

Table 4.24: Top 50 most common words in the *index-first* document set.

After running all the words from table 4.23 and table 4.24 through a filter that removes the stop words I was left with too many words. I decided to manually remove words that do not carry any meaningful information. I ended up with the word "java", which is very specific to

document set as a whole. Thus, in the following experiments I will add the word "java" to the search queries.

4.4.1 Translate-first Experiments

Finding Related Web Pages for Document "JavaFX"

Table 4.25 lists the top 5 words.

Word	tf-idf
node	33.00
scene	20.00
nodes	14.00
graph	9.50
style	7.00

Table 4.25: Top 5 words from the document "JavaFX" in the *translate-first* document set.

The search results based on the query "java node scene nodes graph style":

1. `Node (JavaFX 2.2)`
   `http://docs.oracle.com/javafx/2/api/javafx/scene/Node.html`
2. `javafx.scene (JavaFX 8)`
   `http://download.java.net/jdk8/jfxdocs/javafx/scene/package-summary.html`
3. `JavaFX Architecture` JavaFX 2 Tutorials and Documentation—
   `http://docs.oracle.com/javafx/2/architecture/jfxpub-architecture.htm`
4. `Creating Java User Interfaces with Project Scene Graph` Hello ...—
   `http://www.informit.com/articles/article.aspx?p=1323245`
5. `JavaFX CSS Reference Guide`
   `http://docs.oracle.com/javafx/2/api/javafx/scene/doc-files/cssref.html`

Table 4.26 shows the evaluation of each link.

#	Rating (by Author)	Rating (by Committee)
1	2 – Useful	2 – Useful
2	2 – Useful	2 – Useful
3	3 – Very useful	3 – Very useful
4	3 – Very useful	2 – Useful
5	2 – Useful	2 – Useful
Average	2.4	2.2

Table 4.26: Ratings of the links for the document "JavaFX" in the *translate-first* document set.

Finding Related Web Pages for Document "Tråder med java"

Table 4.27 lists the top 5 words.

Word	tf-idf
taxi	9.50
thread	7.00
process	5.50
driver	5.00
client	4.00

Table 4.27: Top 5 words from the document "Tråder med java" in the *translate-first* document set.

The search results based on the query "java taxi thread process driver client":

1. `C* Summit 2013: Java and .NET Client Drivers - Cassandra ...`
   `http://www.slideshare.net/planetcassandra/cassandra-summit-data-stax-java-driver`

2. `Observer pattern - Wikipedia, the free encyclopedia`
   `http://en.wikipedia.org/wiki/Observer_pattern`

3. `Thread: Runtime Error 216 when using explorer.exe and Spybot`
   `http://forums.spybot.info/showthread.php?69832-Runtime-Error-216-when-using-explorer-exe-and-Spybot`

4. `Compiled Technical Documentation`
   `https://courses.cs.washington.edu/courses/cse403/02su/www/internal/docs/TechnicalDocumentationv8.doc`

5. `Oscar_Delta Toolbar`
   `http://forums.spybot.info/showthread.php?69789-Oscar_Delta-Toolbar`

Table 4.28 shows the evaluation of each link.

#	Rating (by Author)	Rating (by Committee)
1	1 – Somewhat useful	0 – Not useful
2	0 – Not useful	0 – Not useful
3	0 – Not useful	0 – Not useful
4	0 – Not useful	0 – Not useful
5	0 – Not useful	0 – Not useful
Average	0.2	0.0

Table 4.28: Ratings of the links for the document "Tråder med java" in the *translate-first* document set.

Finding Related Web Pages for Document "Swing"

Table 4.29 lists the top 5 words.

Word	tf-idf
jframe	7.00
jpanel	6.00
component	5.00
panel	3.20
listener	3.00

Table 4.29: Top 5 words from the document "Swing" in the *translate-first* document set.

The search results based on the query "java jframe jpanel component panel listener":

1. `How to Write a Component Listener (The Java™ Tutorials ...`
   `http://docs.oracle.com/javase/tutorial/uiswing/events/componentlistener.html`

2. `Java JPanel mouse listener doesn't work over its components`
   `http://stackoverflow.com/questions/9794765/java-jpanel-mouse-listener-doesnt-work-over-its-components`

3. `Java Swing first programs`
   `http://zetcode.com/tutorials/javaswingtutorial/firstprograms/`

4. `java - How to resize components related to panel in swing? - Stack ...`
   `http://stackoverflow.com/questions/10784104/how-to-resize-components-related-to-panel-in-swing`

5. `SWING ComponentListener Interface`
   `http://www.tutorialspoint.com/swing/swing_component_listener.htm`

Table 4.30 shows the evaluation of each link.

