University of Twente

Master Thesis

Ranking Factors for Web Search : Case Study in the Netherlands

Author:
Tesfay Aregay
(s1272616)

Supervisor:
Dr. ir. Djoerd Hiemstra
Dr. ir. Robin Aly
Roy Sterken

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Information and Software Engineering
Department of Computer Science
Faculty of Electrical Engineering, Mathematics and Computer Science

July 2014

Preface

This master thesis defines the research conducted to complete my master study "Computer Science" with a specialization of "Information and Software Engineering" at the University of Twente[1]. It also marks the end of my time as a student, which I enjoyed very much and from which I gained knowledge and experience both in a personal and a professional way.

The research has been established in cooperation with Indenty[2], an online marketing company located in Enschede, The Netherlands. Indenty provides Search Engine Optimization (SEO) and Search Engine Marketing (SEM) tools and services in The Netherlands. At Indenty, I worked closely with the experts of the tooling department, who helped me very much in discussing the interesting issues, as well as in solving the challenges encountered through out the whole period.

This work would not have been possible without the help, guidance and feedback of several individuals. I would like to take the opportunity to express my sincere gratitude towards them. First, I would like to thank my university supervisors Dr. ir. Djoerd Hiemstra and Dr. ir. Robin Aly for their support. They were able to bring me back to the right track when I was lost in a problem and their critical feedback has improved the scientific value of this work.

Furthermore, I would like to thank Indenty for making this project possible, and for giving me a full freedom to make necessary changes on the initial plan whenever needed. I would really like to thank Roy Sterken for his supervision, enthusiasm, and for all the inspiring discussions we had. I would also like to say "thank you" to Piet Schrijver, Daan Nijkamp and Dr. Despina Davoudani who was very involved and assisted me whenever I run into technical problems. In-addition, I would like to thank the colleagues from InnovadisGroep[3] for making me feel welcome, for all the fun we had and for their interesting insights on the project. Finally, I would like to express my gratitude to my family and friends, who were always by my side during the good and difficult times.

It is my sincere wish that you like reading this thesis, and Indenty as well as others finds a way to benefit from the results of this research.

[1] `http://www.utwente.nl/`
[2] `http://www.indenty.nl/`
[3] `http://www.innovadis.com/`

"It is the journey that brings us happiness, not the destination."
Dan Millman

UNIVERSITY OF TWENTE

Abstract

Faculty of Electrical Engineering, Mathematics and Computer Science
Department of Computer Science

Master of Science

Ranking Factors for Web Search : Case Study in the Netherlands

by Tesfay Aregay
(s1272616)

It is essential for search engines to constantly adjust ranking function to satisfy their users, at the same time SEO companies and SEO specialists are observed trying to keep track of the factors prioritized by these ranking functions. In this thesis, the problem of identifying highly influential ranking factors for better ranking on search engines is examined in detail, looking at two different approaches currently in use and their limitations. The first approach is, to calculate correlation coefficient (e.g. Spearman rank) between a factor and the rank of it's corresponding webpages (ranked document in general) on a particular search engine. The second approach is, to train a ranking model using machine learning techniques, on datasets and select the features that contributed most for a better performing ranker.

We present results that show whether or not combining the two approaches of feature selection can lead to a significantly better set of factors that improve the rank of webpages on search engines.

We also provide results that show calculating correlation coefficients between values of ranking factors and a webpage's rank gives stronger result if a dataset that contains a combination of top few and least few ranked pages is used. In addition list of ranking factors that have higher contribution to well-ranking webpages, for the Dutch web dataset(our case study) and LETOR dataset are provided.

Contents

List of Figures

List of Tables

Abbreviations

SEO	**S**earch **E**ngine **O**ptimization
SEA	**S**earch **E**ngine **A**dvertising
SERP	**S**earch **E**ngine **R**esults **P**age
ROI	**R**eturn **O**n **I**nvestment
LETOR	**LE**arning **TO** **R**ank
Ads	**Ad**vertisement
RQ	**R**esearch **Q**uestion
SRQ	**SubR**search **Q**uestion
BL	**B**ack**l**inks
ST	**S**earch **T**erm
FB	**F**ace**b**ook
GP	**G**oogle **P**lus
REF	**Ref**erring
URL	**U**niform **R**esource **L**ocater
API	**A**pplication **P**rogram **I**nterface
NaN	**N**ot **a** **N**umber

Chapter 1

Introduction

In this chapter, an introduction to the research carried out will be presented. Once this is done the reasons that motivated this research and main objective will be outlined. Furthermore, main research questions are formulated and sub questions are defined to aid in answering the main research questions. Finally a short structure of the report is given to guide the reader. Since the detailed explanation for most of the key terms used in this chapter are located in other chapters, a reference to the exact section is given for each of them.

1.1 Background

It is a continuous battle between, on the one end giant search engines (see Section 2.2) like Google continuously updating their ranking algorithms aiming to weed out lower-quality websites from their search result pages (see Section 2.3) to satisfy searchers, on the other end SEO[1] companies and SEO (see Section 2.5) specialists, researchers tirelessly digging to find the secrecy of how exactly these search engines evaluate websites to ultimately determine which site to show for which search term[2](see Section 2.4). This makes it hard task for the later one to keeping track of the algorithms and the ranking factors(see Section 2.6).

Generally there are two approaches to come up with set of ranking factors (also referred as factor or feature) that have higher influence in well-ranking.

The first approach is, calculating correlation coefficient (e.g. Spearman) between a factor and the rank of it's corresponding webpages (ranked document in general) on a

[1] **S**earch **E**ngine **O**ptimization

[2] In this document we use 'search term', 'search query', 'query' and 'keywords' interchangeably

particular search engine. There are several companies, that follow this approach and produce analysis[4] [5] [6] on SEO and SEM[3] to provide advice on which ranking factors should be used, and how it should be implemented. Similarly there are a number of commercial and free SEO tools[4567] that help website owners look into their websites and identify elements of a site that search engines deem as important.

The best example for such tools is the Webmaster Tools, which is a free service offered by Google that helps you monitor and maintain your site's presence in Google Search results. This tool helps to monitor your website traffic, optimize your ranking, and make informed decisions about the appearance of your site's search results[7]. Similarly Indenty, has built a tool called LeadQualifier[8] that perform initial analysis on a website by quickly scanning several online marketing elements. Although they are few in number, the factors checked by the LeadQualifier lie into different categories (technical, content, popularity and social signal) of ranking factors. Some of the checks the tool makes are :

- It checks if a website is accessible to search engines by checking the setting on the Robots.txt file.
- It checks if a website has a sitemap.
- It checks if a website is built with/without frames and flash components.
- It checks if a website has an associated Facebook fan page.
- It also checks the popularity of a website using Google's PageRank[9] and the number of Backlinks[10] it has.

The second approach is, to train a ranking model(also referred as ranker and ranking function) using machine learning techniques, on datasets and select the features that contributed most for a better performing ranker. In the area of machine learning feature selection is the task of selecting a subset of factors to be considered by the learner. This is important since learning with too many features is wasteful and even worse, learning from the wrong features will make the resulting learner less effective[8]. Learning to rank (see Section 2.8) is a relatively new field of study aiming to learn a ranking function from a set of training data with relevance labels [9]. Dang and Croft [8] conducted

[3] **S**earch **E**ngine **M**arketing
[4] `http://moz.com/`
[5] `http://www.screamingfrog.co.uk/seo-spider/`
[6] `https://chrome.google.com/webstore/detail/check-my-links/ojkcdipcgfaekbeaelaapakgnjflfglf?hl=en-GB`
[7] `http://offers.hubspot.com/`
[8] `http://www.leadqualifier.nl/`
[9] `http://en.wikipedia.org/wiki/PageRank`
[10] Currently LeadQualifier gets the Backlinks for a website from other service provider.

an experiment on the LETOR learning to rank dataset with different learning to rank algorithms aiming to select the most important features for document ranking.

The motivation for this research comes from the problems and drawbacks we observed in both of these two approaches. We observe some common limitations with the LeadQualifier in particular and most of the other SEO tools we came across in general. Likewise we have identified a number of limitations regarding the SEO analysis which are published by SEO companies and the dataset used to generate their reports. In-addition, we have noted some drawbacks of the datasets used in learning to rank to train ranking systems . The limitations are discussed below, categorized in to three topics.

1. **Limitations of SEO Tools :**

   (a) The LeadQualifier needs to implement a check for more factors to give a better advice on how to improve a website's search engines visibility, currently it has implemented less than 20 checks. There are over 200 different factors (or signals) used by Google[10] to rank webpages, although it is not known what these factors are.

   (b) The most important factors should be given a priority when performing the checks, therefore knowing which factors are more important is necessary.

   (c) The LeadQualifier should be less dependent on external tools such as the PageRank. Google used to have a publicly available SOAP API to retrieve the PageRank of URL but not any more. As a result there is a growing concern that the PageRank may cease to exist eventually, leaving the LeadQualifier and other similar SEO tools at risk.

2. **Limitations of SEO Companies' Analysis :**

   (a) There is huge difference among the claims being made by different parties concerning which factors are the most influential ones for ranking better on search engines.

   (b) There is no guarantee that the ranking factors suggested by different SEO companies (e.g. SearchMetrics[11], Moz[12]) and experts are valid since most of them are not scientifically supported rather are based on survey, on a non-representative sample dataset analysis and experience.

   (c) Moreover, there is no enough research carried out to approve or disapprove that the generic ranking factors suggested by experts and SEO companies are applicable to searches originating from specific region. For instance we

[11] `http://www.searchmetrics.com/en/`
[12] `http://moz.com/`

are not sure if the ranking factors suggested by NetMark[5] are applicable for search quires submitted on The Netherlands version of Google(i.e. Google.nl). Sometimes search results of same search query on Google.nl and Google.com is different. We found it very interesting, to see the different "Related Searches" suggestion Google provided for exactly same query (i.e. "Jaguar"[13] submitted to Google.nl and Google.com at the same time. Table 1.1 shows, out of the 8 suggestion only one (i.e "jaguar f type") was suggested by both Google.nl and Google.com as a "Related Searches" for the query "Jaguar". This implicates that the ranking algorithm used in one data center is subtly different from the ranking algorithm used in another, thus the factors used might also be different.

(d) Some previous studies on Google's ranking algorithm have not concluded whether or not correlation is causal. For instance SearchMetrics have clearly pointed out that : $correlation \neq causation$. Which means higher correlation does not necessary show that, having that particular factor will bring a lead on search results. Instead a correlation should be interpreted as a characteristics of well ranked pages.

(e) SEO Companies are too reluctant to clearly define the methodology they follow while producing their correlation studies, and only few of them have provided the full dataset (query, url, feature) openly for the public.

3. **Limitations of the Learning To Rank Datasets:**

(a) Most of the common learning to rank benchmark datasets do not disclose the set of queries, documents, factors they used (e.g. Microsoft and Yahoo!).

[13]Jaguar : Jaguar Cars is a brand of Jaguar Land Rover, a British multinational car manufacturer (`http://en.wikipedia.org/wiki/Jaguar_Cars`, July 04, 2014). At the same time Jaguar is a big cat, a feline in the Panthera genus, and is the only Panthera species found in the Americas (`http://en.wikipedia.org/wiki/Jaguar`, July 04, 2014).

TABLE 1.1: List of "Related Searches" suggestion given for a search term "Jaguar" on Google.nl and Google.com

Google.nl	Google.com
jaguar animal	jaguar parts
jaguar price	atari jaguar
jaguar mining	jaguar forum
jaguar fittings	used jaguar
jaguar f type	jaguar xf
jaguar bathroom fittings	jaguar f type
jaguar land rover	jaguar e type
jaguar xk	jaguar xke

It is important for the reader to understand that we do not intend to solve all the limitations listed above in this research. To begin with, as a partial solution for limitation 2(a), we have identified 70 factors and decide to include them in the research. Also we aim that, by calculating correlation values for each factors, we will know which factors are more important than others, which gives an answer to limitation 1(b). Although it is not part of this research, we believe it is possible to calculate PageRank for webpages/websites by analyzing large corpus of webpages like the CommonCrawl[14] data, which could partially solve the limitation mentioned on 1(c).

Limitations 2(a), 2(b) and 2(c) can be regarded as the core problems that initiated this research. As mentioned in 1(a) the different (sometimes colliding) claims released by SEO companies, which are broadly discussed in Chapter 3 Section 3.4, were quite alarming to conduct our own investigation on the issue. While performing the intensive background research, we observe that, there are ranking factor analysis white-paper publications based on datasets optimized to the USA, Germany, France, Italy and UK, on search engines such Bing, and Google. To our knowledge there is none such study conducted mainly for The Netherlands, so we figured to make the case study of this research to The Netherlands, which will help us answer the problem mentioned on 2(c).

When it comes to choosing a dataset suitable to the goals of our research, we had two possible options. The first one was to search for a publicly available dataset that is constructed for similar research. So, we began by looking into the raw data released by Moz which was used in their analysis on the U.S. search results from Google search engine(can be retrieved from this link `http://d2eeipcrcdle6.cloudfront.net/search-ranking-factors/2013/ranking_factors_2013_all_data.tsv.zip`). However, this conflicted with our wish to do analysis on the Dutch web. Then we discovered the learning to rank benchmark datasets such as LETOR4.0 (see Section 3.5). These

[14] http://commoncrawl.org/

benchmark datasets are very well constructed for multiple purposes, but as mentioned on 3(a) they do not suit to be used as dataset on our research, because most of them do not disclose the set of queries, documents (webpages in our case) and factors used. All this lead us to the second option, which was to collect/prepare our own dataset which contains query, url and factors (this set is also referred as "DUTCH WEB" in this document) and perform experimental analysis, at the same time give an answers to the limitation mentioned on 2(c) and 3(a).

The other thing is, as pointed out in 2(d) there is no clear knowledge on how to interpret the previously published (mainly white-papers) correlation result on similar researches. It is therefore our wish to answer this limitation by training a ranking model based on the collected dataset, extract the weight of the factors used in the model and compare it with their relative correlation result. Another possible approach would be to re-rank a test dataset using the trained ranking model and compare the rank with Google's position/rank of each webpages, this way the correlation results could be evaluated.

Finally, the researcher wish to publicly release a clear definition of the methodology followed while producing the results of this thesis and the datasets collected and constructed for this purpose. As mentioned in 2(e) such resources can encourage fellow researchers to perform more analysis on the web particularly the Dutch web and play their role in solving problems similar to the ones discussed here.

Concluding, the challenge of this research is confined to, finding highly influential factors to well-ranking webpages by integrating the two approaches introduced above, using DUTCH WEB and LETOR datasets. At same time compare, complement, evaluate results of the first approach by the second approach, in order to achieve scientifically supported and reliable result.

1.2 Objective

Formally the main objective of this research can be defined as follows:
Design and conduct an empirical research, to come up with scientifically evaluated list of ranking factors that are highly influential (factors that have higher contribution to well ranked webpages), using the DUTCH WEB and LETOR4.0 datasets.

1.3 Research Questions

In this section the general problem discussed above is refined into a clearly formulated research questions and some sub-questions.

The fundamentals of this research are based on the following three main research questions(RQ). To be able answer the main research questions, we break them down into sub research questions (SRQ).

- **RQ-1:** *Which ranking factors influence organic search results(see Section 2.3)?*
  - **SRQ-1.1:** *Which techniques exist to identify the most important factors for ranking well on search engines?*
  - **SRQ-1.2:** *Which ranking factors have been studied in previous researches?*
- **RQ-2:** *Is it better to use only the top well ranked pages (e.g top 40) while computing correlation coefficients instead of using all ranked pages per search term?*
- **RQ-3:** *How can we evaluate the importance of ranking factors?*
  - **SRQ-3.1:** *Is there any sensible relationship between the calculated correlation coefficient of a ranking factor (first approach) and it's corresponding weight assigned by a ranker(second approach)?*
  - **SRQ-3.2:** *Does considering highly correlated ranking factors give a better performing ranker, compared to using the whole set of ranking factors?*

1.4 Approach

In this section a high level view to the approach followed to conduct this research is presented. We reviewed similar works from the academia and the SEO industry, to learn the techniques used, the features analyzed and the results obtained.