#	Rating (by Author)	Rating (by Committee)
1	2 – Useful	2 – Useful
2	1 – Somewhat useful	1 – Somewhat Useful
3	3 – Very useful	2 – Useful
4	1 – Somewhat useful	1 – Somewhat useful
5	2 – Useful	1 – Somewhat useful
Average	1.8	1.4

Table 4.30: Ratings of the links for the document "Swing" in the *translate-first* document set.

Finding Related Web Pages for Document "Bruk av debuggeren i Eclipse"

Table 4.31 lists the top 5 words.

Word	tf-idf
debug	5.00
breakpoint	4.00
stopped	3.00
step	3.00
perspective	2.00

Table 4.31: Top 5 words from the document "Bruk av debuggeren i Eclipse" in the *translate-first* document set.

The search results based on the query "java debug breakpoint stopped step perspective":

1. `Java Debugging with Eclipse - Tutorial`
   `http://www.vogella.com/articles/EclipseDebugging/article.html`

2. `Effective Java Debugging with Eclipse « EclipseSource Blog`
   `http://eclipsesource.com/blogs/2013/01/08/effective-java-debugging-with-eclipse/`

3. `Stepping through the execution of a Java program`
   `http://help.eclipse.org/indigo/topic/org.eclipse.jdt.doc.user/tasks/task-stepping.htm`

4. `About the Debug perspective`
   `http://livedocs.adobe.com/coldfusion/8/usingdebugger_5.html`

5. `Debugging your programs`
   `http://help.eclipse.org/juno/topic/org.eclipse.jdt.doc.user/gettingStarted/qs-13.htm`

Table 4.32 shows the evaluation of each link.

#	Rating (by Author)	Rating (by Committee)
1	3 – Very useful	3 – Very useful
2	3 – Very useful	3 – Very useful
3	3 – Very useful	3 – Very useful
4	0 – Not useful	0 – Not useful
5	3 – Very useful	3 – Very useful
Average	2.4	2.4

Table 4.32: Ratings of the links for the document "Bruk av debuggeren i Eclipse" in the *translate-first* document set.

Finding Related Web Pages for Document "Tall og beregninger"

Table 4.33 lists the top 5 words.

Word	tf-idf
infinity	7.00
range	5.00
wrapper	5.00
numeric	3.50
floating	3.50

Table 4.33: Top 5 words from the document "Tall og beregninger" in the *translate-first* document set.

The search results based on the query "java infinity range wrapper numeric floating":

1. `2.2 Floating point values`
   `http://www.cs.rit.edu/~ats/java-2005-2/2.2.html`

2. `How to implement infinity in Java? - Stack Overflow`
   `http://stackoverflow.com/questions/12952024/how-to-implement-infinity-in-java`

3. `Chapter 5. Conversions and Promotions`
   `http://docs.oracle.com/javase/specs/jls/se7/html/jls-5.html`

4. `Class java.lang.Double`
   `http://www.geom.uiuc.edu/~daeron/docs/apidocs/java.lang.Double.html`

5. `Primitive Data Types (Java in a Nutshell)`
   `http://docstore.mik.ua/orelly/java-ent/jnut/ch02_04.htm`

Table 4.34 shows the evaluation of each link.

#	Rating (by Author)	Rating (by Committee)
1	2 – Useful	2 – Useful
2	1 – Somewhat useful	1 – Somewhat useful
3	2 – Useful	2 – Useful
4	2 – Useful	2 – Useful
5	3 – Very useful	2 – Useful
Average	2.0	1.8

Table 4.34: Ratings of the links for the document "Tall og beregninger" in the *translate-first* document set.

4.4.2 Index-first Experiments

Finding Related Web Pages for Document "JavaFX"

Table 4.35 lists the top 5 words.

Word	tf-idf
node	15.00
scene	14.00
noder	11.00
graph	11.00
id	9.00

Table 4.35: Top 5 words from the document "JavaFX" in the *index-first* document set.

The search results based on the query "java node scene nodes graph id":

1. `Node (JavaFX 2.2)`
   `http://docs.oracle.com/javafx/2/api/javafx/scene/Node.html`
2. `javafx.scene (JavaFX 8)`
   `http://download.java.net/jdk8/jfxdocs/javafx/scene/package-summary.html`
3. `Parent (JavaFX 2.2)`
   `http://docs.oracle.com/javafx/2/api/javafx/scene/Parent.html`
4. `CyberVRML97ForJava < Main < CyberGarage`
   `http://www.cybergarage.org/twiki/bin/view/Main/CyberVRML97ForJava`
5. `Working with the JavaFX Scene Graph | JavaFX 2 Tutorials and ...`
   `http://docs.oracle.com/javafx/2/scenegraph/jfxpub-scenegraph.htm`

Table 4.36 shows the evaluation of each link.

#	Rating (by Author)	Rating (by Committee)
1	2 – Useful	2 – Useful
2	2 – Useful	2 – Useful
3	2 – Useful	2 – Useful
4	0 – Not useful	0 – Not useful
5	3 – Very useful	3 – Very useful
Average	1.8	1.8

Table 4.36: Ratings of the links for the document "JavaFX" in the *index-first* document set.

Finding Related Web Pages for Document "Tråder med java"

Table 4.37 lists the top 5 words.

Word	tf-idf
tråd	5.00
taxisjåføren	5.00
tråden	5.00
taxi	4.50
kunder	4.00

Table 4.37: Top 5 words from the document "Tråder med java" in the *index-first* document set.

The search results based on the query "java Thread taxi driver thread taxi customers":

1. `java - How to handle listeners if both views and models(objects ...`
   `http://stackoverflow.com/questions/11173346/`
2. `Tests Don't Run · Issue #1 · clojure-cookbook/browser-testing · GitHub`
   `https://github.com/clojure-cookbook/browser-testing/issues/1`
3. `Taxi Drivers - Pokémon X & Y Forum - Neoseeker Forums`
   `http://www.neoseeker.com/forums/60943/t1914668-taxi-drivers/`
4. `Stupid cab drivers around metropolitan Phoenix. (Tempe ...`
   `http://www.city-data.com/forum/phoenix-area/947792-stupid-cab-drivers-around-metropolitan-phoenix.html`
5. `Russian app-based taxi startup Wheely takes aim at London's ...`
   `http://www.computing.co.uk/ctg/news/2273632/russian-appbased-taxi-startup-wheely-takes-aim-at-londons-addison-lee`

Table 4.38 shows the evaluation of each link.

#	Rating (by Author)	Rating (by Committee)
1	1 – Somewhat useful	1 – Somewhat useful
2	0 – Not useful	0 – Not useful
3	0 – Not useful	0 – Not useful
4	0 – Not useful	0 – Not useful
5	0 – Not useful	0 – Not useful
Average	0.2	0.2

Table 4.38: Ratings of the links for the document "Tråder med java" in the *index-first* document set.

Finding Related Web Pages for Document "Swing"

Table 4.39 lists the top 5 words.

Word	tf-idf
jpanel	6.00
jframe	6.00
lytteren	4.00
hendelsen	4.00
fyller	4.00

Table 4.39: Top 5 words from the document "Swing" in the *index-first* document set.

The search results based on the query "java JPanel JFrame listener event fill":

1. `java - JButtons inside JPanels, fill up the whole panel - Stack Overflow`
   `http://stackoverflow.com/questions/14550255/jbuttons-inside-jpanels-fill-up-the-whole-panel`

2. Javanotes 6.0, Section 6.1 -- The Basic GUI Application
   http://math.hws.edu/javanotes/c6/s1.html
3. java - JTable update after row selection - Stack Overflow
   http://stackoverflow.com/questions/20443764/jtable-update-after-row-selection
4. JAVA SWING GUI TUTORIAL
   http://cs.nyu.edu/~yap/classes/visual/03s/lect/l7/
5. ChartPanel (JFreeChart Class Library (version 1.0.17))
   http://www.jfree.org/jfreechart/api/javadoc/org/jfree/chart/ChartPanel.html

Table 4.40 shows the evaluation of each link.

#	Rating (by Author)	Rating (by Committee)
1	2 – Useful	1 – Somewhat useful
2	3 – Very useful	3 – Very useful
3	1 – Somewhat useful	0 – Not useful
4	3 – Very useful	3 – Very useful
5	1 – Somewhat useful	0 – Not useful
Average	2.0	1.4

Table 4.40: Ratings of the links for the document "Swing" in the *index-first* document set.

Finding Related Web Pages for Document "Bruk av debuggeren i Eclipse"

Table 4.41 lists the top 5 words.