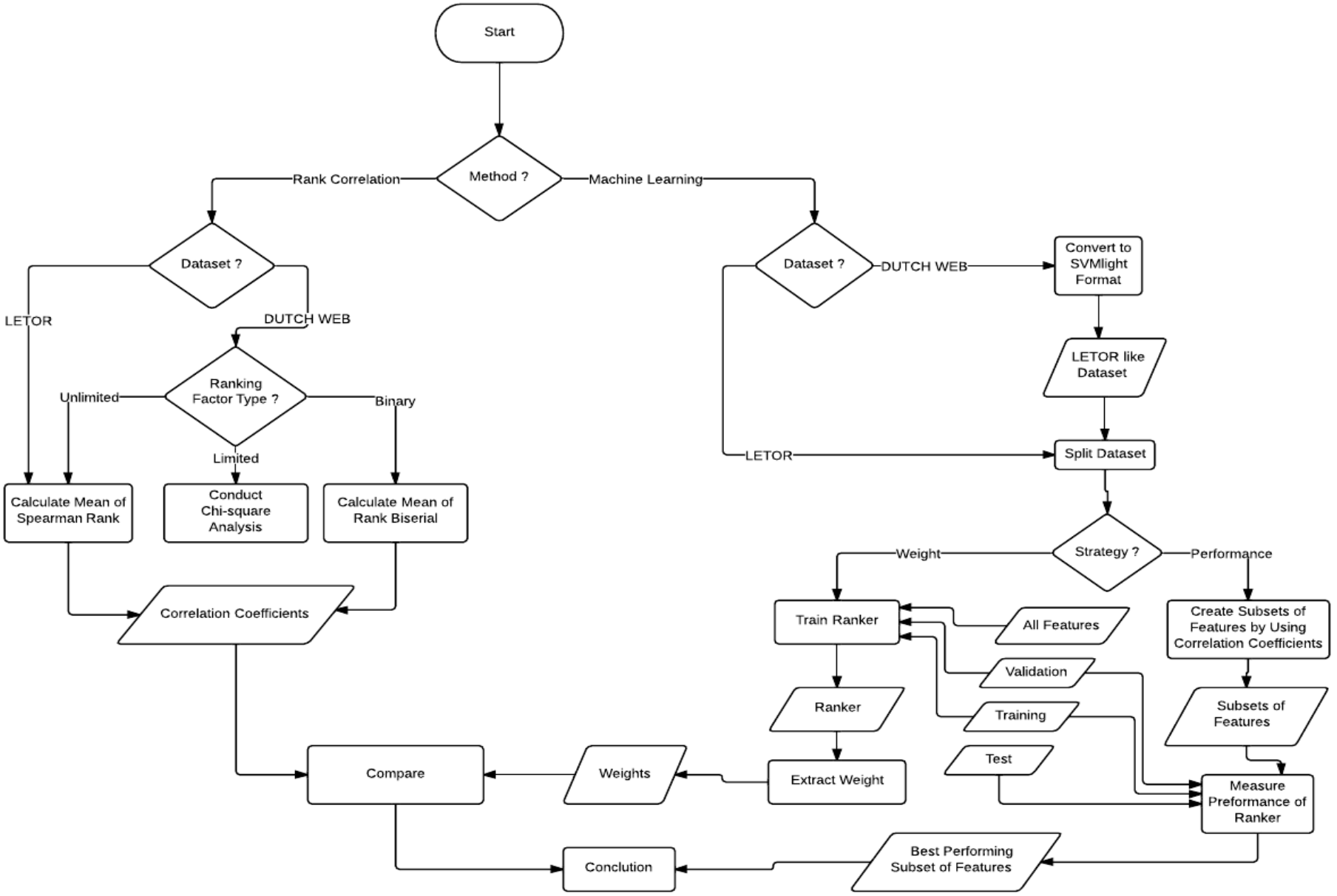
We conduct our research on two different datasets, the LETOR and DUTCH WEB datasets. To construct the DUTCH WEB dataset, we pull a total of 10,000 Dutch search terms from local database of Indenty. As it will be explained in Chapter 3, it is a common practice to use search terms of approximately 10,000 for correlation analysis, hence we believe that using a sample of this size can represent all search terms. Then, we fetched a maximum of top 40 webpages for each search term from the search result page of Google.nl. From each webpage 52 factors are extracted to build the full DUTCH WEB dataset which contains (query, factors and document). More about the datasets, and how they were collected, cleaned etc is discussed in Chapter 4.

For the DUTCH WEB dataset we calculate Rank-Biserial and both Spearman Rank and Kendall Tau correlation coefficients for dichotomous and continuous ranking factors respectively. The reasons for choosing these correlation coefficients will become more

clear in Chapter 3 and Chapter 5. Similarly we computed Spearman Rank correlation between the features of the LETOR dataset and the ground-truth of each ranked webpage. For the DUTCH WEB dataset we also performed additional statistical analysis, and presented the results in percentage.

Later the DUTCH WEB dataset was used to construct another LETOR like dataset in the SVMLight format for the purpose of training a ranking model. This new LETOR like dataset was further divided into three subsets "TRAIN SUBSET" 60%, "VALIDATION SUBSET" 20% and "TEST SUBSET" 20%. Two listwise approaches of learning to rank algorithms namely Coordinate Ascent, and LambdaMART are used to train ranking models using the "TRAIN SUBSET" and "VALIDATION SUBSET". Later, to determine how well the trained models perform we conducted an evaluation using the unseen and untouched "TEST SUBSET". We used ERR@10 and NDCG@10 to measure the performance of the models. A similar model is again trained using the LETOR dataset. We unpack the trained models, and look into the weight of the factors assigned by the trained models. Comparing the weight of these factors with their previously calculated correlation coefficient enabled us to answer the core question of this research. The evaluation process and techniques utilized are broadly discussed in Chapter 7.

The flow diagram below depict the whole process flow of tasks involved in this research. Some of the task names used in the diagram will be more clear in the coming chapters.


Start
Method ?
Rank Correlation
Machine Learning
Dataset ?
LETOR
DUTCH WEB
Ranking Factor Type ?
Unlimited
Limited
Binary
Calculate Mean of Spearman Rank
Conduct Chi-square Analysis
Calculate Mean of Rank Biserial
Correlation Coefficients
Compare
Weights
Extract Weight
Ranker
Train Ranker
All Features
Validation
Training
Test
Dataset ?
DUTCH WEB
Convert to SVMlight Format
LETOR like Dataset
LETOR
Split Dataset
Strategy ?
Weight
Performance
Create Subsets of Features by Using Correlation Coefficients
Subsets of Features
Measure Preformance of Ranker
Best Performing Subset of Features
Conclution

1.5 Contributions

1.5.1 Theoretical

This thesis provides results to show that, considering few from the bottom and few from the top (40 for each in our case) of the ranked webpages gives stronger correlation coefficient. Additionally, we deduced indirectly that strong positive (if not just positive) correlation is not always a cause for well ranking.

1.5.2 Algorithmic

To conduct the research, it was necessary to write our own algorithms for some of the correlations, particularly Rank-Biserial Correlation and Spearman Rank Correlation. In-addition, an algorithm that produce score (relevance judgment) out of Google's position, and an algorithm for converting position into natural ranking were written. Details about algorithms and the over all methodology is broadly discussed in Chapter 5.

1.5.3 Percentage and Correlation Results

As mentioned earlier the main goal of the research is finding a list of ranking factors that have a higher influence on well ranking, given that it also provides some statistical (percentage) results that show insights to the DUTCH WEB.

1.5.4 Prototype

The system developed to gather the dataset, extract the factors and the algorithms implemented to calculate correlations, and to construct dataset for learning to rank algorithms can be taken as prototype, to build commercial tool.

1.5.5 Dataset

Unlike previous studies done with similar goal as this research, the content of the datasets analyzed in this research is mainly composed in Dutch language. Therefore it is our belief that this datasets are unique, useful assets and we consider them as part of the contribution of this research. This datasets (listed below) are gathered, cleaned, filtered and constructed in certain format to fit the goal of the research. Yet, they can be re-used to conduct similar researches. Details of the datasets is provided in Chapter 4

1. Dataset of nearly 7568 Dutch search terms, which passed necessary cleaning and filters.

2. Raw data (i.e. downloaded webpages), approximately 21.6 GB size, which could be re-used to conduct similar researches.

3. Dataset that contains 52 factors along with their values for around 300639 (=7568x40) webpages.

4. Dataset constructed in SVMLight format for the purpose of training a ranking model using learning to rank techniques.

1.6 Document Structure

This document is structured as follow: Chapter 1 (this chapter) provides an introduction to the thesis. Chapter 2 provides preliminary knowledge about the key terms and concepts which are discussed through out the rest of this document. Chapter 3 discusses previous researches conducted related to the topic, both from academia and from the SEO industry including companies and experts. Chapter 4 discusses the process of the data gathering, factor extraction other basic information about the dataset which was used to conduct this research. Chapter 5 presents the methodology used to conduct this research, algorithms developed, and high level design of the prototype system developed. Chapter 6 presents correlation graphs for the DUTCH WEB, and LETOR datasets and discusses the results. In Chapter 7 we talk about our evaluation measures, evaluation strategies and discusses the evaluation results using learning to rank algorithms. Chapter 8 gives the conclusion, recommendation and future studies. Basic information on the LETOR dataset can be found in Appendix A. Appendix B contains, the raw data used to make the graphs in Chapter 7 plus terminal commands and parameter settings used to train models are provided. Further analysis results about the DUTCH WEB are provided in Appendix C.

Chapter 2

Background : Web Search and Ranking

The main goal of this chapter is to provide a background knowledge related to web search and ranking. It includes the definition and explanation of the key aspects and concepts that are discussed through out the rest of the document. This chapter is supposed to help reader define the key terms, so that he/she can have a clear picture of the intention of the research.

2.1 Web Search

Web search is the act of looking for webpages on search engines such as Google or Bing. Webpages are web documents which can be located by an identifier called a uniform resource locator (URL) for example: `http://www.utwente.nl/onderzoek/` (see Section 4.2.1). Webpages are usually grouped into websites, sets of pages published together for example: `http://www.utwente.nl`[11]. The entire collection of all interlinked webpages located around the planet is called the Web, also known as the World Wide Web (WWW)[1]. In 2014, Google announced[2] the web is made up of 60 trillion (60,000,000,000,000) individual pages with makes an index of over 100 million gigabytes, and it is constantly growing. According to WorldWideWebSize.Com[3] the Dutch indexed web alone is estimated to be at least 204.36 million pages until 05 June, 2014.

When someone perform a web search on search engines he will get back a list of hyperlinks to prospective webpages. This list may have a hundred or more links. They are

[1] `http://en.wikipedia.org/wiki/World_Wide_Web(01,May,2014)`
[2] `http://www.google.com/insidesearch/howsearchworks/thestory/`(05,June,2014)
[3] `http://worldwidewebsize.com/index.php?lang=NL`(05,June,2014)

often divided up into a number of SERPs(see Section 2.3). From a SERP, he can decide which link he should try and see if it contains what he is looking for.

2.2 Web Search Engine

Web search engines are very important tools to discover any information in World Wide Web[12]. When Internet users want to work on something they usually start with search engines 88% of the time[4].

To explain what a search engine is we like to use a real world analogy. Search engines such as Google and Bing are like a librarian, not a normal one but a librarian for every book in the world. People depend on the librarian every day to find the exact book they need. To do this efficiently the librarian needs a system, and he needs to know what is inside every book and how books relate to each other. He could gather information about the books by reading the books' titles, categories, abstracts etc. His system needs to take in the gathered information, process it and spit out the best answer for a reader's question. Similarly search engines are librarians of the Internet, their system collect information about every page on the web so that they can help people find exactly what they are looking for. And every search engine has a secret algorithm which is like a recipe for turning all that information in to useful organic or paid search[5].

Search engines such as Google and Bing provide a service for searching billions of indexed webpages for free. The result search engines display for every search query submitted is composed of free (none ads[6]) and paid (ads) webpages. The naturally ranked webpages also known as organic search are webpages determined by search engine algorithms for free, and can be optimized with various SEO practices. In contrast, paid search allows website owners to pay to have their website displayed on the search engine results page when search engine users type in specific keywords or phrases[7]. The figure below [Figure 2.1] depicts the elements inside search engines and flow of the process.

[4] `http://www.nngroup.com/articles/search-engines-become-answer-engines/`(05,June,2014)
[5] `http://www.goldcoast.qld.gov.au/library/documents/search_engine_optimisation.pdf`
[6] Advertisement
[7] http://offers.hubspot.com/organic-vs-paid-search

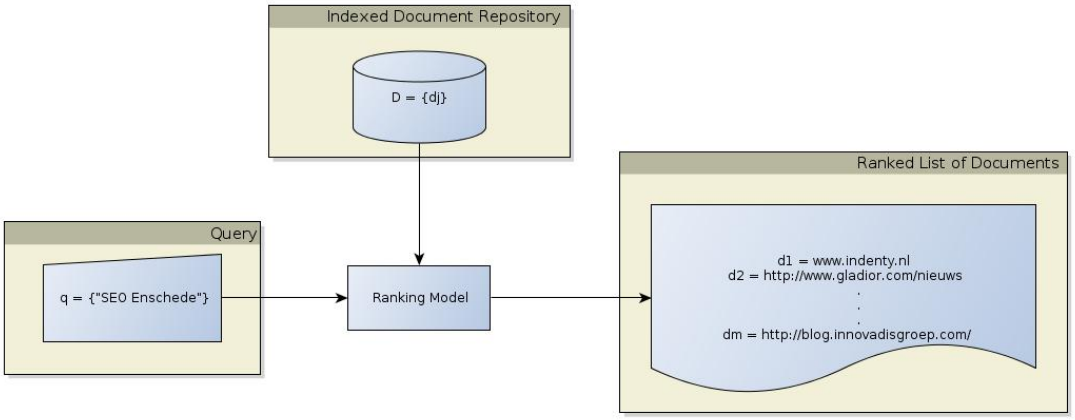


FIGURE 2.1: Ranking inside search engine

2.3 Search Engine Results Page (SERP)

A search engine results page is the listing of results returned by a search engine in response to a keyword query[8]. The results normally include a list of items with titles, a reference to the full version, and a short description showing where the keywords have matched content within the page. If we see into Google's SERP, the elements / listings included in a SERP are growing in number and in type. Some of the elements of a SERP are :

- **Organic Results :** Organic SERP listing are natural results generated by search engines after measuring many factors, and calculating their relevance in relational to the triggering search term. In Google's term, organic search results are web-pages from a website that are showing in Google's free organic search listings[9]. As mentioned above only organic search results are affected by search engine optimization, not paid or "sponsored" results such as Google AdWords[10].

- **Paid Results :** Paid also know as "Sponsored" search results, are listing on the SERP that are displayed by search engines for paying customers (website owners) which are set to be triggered by particular search term (e.g. Google Adwords)[10].

- **Knowledge Graph :** The Knowledge Graph is a relatively newer SERP element observed on search engines particularly Google used to display a block of information about a subject[11]. This listing also shows an answer for fact questions such as "King Willem Alexander Birthday" or "Martin Luther King Jr Assassination".

[8] http://en.wikipedia.org/wiki/Search_engine_results_page
[9] https://support.google.com/adwords/answer/3097241?hl=en(June 11, 2014)
[10] http://serpbox.org/blog/what-does-serp-mean/
[11] http://moz.com/blog/mega-serp-a-visual-guide-to-google

- **Related Searches :** This part of the SERP is where search engines provide suggestion on related search terms to the one submitted.

2.4 Search Term

Billions of people all around the world conduct search each day by submitting search terms on popular search engines and social networking websites. A search term also know as keyword is the textual query submitted to search engines by users.
Note : In this document search term, keyword, and query will be used interchangeably, therefore the reader should regard them as synonymous.

2.5 Search Engine Optimization (SEO)

For companies, or individuals who own a website search results matter, when their page have higher ranking it helps people find them. E-commerce companies are very interested and curious on how the ranking is done. This is due to the fact that being found on the Internet for a given search term is continuously becoming major factor to maximize ROI[12].

The key to higher ranking is making sure the website has the ingredients also known as "raking factors" search engines need for their algorithm that we refer as recipe on the previous sub section, and this process is called Search Engine Optimization (SEO). In other words Search Engine Optimization is often about making small modifications on your website such as the content and code. When viewed individually, these changes might seem like incremental improvements but they could have a noticeable impact on your site's user experience and performance in organic search results[10].

2.6 Ranking Factors

Ranking factors also known as ranking criteria are the factors used by search engines in evaluating the order of relevance of a webpage when someone searches for a particular word or phrase[13]. It is almost obvious that the ranking factors have different weight assigned to them. For instance according to SearchMetrics white paper SEO guideline made for Bing USA 2013[4], "the existence of keyword on domain" is still one of the major ranking factor probably with the highest weight.

[12] Return on Investment
[13] `http://marketinglion.co.uk/learning-lab/search-marketing-dictionary`

Although different entities(companies, individuals) independently suggest various factors for ranking well on search results, there are some basic SEO practices. To give a sense of what these practices are, we will discuss some of them here. First, words used in the content of a webpage matter, search engine account for every word on the web, this way when someone search for "shoe repair" the search engine can narrow results to only the pages that are about those words. Second, titles matter, each page on the web has an official title, users may not see it because it is in the code. Search engine pay a lot of attention to titles because they often summarize the page like a book's title. Third, links between websites matter, when one webpage links to another it is usually a recommendation telling readers this site has good information. A webpage with a lot of links coming to it can look good to search engines but some people try to fool the search engine by creating or buying bogus links all over the web that points to their own website. This phenomenon is called Search Engine Persuasion (SEP) or Web Spamming [13]. Usually search engines can detect when a site has a lot of them, and they account for it by giving links from trustworthy site more weight in their ranking algorithm[14]. Fourth, the words that are used in links also know as anchor text matter too, if your webpage says "Amazon has lots of books" and the word "books" is linked, search engine can establish that amazon.com is related to the word "books", this way when some one search "books" that site will rank well. Lastly, search engines care about reputation, sites with consistent record of fresh, engaging content and growing number of quality links may be considered rising stars and do well in search rankings. These are just the basics and search engine algorithms are fined and changed all the time which makes chasing the algorithms of giant search engines such as Google always difficult. Apparently, good SEO is not just about chasing the algorithm but making sure that a website is built with all the factors search engines need for their algorithms[15].
Note : In this document ranking factor, ranking criteria, and feature will be used interchangeably, therefore the reader should regard them as synonymous.

2.7 Webpage Ranking

Ranking is sorting objects based on certain factors[14]: given a query, candidates documents have to be ranked according to their relevance to the query[15]. Traditionally, webpage ranking on search engines was done using a manually designed ranking function such as BM25, which is based on the probabilistic retrieval framework. Where as now, as it will be discussed in Section 2.8, webpage ranking is consider as a problem of Learning to rank.

[14] `http://searchengineland.com/guide/what-is-seo`
[15] `http://sbrc.centurylink.com/videos/marketing/digital-marketing/search-engine-optimization-seo/`

2.8 Learning to Rank (LETOR)

The task of "learning to rank" abbreviated as LETOR has emerged as an active and growing area of research both in information retrieval and machine learning. The goal is to design and apply methods to automatically learn a function from training data, such that the function can sort objects (e.g., documents) according to their degrees of relevance, preference, or importance as defined in a specific application[16]. The steps followed when learning to rank if it is applied to a collection of documents (i.e. webpages in our case) are :

1. A number of queries or search terms are accumulated to make a training model; each search terms are linked to set of documents(webpages).
2. Certain factors are extracted for each query-document pair, to make a feature vector(i.e. list of factor id and with their relative value).
3. A relevance judgments (e.g. perfect, excellent,good, fair or bad), which indicates the degree of relevance of each document to its corresponding query, are included in the data.
4. Ranking function also know as ranking model is created by providing the training data to a machine learning algorithms, so that it can accurately predict the rank of the documents.
5. In testing, the ranking function will be used to re-rank the list of documents when a new search term is submitted[16].
6. To measure how well the ranking function did the prediction, evaluation metrics like Discounted Cumulative Gain(DCG)[17] or Normalized Discounted Cumulative Gain(NDCG)[18] are required.

Figure 2.2 precisely shows the process flow of learning to rank and the components involved[19].

Generally there are three types of learning to rank approaches, these are :

- **Pointwise Approach :** The pointwise approach regards a single document as its input in learning and defines its loss function based on individual documents[20].
- **Pairwise Approach :** The pairwise approach takes document pairs as instances in learning, formalizes as document A is more relevant than document B with respect to query q.

[16] `http://research.microsoft.com/en-us/um/beijing/events/lr4ir-2008/`

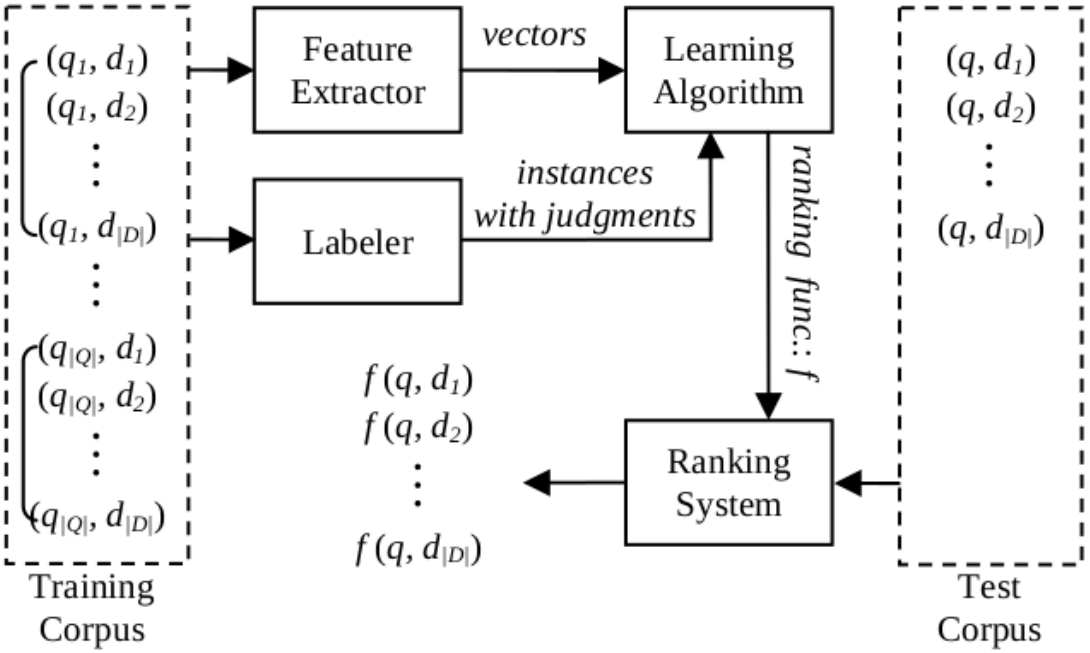


FIGURE 2.2: A general paradigm of learning to rank for IR[19].

- **Listwise Approach :** Listwise learning to rank operates on complete result rankings. These approaches take as input the n-dimensional feature vectors of all m candidate documents for a given query and learn to predict either the scores for all candidate documents, or complete permutations of documents[20]. Some of listwise models are : AdaRank, ListNet,LambdaMART, Coordinate Ascent.

Note : In this document ranking model, ranking function, and ranker will be used interchangeably, therefore the reader should regard them as synonymous.

2.9 Summary

The naturally ranked webpages also known as organic search are webpages determined by search engine algorithms for free, and can be optimized with various SEO practices. Search Engine Optimization (SEO) is often about making small modifications on your website such as the content and code to get higher ranking on search engines.

Chapter 3

Related Work

This chapter will present a review of previous and continuing researches that are related to the topic of this thesis. The review was conducted with an intent to answer the question SRQ-1.2 :*"Which techniques exist to identify the most important factors for ranking well on search engines?"*. In chapter 1 we mentioned that there are two approaches that are currently followed to identify important ranking factor, and here we review previous works for both approaches. Along side, we assess the ranking factors analyzed in these researches, and present a summarized review to answer SRQ-1.2 :*"Which ranking factors have been studied in previous researches ?"*.

A concise result on the search carried out to discover what benchmark datasets exist, how are they constructed/prepared to conduct similar researches is also included in this chapter. At last it gives a comparison tables on the ranking factors analyzed by different SEO companies, as well as the academia, and sum up with a summary.

3.1 Machine Learning Based Studies

The works reviewed here utilized different machine learning techniques to conduct their researches. As introduced in previous chapters, one way of coming up with set of important ranking factors for well ranking is : to train a ranking model using machine learning techniques (ranking algorithms), on datasets and select the factors that contributed most for a better performing ranker. Here, we include two previous works conducted with the same basic approach but different end goal (e.g. reverse engineer Google's ranking algorithm).

The first work, is a research by Su et al. [2], they tried to predict the search results of Google. First they identified 17 ranking factors (see Table 3.4), then prepared a

dataset of 60 search terms, scanned top 100 ranked webpages from Google.com, download webpages from the original website and extract the ranking factors from the pages. Then they train different models on a training subset (15 search terms) and later predict ranks of webpages on Google for a test subset (45 search terms). They experimented on linear programming algorithm which makes a *pairwise* comparison between two documents in a given dataset. Given a set of documents, pre-defined Google's ranking, and a ranking algorithm A, their goal was to find a set of weights that makes the ranking algorithm reproduce Google ranking with minimum errors. Inaddition, they experimented on linear and polynomial implementations of SVM-rank, which also makes a *pairwise* comparison between a pair of documents. They showed results that indicate linear learning models, coupled with a recursive partitioning ranking scheme, are capable of reverse engineering Google's ranking algorithm with high accuracy. More interestingly, *they analyzed the relative importance of the ranking factors towards contributing to the overall ranking of a page by looking into the weights of the ranking factors assigned by trained ranking models.* Based on their experiments, they consistently identified PageRank as the most dominate factor. Keyword in hostname, and keyword in title tag, keyword in meta description tag and keyword in URL path are also among their leading factors. Unfortunately, the general validity of this paper's result made us a bit skeptical due to the limited dataset that was used in the experiments. On top of that, this paper experimented on webpages that are composed in English. However, despite this disclaimer, we used the methodologies of this paper as a foundation to formulate our approach.

A similar research by Bifet et al. [1] tried to approximate the underlying ranking functions of Google by analyzing query results. First they gathered numeric values of observed features from every query result, thus converting webpages in to vectors. Then, they trained their models on the difference vectors between documents at different ranks. They used three machine learning techniques (binary classification, logistic regression and support vector machines) along with the features to build their models. With the binary classification model, they formulate their problem as *pairwise* comparison : given a pair of webpages they try to predict which one is ranked above the other, hence the model do not give a full ranking. With the models from logistic regression and support vector machines, they were able to get full ranking of the webpages. Their main goal was to obtain an estimation function f for the scoring function of a search engine, and then to compare their predicted rankings with the actual rankings of Google. *To analyze the importance of the features they computed precision values obtained using only individual features to predict the ranking.* The authors used a dataset containing keywords from 4 different categories (Arts, States, Spam, Multiple) each holding 12 keywords. These 12 search terms are further divided into three disjoint sets (7 training terms, 2 validation terms and 3 test terms). *However, the search terms they selected sounds arbitrary, and*

fail to represent the typical user query both qualitatively and quantitatively. For each query the top 100 result webpages are downloaded. Using the Google API 5 inlinks for each URLs of each result webpages are retrieved and they considered only HTML pages on their experiment. When we see to their outcome, the models only marginally outperformed the strongest individual feature (i.e., the feature with the most predictive power) for a given keyword category. Based on this result, the authors concluded that Google uses numerous ranking factors that are "hidden" (i.e., not directly observable outside of Google).

Bifet et al. [1] indicated few interesting points as reasons for not performing well. Some of them are, in certain countries search engines voluntarily cooperate with the authorities to exclude certain webpages for legal reasons from the results. It appears that certain webpages are pushed up or down on queries for reasons related to advertisement or other agreements. Another interesting idea pointed out on this paper is that it is possible that search engines take *user profile and geographic location* of query initiators into account. For example someone in a third world country with very slow Internet connection might be interested in result pages totally different than someone in first world country with better connection speed. Their paper also mentioned some room for improvements, the best precision achieved was only 65% for all the features, datasets and methods considered. Better precision can be obtained on the prediction by making substantial change on the features and dataset used.

To summarize, from these works we learn how machine learning techniques could be used to discover the influence of ranking factors in search engines. One of the common shortcomings we observed from these works is : the fact that their results are based on a small and non-representative dataset analysis.

3.2 Rank Correlations Based Studies

An other approach to identify the influence of factors on ranking is to calculate rank correlation coefficients between feature values and rank of webpages on certain search engine. There are many companies which follow this approach, however the review here elaborates works from three of the leading SEO companies currently on the business namely SearchMetrics[1], SEOMoz[2], and NetMark[3]. In this section, brief discussion about the methodology they use and findings of these companies will be presented. The figure below [Figure3.1] is used to elaborate how the correlation coefficients are calculated in the next sections.

[1] `http://www.searchmetrics.com/en/white-paper/ranking-factors-bing/`
[2] `http://moz.com/blog/ranking-factors-2013/`
[3] `http://www.netmark.com/google-ranking-factors`

			Ranking Factors			
	Google's Result		Unlimited		Dichotomous	Limited
Search Terms	Position	URL	Backlinks	Facebook Share	Search Term = Domain	TLD Extention
"restaurants in enschede"	1	http://www.restaurantsinenschede.nl/hor	2000	230	1	.nl
	2	http://www.restaurant1.com	300	123	0	.com
	3	http://restaurants-in-enschede.net/index.html	1000	12141	1	.net

FIGURE 3.1: Fictitious data to help explain the concepts and equations in this chapter which are referring to this table

3.2.1 Spearman Rank Correlation

In statistics, Spearman's rank correlation coefficient is a nonparametric measure of statistical dependence between two variables[4]. A high positive correlation coefficient occurs for a factor if higher ranking pages have that feature / or more of that feature, while lower ranking pages do not / or have less of that feature. SearchMetrics produces a number of white papers and guidelines focusing on the definition and evaluation of most important factors that have high rank correlation with top organic search results of several search engines. Recently they have released evaluation white paper for Bing.com in the USA for the year 2013 [4], similarly they have published white papers optimized for Google.co.uk, Google.fr, Google.it etc.. They use Spearman correlation to assesses how well the relationship between rank of a webpage and a particular ranking factor is. According to their study technical site structure and good content are basic requirements for ranking well. Also social signals have a clear positive correlation to higher ranking, with Google+ leading the rest of the social medias.

SearchMetrics analyses are based on search results for a very large keyword set of 10,000 search terms from Bing USA. The first three pages of organic search results(SERPs) (i.e. maximum of 30 webpages) were always used as a data pool for each search term, which sums up to a maximum of 30*10,000 = 30,0000 webpages in total.

Even though, SearchMetrics's reports are the most recent and detailed analysis on SEO ranking factors (to our knowledge), some SEO experts[5] criticizes SearchMetrics for releasing confusing reports such as saying "keywords in title have 0 correlation coefficient". Another limitation of SearchMetric's reports is the fact that they have not conducted an analysis optimized for Google Netherlands yet.

Similarly Moz [6] runs a ranking factors study to determine which attributes of pages and sites have the strongest association with ranking highly in Google. Their study consists of two parts: a survey of professional SEOs and a large Spearman correlation based analysis. On their most recent study Moz surveyed over 120 leading search marketers who provided expert opinions on over 80 ranking factors. For their correlation study,

[4] `http://en.wikipedia.org/wiki/Spearman's_rank_correlation_coefficient`
[5] `http://www.clicksandclients.com/2013-rank-correlation-report/`

since they had a wide variety of factors and factor distributions (many of which are not Gaussian), they preferred Spearman correlation than the more familiar Pearson correlation (as Pearson correlation assumes the variables are Gaussian)[6]. The dataset they used contains a list of 14,641 queries, and collected the top 50 search results for each of the queries on the query list from Google's U.S. search engine.

Moz's key findings include: Page Authority[7] correlates higher than any other metric they have measured. Social signals, especially Google +1s and Facebook shares are highly correlated to Google's ranking. Despite the updates (Google Panda[8] and Google Penguin[9]), anchor text correlations remain as strong as ever. On its report Moz made it clear that the factors evaluated are not evidence of what search engines use to rank websites, but simply show the characteristics of webpages that tend to rank higher.

With slightly different approach, Netmark [5] calculated mean Spearman rank correlation by first calculating correlation coefficient for each keyword and then averaged the results together. Their main reason for choosing mean Spearman rank correlation coefficient is to keep the queries independent from one another. Below is the formula for Spearman rank correlation coefficient when no duplicates(ties) are expected[21].

$$\rho = 1 - \frac{6\Sigma d_i^2}{n(n^2-1)} \tag{3.1}$$

ρ = rho (the correlation coefficient)
d_i = the differences between the ranks ($d_i = x_i - y_i$)
n = the total number of observations

To explain how this formula (3.1) is used to calculate the mean Spearman correlation coefficient an example is provided below :

Let's say we want to find out how well Facebook shares of a particular website's/webpage's fan page (x) are correlated to Google's ranking (y) for given search term (see 'Position' and 'Facebook Share' columns from Figure 3.1). The first step is to sort the Google results (i.e. the ranked pages) by their Facebook shares in descending order. Next, we take the difference between the rank of Facebook share of a page and the rank of page's position on Google which gives us a the variable $d_i = x_i - y_i$. Now all the variables we need for the above formula (3.1) are provided. To keep the search terms independent

[6] `http://moz.com/search-ranking-factors/methodology#survey`

[7] Page Authority is Moz's calculated metric for how well a given webpage is likely to rank in Google.com's search results. It is based off data from the Mozscape web index and includes link counts, MozRank, MozTrust, and dozens of other factors.(`http://moz.com/learn/seo/page-authority`, July 10, 2014)

[8] `http://en.wikipedia.org/wiki/Google_Panda`

[9] `http://en.wikipedia.org/wiki/Google_Penguin`

from each other, Spearchman rank correlation coefficient is calculated for each search term, and then averaged across all the search terms for the final result (mean Spearman rank correlation coefficient).

3.2.2 Kendall Rank Correlation

The Kendall (1955) rank correlation coefficient evaluates the degree of similarity between two sets of ranks given to a same set of objects[22]. Similar to Spearman, Kendall rank correlation coefficient is another correlation measure for non-parametric data[10] as it compares the rankings of the variables instead of the variables themselves, although by nature Kendall's results usually show weaker correlations [5]. Below is the formula used by Netmark to calculate the Kendall rank correlation.

$$\tau = \frac{C - D}{\frac{1}{2}n(n-1)} \tag{3.2}$$

τ = tau (the Kendall rank correlation coefficient)
C = the number of concordant pairs
D = the number of discordant pairs
n = the total number of observations

To explain how the above equation (3.2) is utilized for this analysis : let's say we decided to compare Google's result(x) against the total number of Backlinks(y) of the ranked pages(see 'Position' and 'Backlinks' columns from Figure 3.1). When moving down the list, any pair of observations (x_i, y_i) and (x_j, y_j) are said to be concordant (C in equation 3.2) if the ranks for both elements agree: that is, if both $x_i > x_j$ and $y_i > y_j$ or if both $x_i < x_j$ and $y_i < y_j$. They are said to be discordant (D in equation 3.2), if $x_i > x_j$ and $y_i < y_j$ or if $x_i < x_j$ and $y_i > y_j$. If $x_i = x_j$ or $y_i = y_j$, the pair is neither concordant nor discordant. Now we have all the variables needed for equation 3.2, after computing Kendall correlation for each search query we average across all results to come up with the final result (mean Kendall correlation).

3.2.3 Rank Biserial Correlation

Biserial correlation refers to an association between a random variable X which takes on only two values(for convenience 0 and 1), and a random variable Y measured on a continuum [23]. Netmark performed an analysis based on Spearman and Biserial correlation. On their report [5] they argue that, for variables that are binomial in

[10]In statistics, the term non-parametric statistics refers to statistics that do not assume the data or population have any characteristic structure or parameters

nature (meaning only one of two results) the Rank-Biserial correlation coefficient is a preferred method of analysis. For example (see 'Position' and 'Search Term = Domain' columns from Figure 3.1), to compare Google's result (Y) with whether or not domain name (i.e. domain name of the raked pages) is an exact match of the search term (X) : the first step is to take the average rank of all observations that have X set to '1' (Y_1), next subtract the average rank of all observations that have X set to '0' (Y_2). Then the results are inserted into equation (3.3) to calculate Rank-Biserial correlation coefficient for each search term. Finally, the final result (mean Rank-Biserial correlation coefficient) is calculated by averaging across all search terms Rank-Biserial correlation coefficients.

$$r_{rb} = \frac{2(Y_1 - Y_2)}{n} \tag{3.3}$$

r_{rb} = Rank-Biserial correlation coefficient
Y_1 = the Y score mean for data pairs with an X score of 1
Y_2 = the Y score mean for data pairs with an X score of 0
n = the total number of data pairs

Their study is conducted on 939 search engine queries with 30 Google results pulled per keyword and 491 variables analysed per result. And it shows that off-page factors still have a much more higher correlation to ranking on Google than on-page factors. It also shows that there is still strong correlation between exact match of domain to the search query and ranking.