Word	tf-idf
step	8.00
breakpoint	4.00
fortsette	4.00
stoppet	3.00
breakpoints	3.00

Table 4.41: Top 5 words from the document "Bruk av debuggeren i Eclipse" in the *index-first* document set.

The search results based on the query "java step breakpoint continue stop Breakpoints":

1. Debugging JavaScript - Chrome DevTools | Google Developers
   https://developers.google.com/chrome-developer-tools/docs/javascript-debugging
2. Debugging Java programs with BlueJ
   http://www.mathcs.emory.edu/~cheung/Courses/170/Syllabus/02/BlueJ/BlueJ3.html
3. Debugging with GDB - Stopping and Continuing
   http://www.chemie.fu-berlin.de/chemnet/use/info/gdb/gdb_6.html
4. Java Debugging with Eclipse - Tutorial
   http://www.vogella.com/articles/EclipseDebugging/article.html

5. `C H A P T E R 6 - Setting Breakpoints and Traces`
   `http://docs.oracle.com/cd/E19422-01/819-3683/breakpnt.html`

Table 4.42 shows the evaluation of each link.

#	Rating (by Author)	Rating (by Committee)
1	0 – Not useful	0 – Not useful
2	2 – Useful	0 – Not useful
3	0 – Not useful	0 – Not useful
4	3 – Very useful	3 – Very useful
5	3 – Very useful	0 – Not useful
Average	1.6	0.6

Table 4.42: Ratings of the links for the document "Bruk av debuggeren i Eclipse" in the *index-first* document set.

Finding Related Web Pages for Document "Tall og beregninger"

Table 4.43 lists the top 5 words.

Word	tf-idf
infinity	5.00
verdiområdet	4.00
talltyper	4.00
wrapper	3.00
flyttall	3.00

Table 4.43: Top 5 words from the document "Tall og beregninger" in the *index-first* document set.

The search results based on the query "java infinity range numeric types wrapper floating point":

1. `Primitive Data Types (Java in a Nutshell)`
   `http://docstore.mik.ua/orelly/java-ent/jnut/ch02_04.htm`
2. `Chapter 5. Conversions and Promotions`
   `http://docs.oracle.com/javase/specs/jls/se7/html/jls-5.html`
3. `Java: Floating-point`
   `http://www.leepoint.net/notes-java/data/basic_types/22floatingpoint.html`
4. `How to implement infinity in Java? - Stack Overflow`
   `http://stackoverflow.com/questions/12952024/how-to-implement-infinity-in-java`
5. `2.2 Floating point values`
   `http://www.cs.rit.edu/~ats/java-2005-2/2.2.html`

Table 4.44 shows the evaluation of each link.

#	Rating (by Author)	Rating (by Committee)
1	3 – Very useful	2 – Useful
2	3 – Very useful	2 – Useful
3	1 – Somewhat useful	1 – Somewhat useful
4	1 – Somewhat useful	1 – Somewhat useful
5	2 – Useful	2 – Useful
Average	2.0	1.6

Table 4.44: Ratings of the links for the document "Tall og beregninger" in the *index-first* document set.

Chapter 5

Evaluation and Conclusion

This chapter concludes the report by evaluating and discussing the results from chapter 4.

5.1 Evaluation

Table 5.1 lists the average rating of each experiment.

	Iteration 1		Iteration 2	
Document	Translate-first	Index-first	Translate-first	Index-first
JavaFX	1.4	0.4	2.4 (2.2)	1.8 (1.8)
Tråder med java	0.0	0.0	0.2 (0.0)	0.2 (0.2)
Swing	2.4	2.0	1.8 (1.4)	2.0 (1.4)
Bruk av debuggeren i Eclipse	2.0	0.2	2.4 (2.4)	1.6 (0.6)
Tall og beregninger	1.4	1.6	2.0 (1.8)	2.0 (1.6)

Table 5.1: The average rating of each experiment (results in parenthesis are evaluated by committee).

Translate-first vs. Index-first

By comparing the results from the *translate-first* experiments and the results from the *index-first* experiments, we can see that the *translate-first* approach gives the highest average rating in most cases.

Iteration 1 vs. Iteration 2

By comparing the results between iteration 1 and iteration 2 we can see that the addition of the word "java" in the search query gives a higher average rating in most cases.

Anomalous Results

In table 5.1 there is one document that stands out from the rest; the document "Tråder med java". It seems like none of the resulting web pages were useful from the perspective of a student.