3.2.4 Variable Ratios

In mathematics, a ratio is a relationship between two numbers of the same kind(e.g., objects, persons, students, spoonfuls, units of whatever identical dimension), usually expressed as "a to b" or a:b, sometimes expressed arithmetically as a dimensionless quotient of the two that explicitly indicates how many times the first number contains the second (not necessarily an integer)[11].

To determine whether Google uses several filters for detecting unnatural[12] Backlinks and social profiles of webpages and websites, Netmark performed ratio analysis on different variables and compare those ratios to Google's search engine results. First they calculated the ratios by taking a variable as denominator (e.g. Page Authority) and several other variable as numerator (e.g. Number of Page Facebook Likes).

[11] `http://en.wikipedia.org/wiki/Ratio_analysis`

[12] Google defines unnatural links as "Any links intended to manipulate a site's ranking in Google search results. This includes any behavior that manipulates links to your site, or outgoing links from your site.

$$PageAuthorityRatio = \frac{NumberofPageFacebookLikes}{PageAuthority}$$

Then they used the resulting ratios to calculate Spearman correlation with Google's search rankings.

3.2.5 Normalization

In statistics, an outlier is an observation point that is distant from other observations. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the dataset[24]. When computing correlation coefficients[5] it is a common practice to normalize the raw data before averaging. In statistics, normalization means adjusting values measured on different scales to a notionally common scale, often prior to averaging[13].

If two variables are compared with a different order of magnitudes, a common way to standardize those variables is by computing a z-score for each observation[5]. The mathematical equation to do this is:

$$z = \frac{(x - \mu)}{\sigma} \tag{3.4}$$

z = the standardized score
x = raw data to standardize
μ = the mean
σ = the standard deviation

3.2.6 P Value

In statistical significance testing, the p-value is the probability of obtaining a test statistic result at least as extreme as the one that was actually observed, assuming that the null hypothesis is true[25].

A researcher will often "reject the null hypothesis" when the p-value turns out to be less than a predetermined significance level, often 0.05[26] or 0.01. Such a result indicates that the observed result would be highly unlikely under the null hypothesis.

P value is a statistical measure that helps scientists determine whether or not their hypotheses are correct. P values are usually found on a reference table by first calculating a chi square value.

[13] `http://en.wikipedia.org/wiki/Normalization_(statistics)`

3.2.7 Chi Square

Chi Square(written x^2) is a numerical value that measures the difference between an experiment's expected and observed values[14]. Netmark used chi square to calculate the correlation of factors that are categorized as limited, such as the Top-Level Domains(TLD). The equation for chi square is:

$$x^2 = \Sigma(\frac{(o-e)^2}{e}) \tag{3.5}$$

Where "o" is the observed value and "e" is the expected value.

3.3 Data Analysis Based Study

Evans [3] tried to identify the most popular techniques used to rank a webpage high in Google. The paper presents the results of a study on 50 highly optimized webpages that were created as part of a Search Engine Optimization competition. The study focuses on the most popular techniques that were used to rank highest in this competition, and includes an analysis on the use of PageRank, number of pages, number of in-links, domain age and the use of third party sites such as directories and social bookmarking sites. A separate study was made on 50 non-optimized webpages for comparison. This paper provides insight into the techniques that successful Search Engine Optimizers use to ensure that a page ranks high in Google. This work also recognizes the importance of PageRank and links as well as directories and social bookmarking sites.

Evans pointed out that the limitation of his work is that it only analyzed the top 50 web sites for a specific query (i.e. "V7ndotcom Elursrebmem"). Analyzing more web sites and comparing with similar studies in different competition would provide more concrete results. His study only analyzed 6 off-page factors, analyzing more factors and considering on-page factors could give more solid outcome.

3.4 Previously Analyzed Ranking Factors

In this section we present tables that summarizes the ranking factors analyzed by the academic papers and white papers reviewed in previous sections. In addition we give basic information about the datasets utilized in these researches.

[14] `http://www.wikihow.com/Calculate-P-Value`

TABLE 3.1: Basic statistics on the dataset used by Bifet et al. [1], Su et al. [2] and [3]

INFO	**Bifet (2005)**	**Su(2010)**	**Evans(2007)**
Search Terms	48	60	1
Webpages per Search Term	Top 100	Top 100	Top 50
Factors per Webpage	22	17	6
Search Engine	Google	Google	Google

3.4.1 Academic Papers

To help the reader get a quick glimpse on the amount of search terms, webpages and ranking factors analyzed by the papers reviewed we prepared table 3.1. Consequently, table 3.4.1 is prepared to show the similarity and disparity of the factors analyzed in these papers (Bifet et al. [1], Su et al. [2] and Evans [3]). To help the reader compare, factors that are similar are filled with same background color(non-white) and ordered accordingly. Only one factor is included in all there (i.e. PageRank), four matching factors used by [1] and [2], one matching factor used by [2] and [3] and two between [1] and [3].

TABLE 3.2: Comparing the factors used by Bifet et al. [1], Su et al. [2] and Evans [3] in their study.

#	**Bifet (2005)**	**Su(2010)**	**Evans(2007)**
1	PageRank of the page, or the approximation of the PageRank in a 0-10 scale obtained from Google's toolbar	PageRank score	PageRank of a web site
2	Fraction of terms in the documents which can not be found in an English dictionary	Age of the web site's domain (in years)	Age of the web site's domain name
3	Term is in the page's URL or not	Keyword appear in hostname (boolean)	Number of webpages in a site indexed by search engine
4	Number of pages linking to a page, in-degree approximated using Google API link: queries	Keyword in the path segment of url (boolean)	Number of in-links to a web site

Continued on next page

Table 3.2 – continued from previous page

#	Bifet (2005)	Su(2010)	Evans(2007)
5	Term is listed in a web directory or not	Size of the web site's domain(number of characters)	Listing in Yahoo and DMoz directories
6	Term as an attribute value (element/@attribute): IMG/@ALT, IMG@TITLE(N)	Keyword in image tag (boolean)	Number of pages listed in Del.icio.us
7	Term in the meta keywords or description	Keyword in meta-keyword tag (boolean)	–
8	Term in a special document zone including HTML tags: B, I, U, FONT, BIG, H1-H6, A,LI and TITLE	Keyword in the title tag of HTML header (boolean)	–
9	Average position of the query terms in the document = 1 at the beginning, 0 if at the end and in-between in the middle.	Keyword in meta-description tag (boolean)	–
10	Average matches of the query terms	Keyword density (percentage)	–
11	Closeness of terms in the query in the webpage (distance in number of terms , smallest windows containing all of them)	Keyword in h1 tag (boolean)	–
12	Anchor text term frequency	Keyword in h2 tag (boolean)	–

Continued on next page

Table 3.2 – continued from previous page

#	Bifet (2005)	Su(2010)	Evans(2007)
13	Similarity of the term to the document, in terms of vector space model. We compute it using the frequency of terms in documents and the inverse document frequency of each term.	Keyword in h3 tag (boolean)	–
14	Average term length	Keyword in h4 tag (boolean)	–
15	Term frequency of query term = Number of occurrences of the query term (averaged over the different query terms)	Keyword in h5 tag (boolean)	–
16	Term in capitals in the page	Keyword in anchor text (boolean)	–
17	Number of bytes of text in the original document	Age of the webpage (in years)	–
18	Number of different terms of the document	–	–
19	Number of out-links in the page	–	–
20	Fraction of out-links to external Web sites	–	–
21	Number of bytes of the original document	–	–
22	Relative frequency of the more frequent term, i.e.: term frequency. If a document has 3 words and the most frequent word repeats 2 times,then this is 2/3	–	–

TABLE 3.3: Basic statistics on the dataset used by SearchMetrics, Moz and Netmark

INFO	**SearchMetrics(2013)**	**Moz(2013)**	**Netmark(2013)**
Search Terms	10,000	14,641	939
Webpages per Search Term	Top 30	Top 100	Top 30
Factors per Webpage	>44	>80	491
Search Engine	Bing US	Google	Google

3.4.2 White Papers

The companies mentioned above make their own white papers of ranking factors based on their own study, and summarized information about search terms and webpages analyzed is provided on table 3.3. The table bellow shows the top 10 factors suggested by SearchMetrics, Moz and Netmark aiming to clarify the similarity and disparity of the ranking factors. 8 of the factors from SearchMetrics lay under the category of social signal where as Moz do not have any social signal on it's top 10. When we look at Netmark most of the factors on the top 10 are related to Backlink, and have one social signal on the 5th place. Having Google+1 page is suggested by both SearchMetrics and Netmark as a major factor, other than that there are no matching factors between them.

Table 3.4: Comparison of the top 10 factors suggested by SearchMetrics, Moz and Netmark for Google.com in 2013

#	**SearchMetrics**	**Moz**	**Netmark**
1	Google +1	Keyword in title	Exact Match Domain
2	Facebook Shares	Organic anchor text distribution	Keyword Anchor Text Links from Domains to Page
3	Number of BackLinks	Number of Links Domains to page	Keyword Anchor Text Links to Page
4	Facebook Total	Keywords on page	Page Authority
5	Facebook Comments	Number of Links Domains to domains	Page Google+1s
6	Facebook Likes	Linking page relevance	Linking Domains to Page
7	Pinterest	Domain Trust	Linking Sub-domains to Page
8	Tweets	PageRank	"No Text" Anchor Text Links from Domains to Page
9	Percentage of Back-Links with rel=nofollow	Number of Links with with partial match	External Links to Page
10	SEO-Visibility of back-linking URL	Link type diversity	"No Text" Anchor Text Links to Page

3.5 Learning to Rank Benchmark Datasets

In an effort to understand the format of the datasets that are widely used for learning to rank studies, we reviewed 3 of the most popular benchmark datasets nowadays. The datasets are namely LETOR, Yahoo and Microsoft datasets. All the datasets contains a set of query-document pairs, represented by 46 up to 700 ranking features, and relevance judgments is provided by professional annotators.

Table 3.5 provides a summarized information on the number of queries, documents, relevance levels, features and year of release of each dataset.

TABLE 3.5: Characteristics of publicly available benchmark datasets for learning to rank

Dataset	**Queries**	**Doc.**	**Rel.**	**Feat.**	**Year**
LETOR 4.0	2,476	85 k	3	46	2009
Yahoo	36,251	883 k	5	700	2010
Microsoft	31,531	3,771 k	5	136	2010

As mentioned in Chapter 1, the common problem observed on all these datasets is that, the queries and URLs are not disclosed, instead these informations are represented by id. The LETOR dataset includes the list of features where as Yahoo has not disclosed the features, similarly Microsoft has only provided a high level description of the features category. The main reason for keeping the features closed is : since features are major component of any learning to rank system, it is important for search engines like Yahoo and Bing to keep the features secrete. Similarly disclosing the queries and URLs could also lead to a risk of reverse engineering the features used by their ranking system which then will be used by SEO community to weaken the effectiveness of the ranking system[15].

We found LETOR4.0-MQ2008-list version of the LETOR dataset very convenient for our study. The main reason for selecting this dataset is the availability of clear definition of the feature sets. In addition this dataset is constructed particularly to train models using listwise learning to rank techniques, which is a perfect fit to our goal. Details regarding how this particular dataset is constructed are briefly presented in Appendix A.

3.6 Summary

To sum up we have reviewed 3 academic papers published focusing on identifying ranking factors that have high correlation with well ranking webpages. In addition to that we

tried to review studies from 3 companies on the SEO industry, one is survey based and the other two are empirical researches. Overall, from this review we learn that, the suggestions made by academic researchers and SEO experts on the issue related to the topic is conflicting, which makes it confusing for Webmasters to implement one. What is more frustrating is that the Internet marketing advice we hear often is not substantiated with any empirical evidence. Another weakness is none of these studies have conducted a similar research which is optimized to The Netherlands. Which tells us there is a gap to be filled, and we believe it deserves a self standing research.

Chapter 4

Datasets and Ranking Factors

In this chapter we discuss the search engine, ranking factors and data source selected for the research. First details about the data gathering is provided, then rules and procedures followed to extract the required factors is elaborated. The goal is to give fellow researchers insights on how to repeat the data gathering, feature extraction process, what challenges to expect and the infrastructures required.

4.1 Data Gathering

The flow diagram on figure 4.1 depict the activities and data involved while making DUTCH WEB dataset. Each major activities and data are discussed in detail next to that.

FIGURE 4.1: Data gathering process flow.

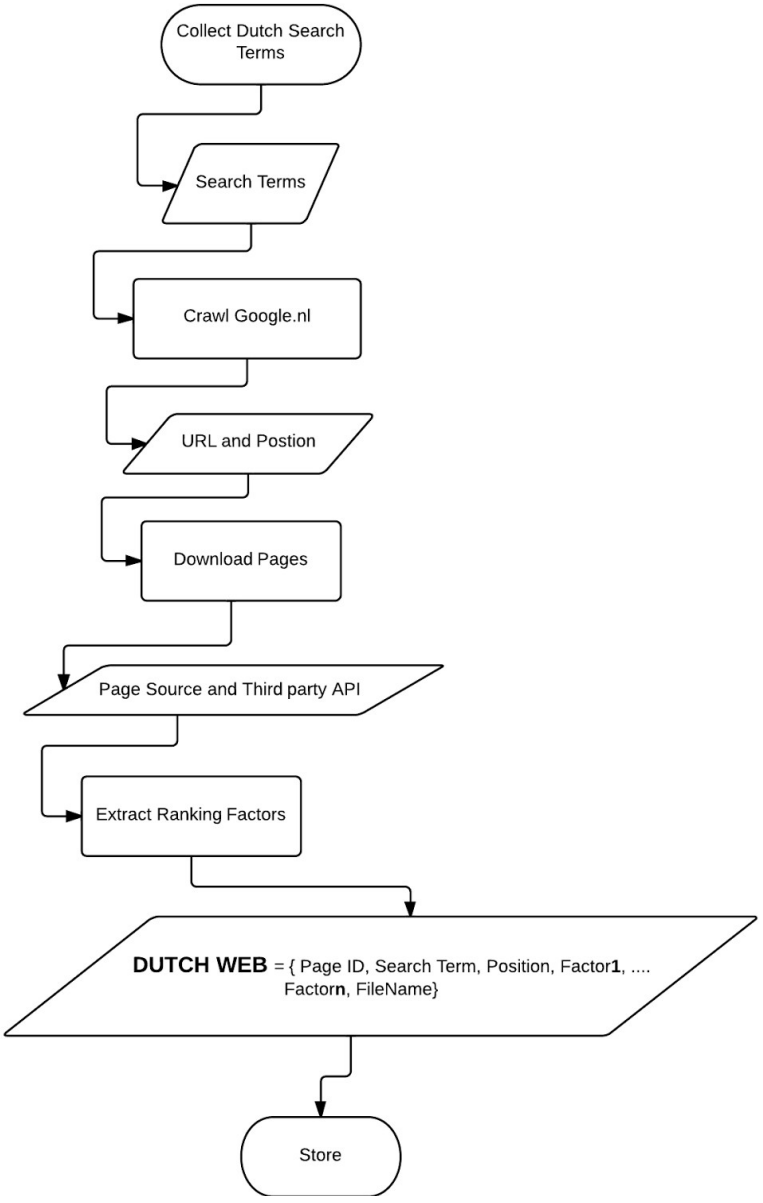


TABLE 4.1: Count of search terms grouped by the number of words they contain

# WORDS	COUNT	EXAMPLE
1	2009	slaapboek
2	3589	vacature automotive
3	1507	zwarte woud wandelen
4	370	50 jarig huwelijksfeest schip
5	75	appartement te koop de panne
6	12	wat zit er in een pijp
7	5	finanacial controller jobs in english in netherlands
9	1	financial jobs for italian native speaker in the netherlands
Total	7568	

4.1.1 Collecting Search Terms

Since the case study of this research is focused on The Netherlands, the first step is to collect Dutch search terms. Initially a total of 10,000 unique Dutch search terms are fetched from an existing proprietorial repository at Indenty. This repository contains Dutch search terms that were collected from campaigns of customers' websites in more than 10 years. After removing duplicates by using exact string matching and some additional data cleaning we end up with 7568 unique search terms. The table 4.1 shows the count of search terms grouped by the number of words they contain. For instance the first row shows search terms that contain only one word are 2009. As mentioned early the search term dataset is mainly of Dutch language, but very few English search term like the last two rows from the table above are also included in it.

4.1.2 Selecting Search Engine

After collecting the necessary search terms, the next step was to choose a search engine, to submit the queries in to. Once again since the case study of this research is for The Netherlands, the main criteria was to find a search engine that is focused on search queries originating from this region. The second criteria was popularity of the search engine. Last, the proprietary "Ranking Scanner" at Indenty (see Section 4.1.3) should have already implemented a search result page scrapper for the search engine.

We choose Google.nl, the Google version for The Netherlands, because it is dedicated to handle search queries originating from this region. Most importantly, Google is by far the largest search engine, and that's not only the case in the U.S., but worldwide[4]. The illustration on figure 4.2 was made by SearchMetrics in 2012, and shows that Google still holds an approximate share of 78% among other search engines. In addition the "Ranking Scanner" already supports scrapping search result pages from Google.nl.

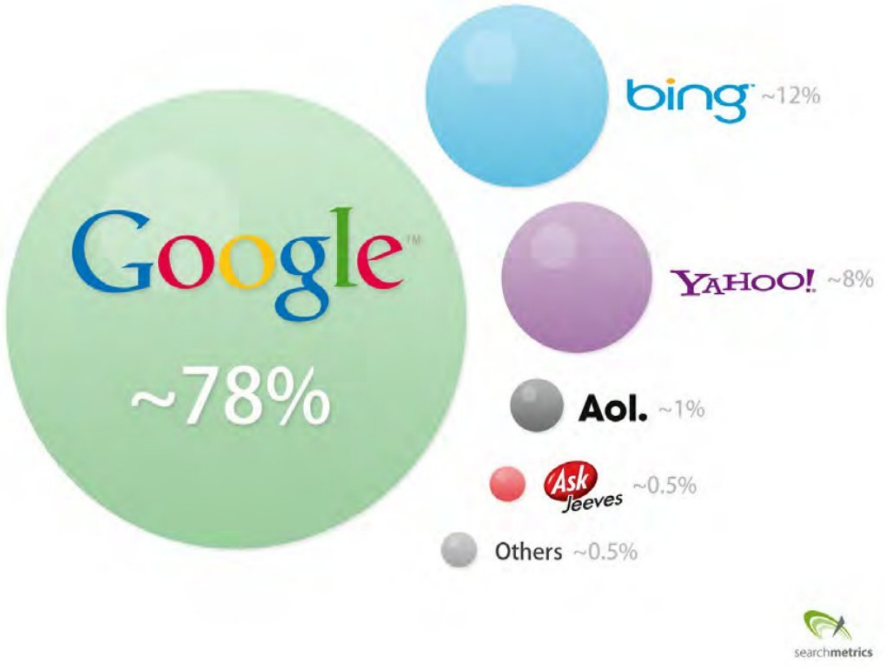


FIGURE 4.2: Illustration of current search engines worldwide share made by SearchMetrics.

4.1.3 Scan SERPs and Fetch Ranked Pages

We used Indenty's proprietorial crawler called "Ranking Scanner" to fetch the ranked webpages on the first 4 SERPs (top 40 pages) of every search query. On Google a single SERP often lists 10 organic search results, but sometimes this number could be lower

for brand searches. The crawler gathered URLs of exactly top 40 ranked webpages for 93% of the search terms. Search terms with less than 40 ranked pages are removed from the dataset to keep it simple and to avoid possible bias.

4.1.4 Download Webpages

To conduct this research it was necessary to download the ranked webpages on specific period of time. It is also possible to extract the desired factors from the ranked webpages online, however the web is so dynamic and the content of webpages and their rank changes instantly. For this reason, we downloaded the top 40 ranked webpages (i.e. the first 4 consecutive SERPs each with maximum of 10 results) for a total 7568 unique Dutch search terms over a period of one month (from 01 to 30 February, 2014). The final dataset contains approximately 302720 webpages with 21.6 GB size. As mentioned previously, this dataset is referred as DUTCH WEB dataset in this document. It must be noted that similar results from other SEO companies such as SearchMetrics reports cannot be compared on a perfectly common basis because the data gathering was executed at different times.

4.2 Extracting Ranking Factors

Here we give specifics of how the ranking factors used in this research are computed. In addition brief definition of some key terms and concepts are provided.

Although Google claims to use 200 features, we analyzed a set of only 52 factors. The intention is to find out the relationship between these identified factors and the well ranked pages on Google's search result page. The process of selecting these factors involves gathering ranking factors suggested by the top three SEO companies at this time(SearchMetrics, SEOMoz, and NetMark), and previous academic researches. In addition some factors suggested by experts in the field are also added to the list. After mixing up all and removing duplicates the final list contains 52 unique factors.

For clarity purpose the factors are categorized in to two main streams: on-page and off-page factors. On-page factors are further grouped in to content and coding factors. Ranking factors related to coding of the webpage are not investigated in this research, because we believe that search engines are not looking into technical quality any more since most websites are using a highly optimized frameworks such as WordPress. The off-page factors are also split in to backlink and social signal.

Before going further, several general remarks need to be outlined. First, most of the factors are computed at the webpage level, but some of them are computed at domain level. Second, some count factors (Continuous variables) are aggregated in a sensible way. For instance, instead of counting the existence of search term in <B>, <I>, and <Strong>tags separately we compute sum of the counts and form a new composite feature.

Note that, not every factor that is included in this research is discussed here. Factors that are very common and repetitive are not discussed to avoid duplication in the document. Another remark is, since all factors cannot be measured with one method, we have classified them into three categories.

- Limited Factors : Limited factors are variables that can take one out of fixed number of possible values. For example the TLD extension of a domain can be eu, nl, net or com.
- Dichotomous Factors : Dichotomous factors also known as binary factors are variables that can take either yes or no for a value. Example EMD and PMD can take yes or no.
- Unlimited Factors : Unlimited factors also known as continuous factors are variables that can take value from 0 upto infinity. For example the number of times a search term appear inside the text between <Body>tags is unlimited.

4.2.1 URL Related Factors

URL related factors are factors that are extracted from the URL of each webpage. A total of 12 variables are extracted for each URL on the DUTCH WEB dataset. A detailed description of all the 12 variables will be presented in this section. Earlier to that some terms which are used in the context are briefly discussed.

What is Uniform Resource Locater ?

A uniform resource locater, abbreviated as URL (also known as web address, particularly when used with HTTP), is a specific character string that constitutes a reference to a resource. An example of a typical URL would be "`http://www.utwente.nl/onderwijs/websites_opleidingen/`". A URL is technically a type of Uniform Resource Identifier (URI). URLs are commonly used for webpages (HTTP), but can also be used for file transfer (ftp), email (mailto) and many other applications[27]. Before discussing the

factors related to URL, it is important to understand the basic parts that make up a URL. Some of the basic parts of URL are described below:

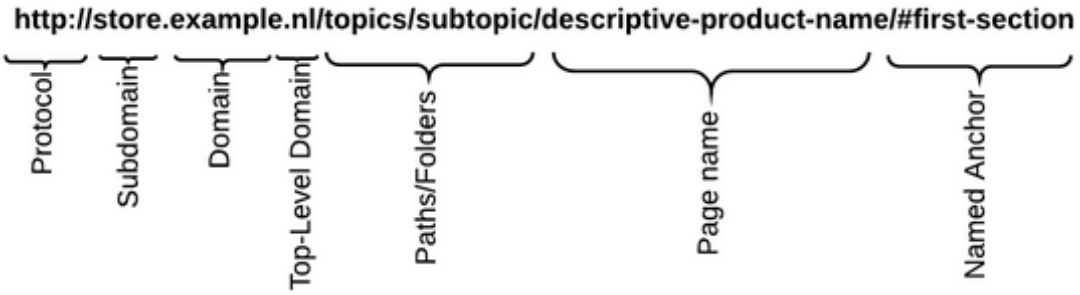


FIGURE 4.3: The structure of SEO friendly URL

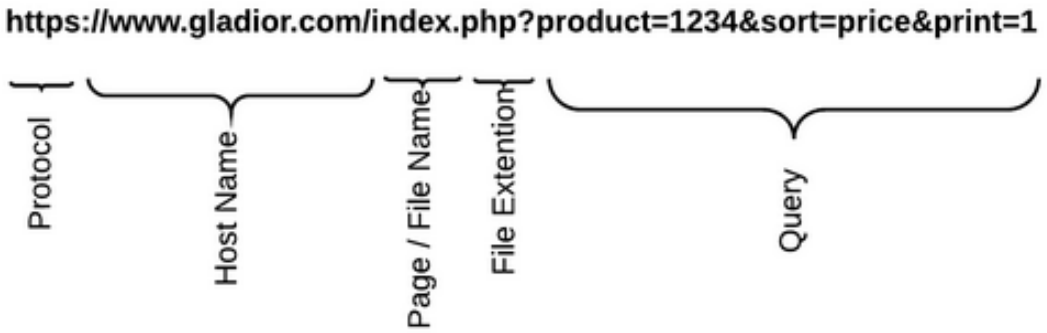


FIGURE 4.4: The structure of old dynamic URL.

- Protocol : Each URI begins with a protocol identifier and ends by a colon and two forward slashes. The protocol identifier indicates the name of the protocol to be used to fetch the resource. For example Hypertext Transfer Protocol (HTTP) is typically used protocol to serve up hypertext documents. HTTP is just one of many different protocols used to access different types of resources on the net. Other protocols include Hypertext Transfer Protocol Secure (HTTPS), File Transfer Protocol (FTP), Gopher, File, and News [1]. Although protocols are case insensitive, the canonical form is lowercase[27]. An implementation should accept uppercase letters as equivalent to lowercase in protocol identifiers (e.g., allow "HTTP" as well as "http").

- Domain Name : Domain names serve as more easily memorable names for Internet resources such as computers, networks, and services. A domain name consists of one or more parts, technically called labels, that are conventionally concatenated, and delimited by dots, such as example.com. The only valid characters for a domain name are letters, numbers and a hyphen "-". Other special characters like

[1] http://docs.oracle.com/javase/tutorial/networking/urls/definition.html

the underscore or an exclamation mark are NOT permitted. For example: `your_name.com` is incorrect because it contain underscore, where as `your-name.com` is a correct domain name.

- Top-Level Domain : The top-level domains (TLDs) such as com, net and org are the highest level of domain names of the Internet. The right-most label of every domain name conveys the top-level domain; for example, the domain name www.example.com belongs to the top-level domain com. In other words, every domain name ends with a top-level domain label. The list below summarizes this:
  - URL: http://www.example.net/index.html
  - Top-level domain name: net
  - Second-level domain name: example.net
  - Hostname: www.example.net
- Second-Level and Lower Level Domains : Below the top-level domains in the domain name hierarchy are the second-level domain (SLD) names. These are the names directly to the left of .com, .net, and the other top-level domains. As an example, in the domain example.co.uk, co is the second-level domain. There can be third, fourth, fifth-level domains, and so on, with virtually no limitation.
- Subdomain : The hierarchy of domains descends from the right to the left label in the name; each label to the left specifies a subdivision, or subdomain of the domain to the right. For example: example.com is a subdomain of the com domain, and in www.example.com www is a subdomain of example.com. The full domain name may not exceed a total length of 253 ASCII characters in its textual representation[28].
- Hostname : A hostname is a domain name that has at least one associated IP address. For example, the domain names www.example.com and example.com are also hostnames, whereas the com domain is not. However, other top-level domains, particularly country code top-level domains, may indeed have an IP address, and if so, they are also hostnames. Hostnames impose restrictions on the characters allowed in the corresponding domain name. A valid hostname is also a valid domain name, but a valid domain name may not necessarily be valid as a hostname.
- Path/Folders : The path or folder (see Figure 4.3) component contains data, usually organized in hierarchical form, that, along with data in the non-hierarchical query component serves to identify a resource within URI. The path is terminated by the first question mark ("?") or number sign ("#") character, or by the end of the URI. A path consists of a sequence of path segments separated by a slash("/") character.

- Query : The query (see Figure 4.4) component contains non-hierarchical data that, along with data in the path component serves to identify a resource within the scope of the URI. The query component is indicated by the first question mark ("?") character and terminated by a number sign ("#") character or by the end of the URI. The characters slash ("/") and question mark ("?") may also represent data within the query component.

- Fragment/Named Anchor : The fragment identifier component also know as 'Named Anchor' by SEO experts (see Figure 4.3) is indicated by the presence of a number sign ("#") character and terminated by the end of the URI.

- Public Suffix : A public suffix also know as eTLD[2] is one under which Internet users can directly register names. Some examples of public suffixes are .com, .co.uk and pvt.k12.ma.us. With Mozilla's initiative we now have a publicly open database of top-level domains (TLDs) including the respective registry's policies on domain registrations at different levels[3], named the Public Suffix List(PSL). This list helps in finding the highest level at which a domain may be registered for a particular top-level domain.

  There is a major confusion between "TLD" and "Public Suffix". The source of this confusion is that people tend to say "TLD" when they mean "public suffix". However, these are independent concepts. So, for example,

  - uk is a TLD, but not a public suffix
  - co.uk is a public suffix, but not a TLD
  - squerf is neither a TLD nor a public suffix
  - com is both a TLD and a public suffix

Below, we discuss URL related factors extracted with example and the constraints considered.

1. URL EXACT MATCH TO DOMAIN : One of the latest changes to the Google search algorithm is something known as Exact Match Domain (also known as Keyword Domains) abbreviated as EMD. Exact Match Domain (will be referred as EMD from here below) imply to the use of a possible search term for your website's domain. For instance, if `www.restaurantsinenschede.nl` is a website's domain, "Restaurants In Enschede" is an exact phrase that visitors might type into Google.nl when looking for restaurants around Enschede. While it might not

[2]Effective Top-Level Domain (eTLD) is a deprecated synonym of Public Suffix
[3]`https://wiki.mozilla.org/Gecko:Effective_TLD_List`

be deliberate, there are companies out there that name their website's domain to match a certain search term in order to get a better ranking on search engines, even though they have poor content. Google sees this as unfair advantage over other sites that offer legitimate content without using this practice. In 2013 Google announced that they have updated their algorithm with EMD to penalize low quality websites. We wanted to see how Google's EMD update is affecting the Dutch web on our dataset by analyzing the "URL EXACT MATCH TO DOMAIN" dichotomous factor.

The string matching we used for this factor is quite strict. We set the EMD factor as "true" only if all the words in a search term are also found on the domain name of the ranked webpages, concatenated in the same order, with or without a hyphen "-", and case insensitive string matching. Below, some examples of EMD are given:

(a) If search term = "iphone" and domain = `www.iphone.com` then this is an EMD.

(b) If search term = "t mobile" and domain = `www.t-mobile.com` then this is an EMD.

(c) If search term = "q able" and domain = `q-able.com/` then this is an EMD.

2. URL PARTIAL MATCH TO DOMAIN : Partial Match to Domain abbreviated as PMD is a domain name combined of search term and other extra information (e.g. business name, location etc). The string matching we used for EMD is quite strict, so we decided to check the PMD with more flexible string matching rules. For instance if the search term contains two words, and the domain name contains at least one of these words, then we say there is a partial match between the search term and the domain name. The following examples elaborate PMD :

   (a) search term : "iphone" , domain : www.**iphone**club.com : this is a PMD.

   (b) search term : "iphone house" , domain : www.phone**house**.nl : this is a PMD.

   Percentage of URLs in the DUTCH WEB dataset which have "true" value for EMD and PMD is provided in Appendix C.5.

3. URL HAS PUBLIC SUFFIX : As mentioned earlier it is useful to determine whether a given domain name might represent an actual domain on the Internet. For this purpose, we use data from the Public Suffix List (PSL). More precisely, if domain.hasPublicSuffix() returns "true", then the domain might correspond to a real Internet address otherwise, it almost certainly does not relate to actual domain. For example, URLs which have IP address as domain name (e.g.

`http://195.193.209.12/subdomain/document.pdf`) often return "false" for this check. Similarly URLs which are pointing to pages with error (e.g. "HTTP 404 : Page Not Found") or simply invalid URLs also returned "false" for this check.

4. URL DOMAIN IS VALID : Indicates whether the domain is a syntactically valid domain name using lenient validation. Specifically, validation against RFC 3490 [4] ("Internationalizing Domain Names in Applications") is skipped. Some URLs, even if they does not have a valid public suffix happen to have a valid domain (i.e. URL DOMAIN IS VALID was set to "true"). On the other hand all URLs with invalid domain were immediately treated as if they don't have public suffix (i.e. URL HAS PUBLIC SUFFIX was set to "false").

5. URL SEARCH TERM OCCURRENCE COUNT IN URL PATH : This factor simply contains the number of occurrences of a search term in the URL's path part. This particular URL's path from our dataset (`/technology-built-environment/electrotechnology/welcome_electrotechnology/welcome_electrotechnology_home.cfm`) contains the search term "electrotechnology" 3 times. The string matching is configured both for exact match and partial match.

6. URL DEPTH : Depth of a URL is determined by counting the number of "/" in it's path part. The following two URLs have URL DEPTH of 1 : `http://www.youtube.com/playlist?list=PLxzMZI`, and `http://www.sub-domain.com/`. While the following has a URL DEPTH of 2 : `http://nl.wikipedia.org/wiki/seo`.

7. URL PROTOCOL : HTTP (HyperText Transfer Protocol) is the basic transport protocol of the Web. The Web is a client/server system and some mechanism needs to exist to move data between servers and clients. HTTP provides that mechanism. Most Web browsers also uses other protocols, such as FTP, but the vast majority of Web traffic is moved by HTTP. On the other hand nearly all secure web communication takes place over HTTPS including online banking, e-mail, and e-commerce transactions[29]. As of 2013-09-02, 24.6% of the Internet's 168088 most popular web sites have a secure implementation of HTTPS[30]. In addition, Google's Matt Cutts[5] said he would like to give ranking boost to websites that have implemented SSL[6]. All these indicate that website security is still a hot topic : and we suspect that search engines might prefer webpages that have implemented HTTPS, and rank them on the top. Motivated by this, we conducted an analysis to find out if there is any kind of relationship between URLs protocol (HTTP or HTTPS) and it's rank and the result are presented in Appendix C.2.1.

[4] `http://www.ietf.org/rfc/rfc3490.txt`
[5] `http://www.mattcutts.com/blog/(Jun10,2014)`
[6] `http://www.seroundtable.com/google-ssl-ranking-18256.html`(June 10, 2014)

8. URL PUBLIC SUFFIX : This factor holds the public suffix of each URL.

9. URL TOP PRIVATE DOMAIN : This factor holds the TLD of each URL. The top 25 most abundant TLDs as well as eTLDs are given in Appendix C.3.

4.2.2 Social Media Links On Page

This section provides details about factors which are related to social media in the DUTCH WED dataset. We give a special emphasis to the social media influence on webpage's ranking because of the fact that the number of social media activity has been growing rapidly in recent years. Visitors like to share and recommend contents with their own network. This makes it important to have "Share", "Like Us", "Recommend" etc links/buttons on company's websites and blogs. All the factors included in this category are on-page factors : which are features that can be found and extracted from the content and coding of webpages. It was not feasible to collect all the factor that was initially planned, mainly because of technical barriers and partially due to time limitation. We checked links for Facebook fan page, Google Plus fun page, Twitter page, LinkedIn page etc. Since the factors are extracted in a similar fashion only one of them is discussed below.