By looking at the results from the experiments (from section 4.3 and section 4.4) one common feature among the web pages is that they are related to the word "taxi", in the context of the profession taxi driver. This might indicate that Google Search weights the word "taxi" higher than the others words. Table 5.2 lists the search queries used in the different experiments.

Experiment	Search Query
Iteration 1, Translate-first Experiments	taxi thread process driver client
Iteration 1, Index-first Experiments	Thread taxi driver thread taxi customers
Iteration 2, Translate-first Experiments	java taxi thread process driver client
Iteration 2, Index-first Experiments	java Thread taxi driver thread taxi customers

Table 5.2: The search queries used to find related web pages for the document "Tråder med java".

But why is the word "taxi" in the search query in the first place? By looking at the content of the document "Tråder med java" (See attachment Attachment 1:) we can see that the word "taxi" occurs multiple times in an example which explains threads in Java.

5.2 Discussion

Because the documents that were selected were the ones with the highest average *tf-idf*, this approach might not work as well for documents with a lower average *tf-idf*.

Translate-first vs. Index-first

The results indicate that the *translate-first* is a better approach than *index-first* approach. There might be several reasons for this:

- In the case of *translate-first* approach, a complete document is translated at the time, which gives the translator the full context of all the words. This might give a more accurate translation of all the words.
- The documents in the wiki does not always use the Norwegian translation of the terminology in computer science. E.g. both "tråd" and "thread" are added to the index. Because of this, the *index-first* index ends up with a lot more words (3577 distinct words) than the *translate-first* index (2374 distinct words).

Iteration 1 vs. Iteration 2

The results indicate that the addition of "java" improves the results. The main reason for this is probably that the search query is more specific (includes more keywords). The keyword "java" is a word that describe the whole document set.

To further improve the algorithm, keywords that only characterize a subset of the document set could be used. For example, if the document is structured below the Eclipse-category, the word "Eclipse" could be a a keyword in the search query.

Anomalous Results

As described in section 5.1, the document "Tråder med java" did not yield any related web pages. This was mainly caused by a word ("taxi") that was accidentally picked as an important word for that document. One way to fix this is to tag a portion of the document as an example, or more fine-grained, tag specific words or phrases, e.g. "Taxi Trondheim AS", as entities. This could be done manually or automatically using a technique called Named-Entity Recognition (NER) [1]. By tagging the content it is easier to extract only the desired text.

5.3 Conclusion

The *translate-first* approach with additional keywords (which characterize the document set or a subset of it) does show some potential. However, this report has focused on the best-case scenario, and the approach might not work as well for all documents.

5.4 Future Work

As seen in section 5.2, there are a number of changes that might improve the results in this project:

- Add more keywords to the search query – This will give more specific results but at the risk that it yields no results.
- Use keywords that are specific to a only subset of the doument set – This will for example add "Eclipse" to the search query for the documents that are located below the Eclipse-category.
- Boost fields in the Lucene index – This will give extra weight to certain words, e.g. the title of the documents.
- Evaluate relevancy of the search results – This will filter irrelevant links before the user even sees them.
- Add Named-Entity Recognition – This will give a higher control of what content that will be added to the Lucene index.

Finally, the implementation could be turned into a working plugin for Confluence. This plugin can either embed the links automatically or suggest links to the author who must approve them.

Bibliography

[1] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *An Introduction to Information Retrieval*. Cambridge University Press, 2009.

[2] Atlassian. Writing confluence plugins. `https://developer.atlassian.com/display/CONFDEV/Writing+Confluence+Plugins`, 2013. [Online; Accessed 2013-12-19].

[3] Rich Hickey. Clojure. `http://clojure.org`, 2012. [Online; Accessed 2013-12-19].

[4] The Apache Software Foundation. Apache tika. `http://tika.apache.org`, 2013. [Online; Accessed 2013-12-19].

[5] Jonathan Hedley. jsoup: Java html parser. `http://jsoup.org`, 2013. [Online; Accessed 2013-12-19].

[6] Google. Google translate api. `https://developers.google.com/translate/`, 2012. [Online; Accessed 2013-12-19].

[7] The Apache Software Foundation. Apache solr. `http://lucene.apache.org/solr/`, 2012. [Online; Accessed 2013-12-19].

[8] Google. Custom search. `https://developers.google.com/custom-search/`, 2013. [Online; Accessed 2013-12-19].

Attachments

1. Confluence Printout – “Tråder med java”