1. FACEBOOK FOLLOW US LINK ON SITE : It is no secret that Facebook has become a major traffic driver for all types of websites. And we believe there are Facebook "Like" and "Recommend" widgets on almost every website. We wanted to see the relationship between this factor and a webpage's position on Googl.nl. Gathering links from a website that are pointing to the websites's Facebook fun page is bit tricky. First of all, the plugin code for the "Like" or "Follow Us" widgets come in HTML5, XFBML, IFRAME and URL formats. Hence, a check for all the formats is implemented, then duplicated and black listed links are removed. The following box shows the checks implemented for different format of the "Like" button on a website using Jsoup in Java.

```
HTML5 : doc.select("[data-href*=www.facebook.com]")
XFBML : doc.select("[href*=www.facebook.com]")
IFRAME : doc.select("iframe[title*=Facebook][src*=www.facebook.com]")
```

4.2.3 Markups

Some of the markup related factors we extracted are briefly discussed here.

1. GOOGLE PLUS PUBLISHER MARKUP ON SITE : One advantage of having this markup on a website is that when branded search queries are made in Google, the Knowledge Graph (see Section 2.3) for that query should display a widget for that brand's corresponding Google+ page which includes: the name of the Google+ page, number of followers of the page, last update from the page and an option to follow the page for logged in users. Checking for this markup on the page source of a website was pretty much straight forward as it is implemented only in the following two formats.

```
<link href="[Google+ Page URL]" rel=publisher />
<a href="[Google+ Page URL]" rel="publisher">Google+</a>
```

2. TWITTER CARD MARKUP ON SITE : With Twitter Cards, one can attach rich photos, videos and media experience to Tweets that drive traffic to his/her website. By adding a few lines of HTML to a webpage, users who Tweet links to the content of that webpage will have a "Card" added to the Tweet that is visible to all of their followers. The following snippet shows how we extracted this markup from the page source of a website using Jsoup in Java.

```
doc.select("meta[name=twitter:card]")
```

3. FACEBOOK OPEN GRAPH MARKUP ON SITE : Facebook's Open Graph protocol allows for web developers to turn their websites into Facebook "graph" objects, allowing a certain level of customization over how information is carried over from a non-Facebook website to Facebook when a page is "recommended", "liked", or just generally shared. All of Facebook's Open Graph META tags are prefixed with *og* : then continued with the specific property to be set.

4. NO FOLLOW MARKUP ON SITE : This markup provides a way for webmasters to tell search engines "Do not follow links on this page"[7]. More importantly, instead of telling search engines and bots not to follow any links on the page, it lets webmasters to easily instruct robots not to crawl a specific link. For example:

```
<a href="signin.php" rel="nofollow">sign in</a>
```

   Would it affect ranking if too many of this markup are discovered on the links in a webpage. To answer this question we analyzed this factor by counting the number of times it occurred in a page.

[7] https://support.google.com/webmasters/answer/96569?hl=en

4.2.4 Content Related Factors

We performed two fundamental investigation on the scraped content of the ranked pages for a given search query. First we check the existence of the search keyword or phrase on different tags and then count the total number of words with in these tags. Below is a table with the full list of content related factors and their description.

TABLE 4.2: On-page factors, content related

#	Factor	Category	Description
1	Title word count	Unlimited	number of words with in <title>tag.
2	Body word count	Unlimited	number of words with in <body>tag.
3	Meta description word count	Unlimited	number of words with in <meta name="description">tag.
4	Meta keyword word count	Unlimited	number of words with in <meta name="keywords">tag.
5	Sum word count in <b>, <i>and	Unlimited	number of words with in <b>,<i>and tags.
6	Search term in <title>tag	Dichotomous	a boolean that is set to true if the search term exists with in the <title>tag.
7	Search term in <body>tag	Dichotomous	a boolean that is set to true if the search term exists with in the <body>tag.
8	Search term in meta description tag	Dichotomous	a boolean that is set to true if the search term exists with in the meta description tag.
9	Search term in meta keyword tag	Dichotomous	a boolean that is set to true if the search term exists with in the meta keyword tag.
10	Search term in <p>tag	Dichotomous	a boolean that is set to true if the search term exists with in the <p>tag.
11	Search term in <a>tag	Dichotomous	a boolean that is set to true if the search term exists with in the <a>tag.

Continued on next page

Table 4.2 – continued from previous page

#	Factor	Category	Description
12	Search term in or tags	Dichotomous	a boolean that is set to true if the search term exists with in the or tags.
13	Search term in the <b>or <i>or <strong>tags	Dichotomous	a boolean that is set to true if the search term exists with in the <b>or <i>or <strong>tags.
14	Search term in image alt tag	Dichotomous	a boolean that is set to true if the search term exists with in the alt image tag.
15	Search term in <h1>tag	Dichotomous	a boolean that is set to true if the search term exists with in the <h1>tag.
16	Search term in <h2>tag	Dichotomous	a boolean that is set to true if the search term exists with in the <h2>tag.
17	Search term in the <h3>tag	Dichotomous	a boolean that is set to true if the search term exists with in the <h3>tag.
18	Search term in the <h4>tag	Dichotomous	is a boolean that is set to true if the search term exists with in the <h4>tag.
19	Search term in the <h5>tag	Dichotomous	is a boolean that is set to true if the search term exists with in the <h5>tag.
20	Search term in the <h6>tag	Dichotomous	is a boolean that is set to true if the search term exists with in the <h6>tag.

4.2.5 Backlinks and Outlinks Related Factors

This type of factors tries to determine the quality or the popularity of a webpage/website based on its connectivity in the web. The factors we managed to collect are backlinks and outlinks. Backlinks, also known as incoming links, inbound links or inlinks are incoming links to website or webpage. Backlinks are what Kleinberg called Authority

on his HITS algorithm[31], a good authority represented a page that was linked by many different hubs. MajesticSEO api allow access to several Backlink related information. It is almost disappointing that we are not able to gather the famous PageRank, for understandable reasons (see Chapter 5).

For outlinks, we extracted three factors : links pointing to internal pages (i.e. with same domain name), links pointing to external pages, and total number of all links found in page.

TABLE 4.3: Backlinks and outlinks related factors

#	**Factor**	**Category**	**Description**
1	Number of Backlinks	Unlimited	the total number of inbound links of the website.
2	Indexed URLs	Unlimited	the total number of indexed URLs of the website
3	Referring IP addresses	Unlimited	the total number of IP addresses referring to the website. This factor is good to have to avoid link spam farms.
4	Referring domains	Unlimited	the total number of unique domains referring to the website.
5	Internal Links on Page	Unlimited	number of links found on a page that point to internal pages (pages of same website).
6	External Links on Page	Unlimited	number of links found on a page that point to other websites.
7	All Links on Page	Unlimited	total number of links found on a page.

4.3 Summary

To sum up this part of this document is written to report the data gathering and feature extraction process followed. A total number of 7568 unique Dutch search terms are considered in this research. Top 40 ranked webpages for each search term on Google.nl

are scanned and 52 factors are extracted from each webpages. The majority of the ranking factors extracted are on-page factors, and few off-page factors. We have extracted URL related factors, social media links related factors, markup related factors, and backlink/outlink related factors. We discussed in detail the motivation for extracting a certain factor, and provide enough information to repeat the process.

Chapter 5

Algorithms and System Design

The goal of this chapter is, first to clearly present the mathematical definition of the rank correlation coefficient equations utilized in this research. Then, to discuss the algorithms designed to implement these equations and the constraints considered. Lastly, to briefly discuss the high level system design, the technologies used and the challenges encountered.

5.1 Correlation Coefficients

When ever there is a need to know if two variables are related to each other, we usually compute a correlation coefficient. The "correlation coefficient" was coined by Karl Pearson in 1896. Accordingly, this statistic is over a century old, and is still going strong. It is one of the most used statistics today, second to the mean[32]. The correlation coefficient, denoted by r, measures the strength and the direction of a linear relationship between two variables.

The following points are the accepted guidelines for interpreting the correlation coefficient:

- Range : Theoretically r can be any value in the interval between +1 and -1, including the end values ± 1.
- Positive correlation : A positive coefficient indicates that two variables systematically vary in the same direction : as one variable increases in its values, the other variable also increases in its values. When the value of r is closer to +1, it means stronger positive association.

- Negative correlation : A negative coefficient indicates that two variables systematically vary in opposite directions: as one variable increases in its values, the other variable decreases in its values. When the value of r is closer to -1, it means stronger negative association.
- No correlation : If value of r is 0 then it indicates no linear relationship. In other words, there is random, nonlinear relationship between the two variables.
- Perfect correlation : A perfect correlation of ±1 occurs only when all data points lie exactly on a straight line. If r = +1, the slope of this line is positive. If r = -1, the slope of this line is negative.

5.1.1 Spearman Rank Correlation

Spearman's rho yields a correlation coefficient between two ordinal, or ranked,variables[1]. The equation for Spearman rank correlation denoted by ρ is given as follow.

$$\rho = 1 - \frac{6\Sigma d_i^2}{n(n^2-1)} \tag{5.1}$$

ρ = rho (the correlation coefficient)
d_i = the differences between the ranks ($d_i = x_i - y_i$)
n = the total number of observations.
the number "6" is a constant.

Suppose we want to compute the relationship of a webpage's position (SET A) on Google.nl and it's total number of backlinks (SET B) by calculating Spearman rank correlation coefficient. The first step is to compute a relative natural rank for each position (RANK A) and same for the backlinks (RANK B). Then we compute the difference between the ranks(d), square the differences(d^2), sum the squares(Σd^2) and substitute into equation 5.1. The table below summarizes the process.

TABLE 5.1: Example of calculating Spearman rho on sample data.

SET A	**RANK A**	**SET B**	**RANK B**	**d**	d^2
1	5	2314450	4	1	1
2	4	3416540	3	1	1
3	3	1234120	5	2	4
4	2	12341434	2	0	0
5	1	23453445	1	0	0

[1] `http://www.napce.org/documents/research-design-yount/22_correlation_4th.pdf`

$\rho = 1 - \frac{6(1+1+4+0+0)}{5(25-1)} = 0.7$

In this example, the position of each webpage is directly used in the calculation however this approach is wrong. Position of webpages is often sorted in ascending order: the most relevant page gets position 1, the least relevant page gets a position equal to the total number of ranked pages. Spearman ranker on the other hand, only considers the magnitude of a number and it assigns the highest rank(i.e. 1^{st}) to the largest number: in the example the page at position 5 is assigned the first rank. *Therefore, a rank for each webpage is calculated using the position of the page on Googl.nl before feeding it to equation 5.1 : in the table above* 1^{st} *column (SET A) will be replaced by the* 2^{nd} *(RANK A) and vise versa. In that case the result of the above example would be the exact negative (i.e. -0.7).*

Algorithm

The implementation of algorithm 1 performs a rank transformation on the input data and then computes Pearson's Correlation on the ranked data, which yields exactly same result as equation 5.1. By default, ranks are computed using natural ranking(see Section 5.1.5) with default strategies for handling NaNs and ties in the data (NaNs maximal, ties averaged). It takes input arrays which holds, position of the webapages and value of a particular feature for every webpage. The size of the both arrays must be equal, at same time greater or equal to minimum number of ranked webpages(MIN_PAGES) and less than or equal to maximum number of ranked pages(MAX_PAGES). If the NaN strategy is set to REMOVE, then all of NaN elements in both arrays are removed. Then a natural rank for each element of both arrays is computed using rank() function before invoking the PearsonsCorrelation() method.

Algorithm 1 Spearman Rank correlation algorithm

```
1: procedure SpearmanRank(xArray[], yArray[])   ▷ xArray holds rank, yArray holds feature value
2:     n_x ← xArray.length
3:     n_y ← yArray.length
4:     if n_x ≠ n_y then
5:         throw error                          ▷ dimension mismatch exception
6:     else if n_x < MIN_PAGES ‖ n_x > MAX_PAGES then
7:         throw error                          ▷ insufficient dimension exception
8:     else
9:         if NaNStrategy = REMOVE then
10:            nanPositions ← getNaNPositions(xArray, yArray)
11:            xArray ← removeValues(xArray, nanPositions)
12:            yArray ← removeValues(yArray, nanPositions)
13:        end if
14:        return PearsonsCorrelation(rank(xArray), rank(yArray))
15:    end if
16: end procedure
```

5.1.2 Rank Biserial Correlation

The rank biserial correlation coefficient is computed between one continuous (also referred as ordinal) and one dichotomous(also referred as binary) variable [23]. The term "biserial" refers to the fact that there are two groups(Y= 0,1) being observed on the continuous variable(X). This coefficient, denoted as r_{rb}, is used to measures degree of relationship between a dichotomous feature(1,0) and the rank of a webpage from which the feature was extracted. The formula for rank biserial correlation is given below:

$$r_{rb} = \frac{2(Y_1 - Y_0)}{n} \tag{5.2}$$

r_{rb} = Rank Biserial correlation coefficient
Y_1 = the Y score mean for data pairs with an X score of 1
Y_0 = the Y score mean for data pairs with an X score of 0
n = the total number of data pairs

Assume we want to calculate the correlation between a page's rank(SET A holds rank instead of position) on Google.nl and the presence of link to the page's Facebook fan page on the site(SET B) using rank biserial correlation. The first step is compute the natural rank of each element in SET A to produce RAKE A, then collect values of RANK A into to SET Y_1 if they have the link or to SET Y_0 otherwise. After that, calculate the mean of the two sets, SET Y_1 and SET Y_0, to produce Y_1 and Y_0 respectively, subtract the second from the first, divide the result by the total number of elements (size of SET Y_1 plus size of SET Y_0) to get the coefficient r_{rb}.

TABLE 5.2: Example of calculating Rank Biserial correlation coefficient on sample data.

SET A	**RANK A**	**SET B**	**SET Y_1**	**SET Y_0**
1	5	0		5
2	4	1	4	
4	3	1	3	
5	2	1	2	
7	1	0		1

$$Y_1 = \frac{4+3+2}{3} = 3$$
$$Y_0 = \frac{5+1}{2} = 3$$
$$r_{rb} = \frac{2(3-3)}{5} = 0$$

The biserial correlation coefficient of this example is 0.0, which is interpreted as there is no correlation between these two variables.

Algorithm

As one can see, algorithm 2 is straight forward implementation of equation 5.2, hence the example provide for it can well summarize it. In this algorithm, when the values are all 0, Y_1 will be 0/0 which gives undefined result, and the formula doesn't give direction what to do in such cases. Hence, we decided to check the total numbers of zeros in the second array against a constant $MAX\_ZERO\_TOLERANCE$ in addition to the other two constraints discussed in the previous algorithm.

Algorithm 2 Rank Biserial correlation algorithm

```
1: procedure RankBiserial(xArray[], yArray[]) ▷ xArray holds rank, yArray holds feature value
2:     n_x ← xArray.length
3:     n_y ← yArray.length
4:     if n_x ≠ n_y then
5:         throw error                                  ▷ dimension mismatch exception
6:     else if n_x < MIN_PAGES ‖ n_x > MAX_PAGES then
7:         throw error                                  ▷ insufficient dimension exception
8:     else
9:         j ← 0
10:        k ← 0
11:        Y_0 ← []
12:        Y_1 ← []
13:        for i ← 0, n_x − 1 do                        ▷ split the ranks based on their factor values
14:            if yArray[i] = 0 then
15:                Y_0[j] ← xArray[i]
16:                j ← j + 1
17:            else
18:                Y_1[k] ← xArray[i]
19:                k ← k + 1
20:            end if
21:        end for
22:        if Y_0.length < MAX_ZERO_TOLERANCE then
23:            throw error                              ▷ # 0 exceed maximum tolerance
24:        end if
25:        meanY_0 ← 0.0
26:        meanY_0 ← 0.0
27:        meanY_0 ← Mean(Y_0[])                        ▷ calculate mean of the ranks
28:        meanY_1 ← Mean(Y_1[])
29:        rbc ← 2 ∗ (meanY_1 − meanY_0)/n_x
30:        return rbc                                   ▷ rbc is the rank biserial correlation coefficient
31:    end if
32: end procedure
```

5.1.3 Generate Rank From Position

As mentioned in the Spearman correlation example, rank is computed by sorting the position of the webpages per query in descending order and taking the index of each in the array (starting from 1), after NaN and tie values are handled, using algorithm 3.

Algorithm

Algorithm 3 Generate Rank from Position

```
1: procedure GenerateRank(xArray[])                ▷ xArray holds position
2:     n_x ← xArray.length
3:     j ← 0
4:     Rank ← []
5:     for i ← 0, n_x − 1 do
6:         Rank[j] ← n_x + 1 − xArray[i]
7:     end for
8:     return Rank[]                    ▷ Rank[] holds the new generated rank
9: end procedure
```

5.1.4 Mean Correlation

To keep the correlation coefficient independent of the search term, we computed an average correlation value of each factor over all search terms. While computing the mean, search term that happen to yield NaN correlation value for a feature are removed first. Algorithm 4 summarizes the mean computation.

Algorithm

Algorithm 4 Calculate Mean Correlation

```
1: procedure MeanCorrelation(xArray[])  ▷ xArray holds correlation coefficients
   of a feature for each search term
2:     n_x ← xArray.length
3:     for i ← 0, n_x − 1 do
4:         if xArray[i] = NaN then
5:             remove(xArray[i])
6:         end if
7:     end for
8:     return Mean(xArray[])
9: end procedure
```

5.1.5 Natural Ranking Algorithm

Natural ranking algorithm is used for ranking based on the natural ordering on doubles [2]. In this algorithm NaNs are treated according to the configured NaN Strategy and ties are handled using the selected Ties Strategy. The following example elaborates with sample input data.

Input data: (20, 17, 30, 42.3, 17, 50, Double.NaN, Double.NEGATIVE_INFINITY, 17)

TABLE 5.3: Example of handling NaN and tie occurrences on input data.

NaNStrategy	**TiesStrategy**	**rank(data)**
default (NaNs maximal)	default (ties averaged)	(5, 3, 6, 7, 3, 8, 9, 1, 3)
default (NaNs maximal)	MINIMUM	(5, 2, 6, 7, 2, 8, 9, 1, 2)
MINIMAL	default (ties averaged)	(6, 4, 7, 8, 4, 9, 1.5, 1.5, 4)
REMOVED	SEQUENTIAL	(5, 2, 6, 7, 3, 8, 1, 4)
MINIMAL	MAXIMUM	(6, 5, 7, 8, 5, 9, 2, 2, 5)

[2] `http://commons.apache.org/proper/commons-math/apidocs/org/apache/commons/math3/stat/ranking/NaturalRanking.html`(July 14, 2014)

5.2 System Design

From high level view, the system developed while conducting this research contains four major components. These are : webpage downloader, ranking factor extractor, correlation calculators and the database. The correlation calculators are basically the implementation of the algorithms discussed above, so we will not talk about it here. The rest three are briefly discussed in this section.

5.2.1 Webpage Downloader

For loading and downloading the pages we had three different tools to choose from. These tools are briefly introduced below.

- PhantomJS : PhantomJS is a headless WebKit scriptable with a JavaScript API. It has fast and native support for various web standards: DOM handling, CSS selector, JSON, Canvas, and SVG[3]. Because PhantomJS can load and manipulate a webpage, it is perfect to use it for downloading.

  A webpage can be loaded, analyzed, and rendered by creating a webpage object.

  The following script demonstrates the simplest use of page object. It loads example.com and then saves it as an image, example.png.

```
var page = require('webpage').create();
page.open('http://example.com', function() {
  page.render('example.png');
  phantom.exit();
});
```

- Wget : GNU Wget is a free utility for non-interactive download of files from the Web. It supports HTTP, HTTPS, and FTP protocols, as well as retrieval through HTTP proxies[4].

  By default, Wget is very simple to invoke. The basic syntax is:

```
$ wget [option]... [URL]...
```

  Wget will simply download all the URLs specified on the command line without interacting, however it does not execute JavaScript scripts on a webpage, and some pages use JavaScript to load their content. For this reason the pages that were

[3] http://phantomjs.org/
[4] http://www.gnu.org/software/wget/manual/wget.html#Overview

downloaded with Wget didn't contain all the content the actual page would display when opened with browser.

- Selenium WebDriver : Is a tool for automating web application testing, and in particular to verify that they work as expected[5]. Selenium-WebDriver provides a handy unified interface that works with a large number of browsers, it works by making direct calls to the browser using each browser's native support for automation. How these direct calls are made, and the features they support depends on the browser being used.

  Selenium-WebDriver allows you to write tests in almost every language you can imagine (Java in our case). The easiest way to set up a Selenium 2.0 Java project is to use Maven. Maven will download the Java bindings (the Selenium 2.0 Java client library) and all its dependencies, and will create the project, using a maven pom.xml (project configuration) file.

  As mentioned earlier Selenium WebDriver can work with many different browsers, such as Firefox and Internet Explorer. For Firefox, the WebDriver named FirefoxDriver, is implemented as an extension. The content of the page might not be the same if JavaScript is disable when it loads, so it is very important to execute JavaScript. Since the Firefox driver support Javascript, we choose Selenium WebDriver along with FirefoxDriver over the two other options. On top of that, we find it very easy to set many different preferences using FirefoxDriver. The snippet code below shows how FirefoxDriver was used to load page source of a webpage using URL.

```
...
WebDriver driver = new FirefoxDriver();
driver.manage().timeouts().pageLoadTimeout(180, TimeUnit.SECONDS);
driver.get("http://www.utwente.nl");
String pageSource = driver.getPageSource();
...
```

The webpage downloader is written in Java with Selenium WebDriver. It reads URL of a webpage from database, load the page with FirefoxDriver, and store it in local disk with certain file format. The diagram below [Figure 5.1] depicts the webpage downloading process.

[5] http://docs.seleniumhq.org/docs/03_webdriver.jsp#introducing-webdriver

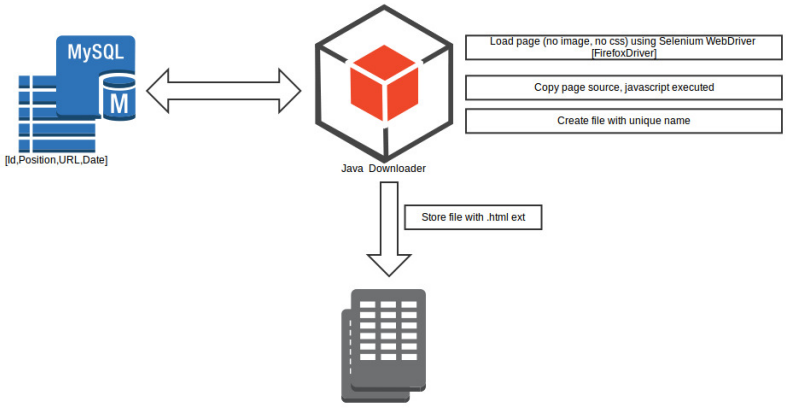


FIGURE 5.1: Webpage Downloader

The pages are stored with .html extension in following file name format.

[PageId]-[SearchTerm]-[UniqueNumber].html

To speed up the download process it was necessary to create a profile for the FirefoxDriver and set prefrences. Typing (*about* : *config*) on Firefox browser's address (or URL) bar displays all possible Firefox preferences, with their status, type and value. Out of these preferences, options that ignore parts of the page such as CSS, images while loading are set to make the process faster. Regardless, some exceptional pages are downloaded with their image and CSS because the images and CSS had full path on the page's code. In addition preference such as browser on disk caching, browser on memory caching, browser offline caching and network caching are also disabled to limit the browsers memory usage. We also set a preference to limit the wait time and just skip pages if they're taking too long to load. For example, I navigate to Google and it takes like 5 seconds. But then I navigate to some random site 30 seconds passes and the server still hasn't responded. To solve such exception, the program catch the timeout exception and quit the FirefoxDriver and restart it. The maximum load time is set to 180 seconds, if a page can't be loaded in this time the driver will throw exception. We keep a log of such pages to try them later.

Another important setting in the webpage downloader is limiting the memory consumed on FirefoxDriver by plugin-container. A plugin is a piece of software that displays Internet content that Firefox is not designed to display[6]. Some common plugins we encountered in the pages are Adobe Flash, Quicktime, and Silverlight. Each plugin are loaded separately from Firefox in a plugin-container process, allowing the main Firefox process to stay open if a plugin crashes. There are as many plugin-container processes

[6] `https://support.mozilla.org/en-US/kb/What%20is%20plugin-container`

as plugins launched since the Firefox session startup. If not limited this can create a memory over usage by the browser instance, therefore we set the maximum number of plugin that can be cached in the plugin-container to 2, which is 10 by default.

FirefoxDriver launches the browser and load the page and then download it. Even though we programmatically tried to avoid downloading pdf and doc files, we still get "Save File" pop-up. The reason for this is, our program identifies such files by the extension of the URL (e.g. `http://www.example.com/file.pdf`) but some URLs are a direct link to a PDF or a Doc file and launches the "Save As" pop-up. Other pop-ups like "Print", "Send Email" were also encountered but these pop-ups didn't affect the download process. We found it very handy to use Selinium WebDriver (FirefoxDriver) for loading and downloading the webpages because we were able to see the page getting loaded on the browser and notice if something is wrong with page while loading. For instance we encountered some "404 page not found" errors, and some unresponsive pages which took longer to be loaded.

The downloader (two instances) was downloading 55 webpages per minute on average (including the time needed to write to file and to database). Based on this calculation, it download 79200 pages per day (24 hours).

5.2.2 Ranking Factors Extractor

We write Java code to extract each factor from the webpages. As explained in Chapter 4 we used JSoup Java library for extracting and manipulating data from the webpages. JSoup parses the HTML of each page into DOM tree, from that we extract all the tags we need, then analyze attributes of the tags if required. For the URL related factors our Java-based extractor rely on the Guava[7] project which contains several of Google's core libraries. Particularly, we took advantage of a useful tool called InternetDomainName from Guava library, for parsing and manipulating domain names. The example below shows, how to fetch the Wikipedia homepage, parse it to a DOM, and select the headlines from the news section into a list of Elements using JSoup.

```
Document doc = Jsoup.connect("http://en.wikipedia.org/").get();
Elements newsHeadlines = doc.select("#mp-itn b a");
```

5.2.3 Database

The system makes use of database tables to store all the data. Several tables are used to store intermediate and final result of the feature extraction process. Our factor extractor

[7] `https://code.google.com/p/guava-libraries/`

uses the Spring framework which takes care of all the low-level details of JDBC to access the database tables. Description of the table used to store the final result is provided in Appendix C.

5.3 Technical Challenges

We have tried many things to the get the number of +1s of a Google plus page/account but could not find a working API. As a workaround, we tried to fetch it directly from a page that have +1 button. To avoid blocking by Google+ we used proxies. Unfortunately this all did not help, because the process consumed longer time than expected. Similarly, on our trial to collect Facebook signals of fan pages using their graph API, we repeatedly get error message "Application request limit reached" for making burst calls. Because of this basic technical difficulty we restrained our factor extractor to work on the on-page factors only.

5.4 Summary

The position of webpages in our input data is converted to rank by sorting the position of the webpages per query in descending order and taking the index of each in the array (starting from 1). Spearman rank correlation and rank biserial correlation coefficient equations do not tell what to do in some special cases, therefore we have introduced some additional constraints such as maximum number of zeros, minimum/maximum number of webpages etc. in our algorithms. The algorithms performs a rank transformation on the input data, by handling NaN and tie occurrences based on the required strategies. Selinium WebDriver (FirefoxDriver) is used for loading and downloading the webpages, and each page is parsed to DOM tree using JSoup Java library.

Chapter 6

Results

This chapter discusses the correlation results and explains which factors are positively correlated, not correlated (with no linear correlation), and which are negatively correlated on both the DUTCH WEB and LETOR4.0 datasets. In addition, this chapter provides an answer to the research question RQ-2: *"Is it better to use only the top well ranked pages (e.g top 40) while computing correlation coefficients instead of using all ranked pages per search term?"*.

6.1 Introduction

Correlation results for the DUTCH WEB dataset is presented and discussed first, subsequently correlation results for the LETOR dataset are discussed. Prior to that basic statistics about the DUTCH WEB dataset used to calculate the results and draw the conclusions which are reported in this chapter is given on table 6.1. Similar information about the LETOR dataset is given in Appendix A.

TABLE 6.1: Basic statistics on the final dataset used in this research

INFO	VALUE
Total keywords (search engine queries) used	7568
Total Google results pulled per keyword	40
Total factors (variables) analyzed per result	52
Total data points	15741440

6.2 Correlation Results : DUTCH WEB Dataset

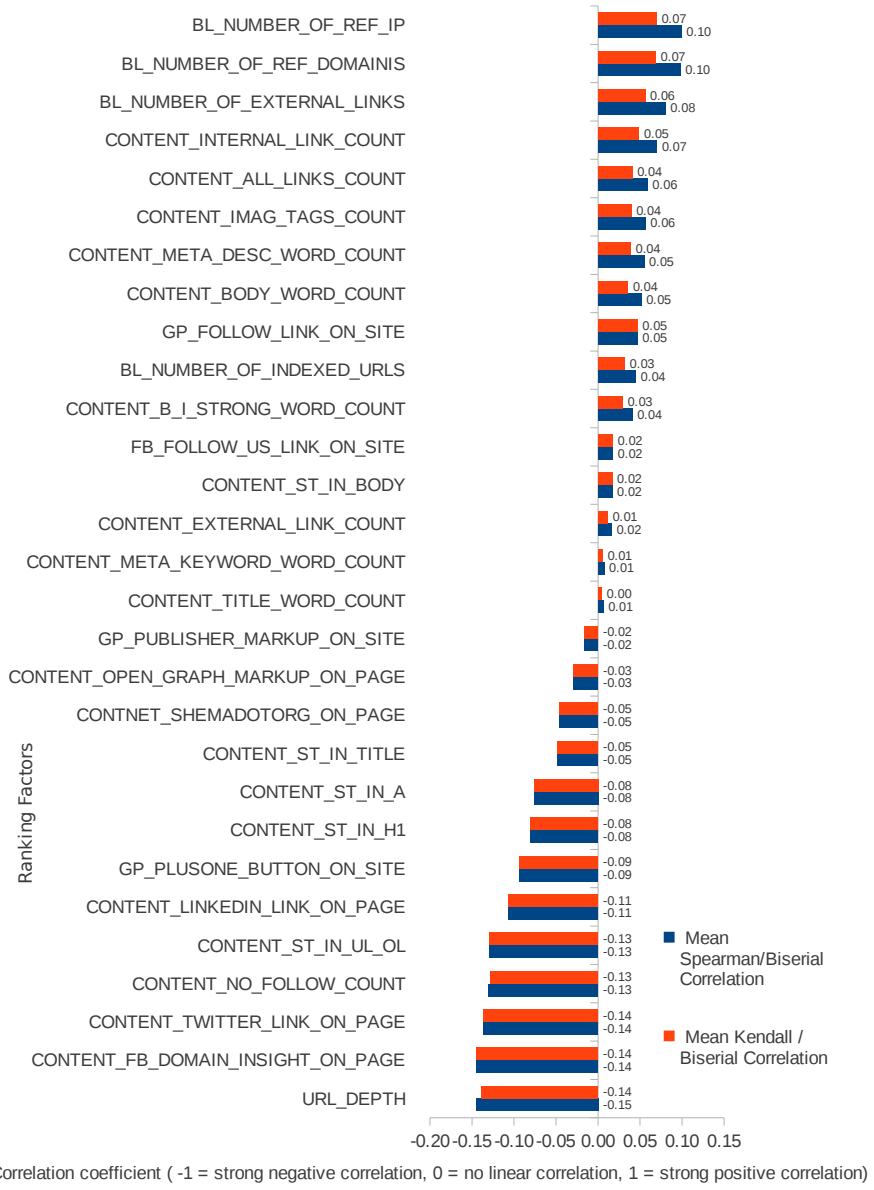
The graph on figure 6.1 shows correlation of 29 factors extracted from ranked webpages and their rank (calculated from position of each page on Google.nl). X-axis shows correlation coefficients of Spearman and Kendall (for continuous factors) or Biserial (for dichotomous factors). Y-axis shows the factor's name in a self explanatory way. The coefficients range from 1 to -1, 1 indicates a strong positive correlation where as -1 shows a strong negative correlation. When the coefficient lie at 0, it means there is not any linear relationship between the two variables (see Section 5.1). We compute both Spearman and Kendall rank correlation coefficients for continuous factors for comparison purpose. *As expected the result shows that Kendall tau is smaller than Spearman rho in all cases, which gives us a confidence on the validity of the calculation.*

We have collected different results of correlation by varying argument values on the correlation algorithms. One of the observation is that when Total Number of Webpages Per Search Term is decreased (e.g. consider only top 10 pages) the correlation gets weaker, which indicates *using larger dataset to compute correlation gives stronger result.* Another one is, when the Maximum Zero Tolerance is decreased (e.g. to 20) the correlation gets stronger. Although we have many results, here we discuss only one result which is computed based on following configuration.

- NaN Strategy = FIXED (when NaN is encountered in the dataset, the algorithm takes the position of the NaN element in the array).
- Tie strategy = MINIMAL (when there is a tie, two or more elements have same assigned rank, then the algorithm takes the minimum possible rank)
- Maximum Zero Tolerance = 40 (a factor can have up to 40 zeros in it's value per search term)
- Minimum Ranked Pages Per Search Term = 30 (if a search term have less than 30 ranked pages then no correlation is calculated for it)
- Maximum Number of Ranked Pages Per Search Term = 40
- Total Number of Unique Search Terms = 7568

An example to elaborate these strategies and constraints is provided in Section 5.1.5.

FIGURE 6.1: Mean of Spearman-Biserial and Mean of Kendall-Biserial rank correlation coefficients of ranking factors (see Section 4.2) computed on Google.nl, 2014 (DUTCH WEB dataset).



As shown in the graph[1] on Figure 6.1, backlink related factors, such as total number of unique IP referring to domain of a webpage, total number of unique domains referring to domain, total number of indexed pages per domain, and total number of backlinks come out on the top of the chart with relatively stronger positive correlation. Some social signals are also our highest correlated factors, with Google+ (i.e. having link to Google+ page on site) edging out Facebook (i.e. having link to Facebook fan page on site), even though LinkedIn and Twitter turn out to be negatively correlated. Similarly having more internal links, as well as links that point to other websites shows a positive correlation to well ranking. Not only does the number of URLs (links) in page affects ranking, but also the length of the URL (URL depth) itself seems to have negative correlation with well ranking, and we advise Webmasters to regard it as bad practice of URL structuring. Out of the 16 positively correlated (weakly correlated) factors 13 are continuous and 3 are dichotomous variables.

The feature "word count in text" is correlated positively with well ranked pages on Google.nl. Pages that contain more words in the description meta tag, image tag, bold, strong and italic as well as body tag seems to have a better correlation. Although not strong it was surprising to see positive correlation between number of words in the keyword meta tag and Google's ranking, which basically indicates the more keywords you put in this tag the higher you rank.

Looking at the chart shows a marginally negative correlation for the presence of search term in title, therefore there is no correlation (if not weak correlation) for this factor. Similarly existence of keyword in anchor tag, ordered list, unordered list and headings show weak negative correlation to well ranking. Although not included in this graph, the EMD (i.e exact match of search term to URL domain) and PMD (i.e. partial match of search term to URL domain) resulted with significantly low correlation value (negative correlation), this leads as to conclude that Google seems to be much better at distinguishing irrelevant ranked EMD and PMD pages. Counter to our expectation the presence of different markup tags such as publisher markup, Schema.org markup, author markup and open graph markup seems to have no (if not negative) influence on well ranking.

To summarize, on a larger scale, the statistics (see Appendix C) and correlation results on the DUTCH WEB dataset do not show strange looking results compared to similar studies reviewed in Chapter 3. In the statistics, even though it is almost obvious, it was interesting to see very large majority of the ranked webpages are with .nl TLD. The statistics also shows there is still much to do regarding to SSL implementation on the

[1]The reader is advised to refer to the following abbreviations: BL denotes Backlink, FB denotes Facebook, GP denotes Google Plus, ST denotes Search term.

Dutch web, since only 3% of URLs use HTTPS. Our correlation result shows building huge number of backlinks towards a webpage is immensely important for achieving top ranking on Google.nl, thus it should get a primary focus. The majority of websites ranked among the top 40 tends, to a certain extent, to have more text, therefore putting enough and well contented text on body, description, and image tags of a website is still part of the SEO basics. As expected, Google.nl does not seem to give any emphasis to EMD and PMD websites, although most of the EMDs and PMDs are more abundant on top 10 set compare to top 40 set. In addition, linking from a site to it's Google+ and Facebook fan pages shows positive correlation to well ranking on Google.nl.

Note : It must be noted that similar results from other SEO companies such as MOZ cannot be compared on a perfectly common basis to the results presented here, because the data gathering was executed at different times.

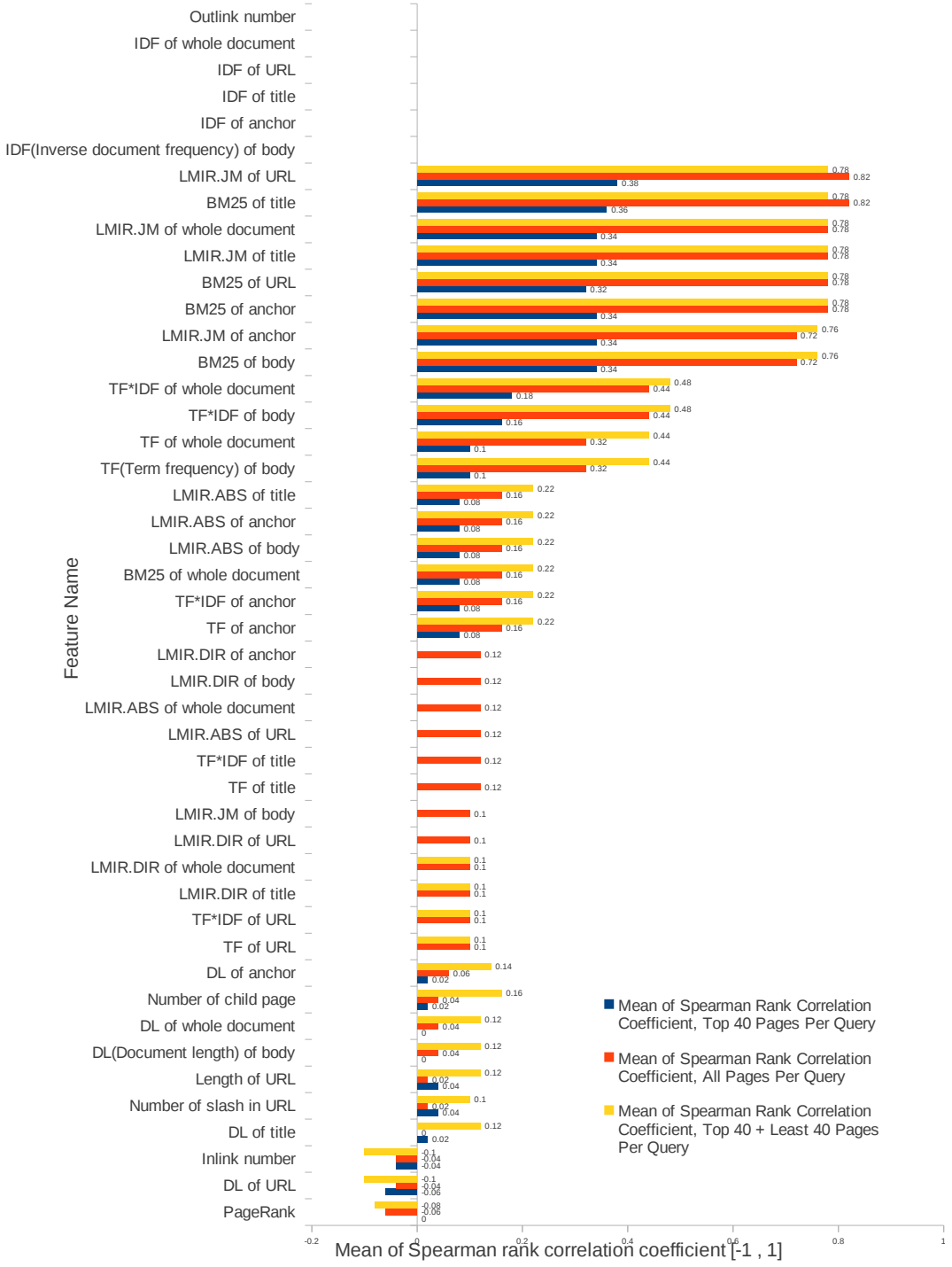
6.3 Correlation Results : LETOR4.0 Dataset

Here we present the results obtained in an effort to answer the question RQ-2: *"Is it better to use only the top well ranked pages (e.g top 40) while computing correlation coefficients instead of using all ranked pages per search term?"*. We suspect that considering only the top 40 or top 100 like most other similar studies, might introduce some bias because it ignores the low ranked pages. Gathering all ranked webpages for a particular search term from search engines is often impractical and challenging. As a common practice in Information Retrieval, given a query, only some "possibly" relevant documents are selected for judgment[20]. The LETOR4.0 dataset contains relatively large number of "possibly" relevant documents per search term[2], which gives us a chance to experiment by varying the amount of documents in our input data while computing correlation coefficients.

We first computed Spearman rank correlation coefficient of each feature (46 features) and the rank of *all webpages* (i.e. ground-truth) on the MQ2008-list dataset (computation of the features is discussed in [33][20]). In order to keep the correlation independent from each query, we computed the average (mean) over all queries, for each feature. The next step was to compute the mean Spearman rank correlation coefficient same way but this time by analyzing *only the top 40 ranked webpages*. Lastly we repeat same procedure by analyzing a dataset that contain a *combination of top 40 and least 40 ranked webpages.*

[2]To construct the LETOR dataset, they used the BM25 model to rank all the documents with respect to each query, then selected the top 1000 documents for each query for feature extraction.

FIGURE 6.2: Mean of Spearman rank correlation coefficient of each feature computed for LETOR4.0-MQ2008-list dataset using **top 40**, combination of **top 40 + least 40**, and **all** ranked pages.



The graph on figure 6.2 illustrates, overall the *correlation gets weaker when using top 40 instead of all ranked pages. For most features a relatively stronger correlation is obtained with set that contains the top 40 and least 40 ranked pages.* Also, it is clear by looking at the graph that, the correlation decreases from top to bottom with slight inconsistency for all cases. Except for 3 features which shows weak negative correlation and the features with NaN values, the rest are positively correlated to their ground-truth. Further more it was surprising to see absolute consistency in the sign of the coefficient (negative or positive) for all cases : a feature's correlation sign remained same for all three cases (e.g. PageRank shows negative correlation in all cases).

At the top of the chart, 5 features which are related to IDF (Inverse Document Frequency), yield mean correlation coefficient equal to NaN. Since these features are query dependent (or document independent), they are the same for all ranked pages under a query with normalized value of 0.0. Another feature called "Outlink number" which happed to have a normalized value of 0.0 for all queries also give NaN. In the middle of the chart, 8 features show result (not NaN) only when all pages are considered. For the other two cases, these futures give a mean correlation coefficient equal to NaN, because these feature are not found on the top 40 and least 40 ranked pages. The graph also seems to indicate that certain features related to two weighting schemes : LMIR[3] calculated according to [34] and BM25[4] calculated according to [20] have the highest correlation with well ranking in all cases. Factors related BM25 are positively strongly correlated, probably because the relevant documents ($\leq$ 1000) are selected by ranking all documents with BM25 model when making this benchmark dataset. What is even more interesting is, to see the Pagerank and number of Inlinks, factors which are widely believed as important ranking factors, are weakly negatively correlated with their ground-truth in this result. In contrast to a finding in our analysis with the DUTCH WEB, we see a positive correlation between the depth of a URL (i.e. Number of slashes in URL) and well ranking on the LETOR4.0 dataset.

To sum up, we can evidently concluded that, the strongest correlation coefficient between a feature value and the rank of relevant webpages per query, is obtained when a dataset that contain a combination of top few and least few ranked pages is utilized. That being said, the number of features which yield NaN correlation coefficient decreases when larger amount (if not all) ranked pages are considered.

[3]LMIR stands for Language Model for Information Retrieval, JM stands for Jelinek-Mercer, DIR stands for Dirichlet prior and ABS stands for absolute discounting

[4]BM25 weighting scheme (BM stands Best Match) is used to measure term weights, recent TREC tests have shown BM25 to be the best of the known probabilistic weighting schemes.(`http://xapian.org/docs/bm25.html`, July 29, 2014)

6.4 Summary

With the DUTCH WEB dataset, we have showed that backlink related factors yield relatively stronger positive correlation. Similarly links pointing to social media fan pages, internal links, and links that point to other websites shows a positive correlation to well ranking. The length of the URL seems to have negative correlation with well ranking, and should be regarded as bad practice of URL structuring. With the LETOR dataset, we have mainly showed that, relatively stronger correlation coefficient between features and ranking is achieved when a dataset that contain webpages from the upper and lower part of a search result is used as input data.

Chapter 7

Evaluation

In this chapter we provide the evaluation measures used, our proposed strategies and results found while answering two core question of this research: SRQ-3.1: *"Is there any sensible relationship between the calculated correlation coefficient of a ranking factor (first approach) and it's corresponding weight assigned by a ranker (second approach)?"* and SRQ-3.2: *"Does considering highly correlated ranking factors give a better performing ranker, compared to using the whole set of ranking factors?"*

7.1 Rank-Based Evaluation Measures

The evaluation measures used in our experiment expects system's (trained ranker) output in the form of ranked documents derived by scoring function[35]. We used NDCG@K (NDCG@10) to optimize the training data for both the LETOR4.0 and DUTCH WEB datasets. The metric used to evaluate on the test data is ERR@K (ERR@10). Both this two metrics as well as other rank based measurements are briefly explained next:

- **Normalized Discounted Cumulative Gain (NDCG) :** Valizadegan et al. [9] defines NDCG as follow, suppose that we have a list of n queries for training, denoted by $Q = \{q^1, ..., q^n\}$. For each query q^k, we have a list of m^k documents $D^k = \{d_i^k, i = 1, ..., m_k\}$, whose relevance to q^k is given by a vector $r^k = (r_1^k, ..., r_{m_k}^k) \in Z^{m_k}$. The ranking function denoted by $F(d, q)$ takes a document-query pair (d, q) and outputs a real number score. j_i^k denotes the rank of document d_i^k within the collection D^k for query q^k. The NDCG value for ranking function $F(d, q)$ is then computed as follow:

$$L(Q,F) = \frac{1}{n}\sum_{k=1}^{n}\frac{1}{Z_k}\sum_{i=1}^{m_k}\frac{2^{r_i^k}}{\log 1 + j_i^k} \tag{7.1}$$

Where Z_k is the normalization factor. NDCG at position k for query q is computed as follow (explanatory example is provided at APPENDIX B.1):

$$NDCG@k = Z_k\sum_{j=1}^{k}\frac{(2^{c(j)})}{\log 1 + j} \tag{7.2}$$

- **Expected Reciprocal Rank (ERR) :** This metric is defined as the expected reciprocal length of time that the user will take to find a relevant document. The metric supports graded relevance judgments and assumes a cascade browsing model. In this way, the metric quantifies the usefulness of a document at rank i conditioned on the degree of relevance of the items at ranks less than i[36].

$$ERR = \sum_{r=1}^{n}\frac{1}{r}P \tag{7.3}$$

Where n is the number of documents in the ranking. P is the probability of the user stopping at position r.

- **Winner Take All (WTA) :** For query q, if top ranked document is relevant: $WTA(q) = 1$; otherwise $WTA(q) = 0$. It do not care about other documents, averaged over all queries[1].

- **Mean Reciprocal Rank (MRR) :** For query q, rank position of the first relevant document is denoted as $R(q)$, documents ranked below $R(q)$ are not considered. $\frac{1}{R(q)}$ is used as the measure for query q and averaged over all queries[2].

- **Mean Average Precision (MAP) :** It is precision at position n for query q[3]

$$P@n = \frac{\#relevant : documents : in : top : n : results}{n} \tag{7.4}$$

Average precision for query q:

$$AP = \frac{\Sigma P@n * I(document : n : in : relevant)}{\#relevant : documents} \tag{7.5}$$

[1] `http://en.wikipedia.org/wiki/Winner-take-all`
[2] `http://web.stanford.edu/class/cs276/handouts/EvaluationNew.ppt`
[3] `https://www.kaggle.com/wiki/MeanAveragePrecision`

7.2 Evaluation Strategies

Here, we present our two proposed evaluation strategies, the steps followed to execute them and the results obtained.

Strategy-1 : Relationship Between Correlation and Weight

The objective of this evaluation strategy is to show if there is sensible relationship between the *correlation coefficients of features to their ground-truth/Google rank* **AND** corresponding *weight assigned to features by ranking function*. Formally, we intend to answer SRQ-3.1: *"Is there any sensible relationship between the calculated correlation coefficient of a ranking factor (first approach) and it's corresponding weight assigned by a ranker (second approach)?"*. We hypothesize that there is a linear relationship between the correlation coefficients and weight of a certain factor. In other words, we wish to see that highly correlated features also have higher assigned weight by trained ranker.

The strategy is executed based on the following basic approach : Given a set of webpages $P = (p_1, p_2, ..., p_n)$, factors extracted from each webpage i $F_i = (f_1, f_2, ..., f_n)$, pre-calculated Google ranking $G = (1, 2, ..., n)$ for the DUTCH WEB dataset (or pre-calculated ground-truth for the LETOR dataset) and a listwise ranking algorithm A, the goal is to find a set of weights $W = (w_1, w_2, ..., w_m)$ that are assigned by the ranking algorithm to ranking factors, and compare it to the set of pre-calculated Spearman rank correlation coefficients of ranking factors $S = (s_1, s_2..., s_n)$.

A set of correlation coefficients S is calculated and discussed in chapter 6, hence below we only discuss the steps we pass through while training a ranking function on the DUTCH WEB dataset and the LETOR4.0 dataset particularly the MQ2008-list version[4] to get a set W of weights assigned for each ranking factor.

First, the ranking algorithm we worked on uses listwise approach of learning to rank, therefore we had to make our dataset suitable by finding a way to add ground-truth to the query-document pairs. In a similar situation, Cao et al. [16] on their experiment of listwise methods (i.e. ListNet) with different datasets such as TREC[5] and OHSUMED[6], simply used the 'rank' of related documents to obtain the ground truth (i.e., the ranking list of true scores) for each query. In the same way, since we do not have relevance judgment for DUTCH WEB dataset, we made an assumption that, the ranked list of webpages fetched from Google.nl are ordered according to their relevance value, with

[4]The MQ2008-list dataset can be fetched from this link : `http://research.microsoft.com/en-us/um/beijing/projects/letor/LETOR4.0/Data/MQ2008-list.rar`(June 10, 2014)

[5] A dataset obtained from web track of TREC 2003

[6] A benchmark dataset for document retrieval

most relevant page on the top. *Therefore we calculated a set G that contains rank of every ranked webpages per search term generated from their position on Google.nl using algorithm 3 and consider it as our ground-truth.*

Second, it was necessary to choose an algorithm A such that it is easier to extract the weight of each factor from the trained model after training is completed. We first experiment with LambdaMART which is the boosted tree version of LambdaRank. This method has proven to be successful algorithm for solving ranking problems, by woning Track 1 of the 2010 Yahoo! Learning to Rank Challenge [37]. It worked perfect with the DUTCH WEB as well as the LETOR4.0 datasets, but the trained model is build in tree format, which made it hard to select single weight assigned for each feature. Later, we discovered that Coordinate Ascent gives an optimized single weight for each feature which fits to our goal. The relative value of each feature implies to the relative importance of the features. Coordinate ascent has been used for the purpose of feature selection for its characteristics of optimizing multivariate objective function by sequentially doing optimization in one dimension at a time. It cycles through each parameter and optimizes over it while fixing all the others. Dang and Croft [8] proposed feature selection that uses Coordinate Ascent to combine subset feature into single feature. Similarly, Metzler and Croft [38] have proposed a listwise linear model for information retrieval which uses Coordinate Ascent to optimize the model's parameter.

Third, the RankLib[7] package requires maximum number of ground-truth (which is 4 by default) to be inputed as argument in order compute the ERR@10 metric. However we could not find any information regarding this, so we wrote small script to extract the maximum number of ground-truth used in the dataset which was : 1831. The maximum ground-truth value per query-document pairs varies from query id to query id.

Fourth, we trained five models using five-fold cross validation on the MQ2008-list dataset. Each fold contains the whole dataset divided into 60% (471 queries : 540679 documents) training subset, 20% (157 queries : 180664 documents) validation subset and 20% (156 queries : 180877 documents) test subset. After training five different models using the training subsets, we select the model that yielded the highest NDCG@10 on the validation set. It was then evaluated using the test set and the performance was measured with ERR@10. We set different general and algorithm specific parameters. As mentioned earlier the algorithm used is Coordinate Ascent and the whole feature set is utilized. Zscore is used to normalize the data before using it for training. For computing ERR@10, the largest ground-truth value in listwise approach was set to 1831. We set the number random restarts to 5 to avoid local extrema. The number of iterations to

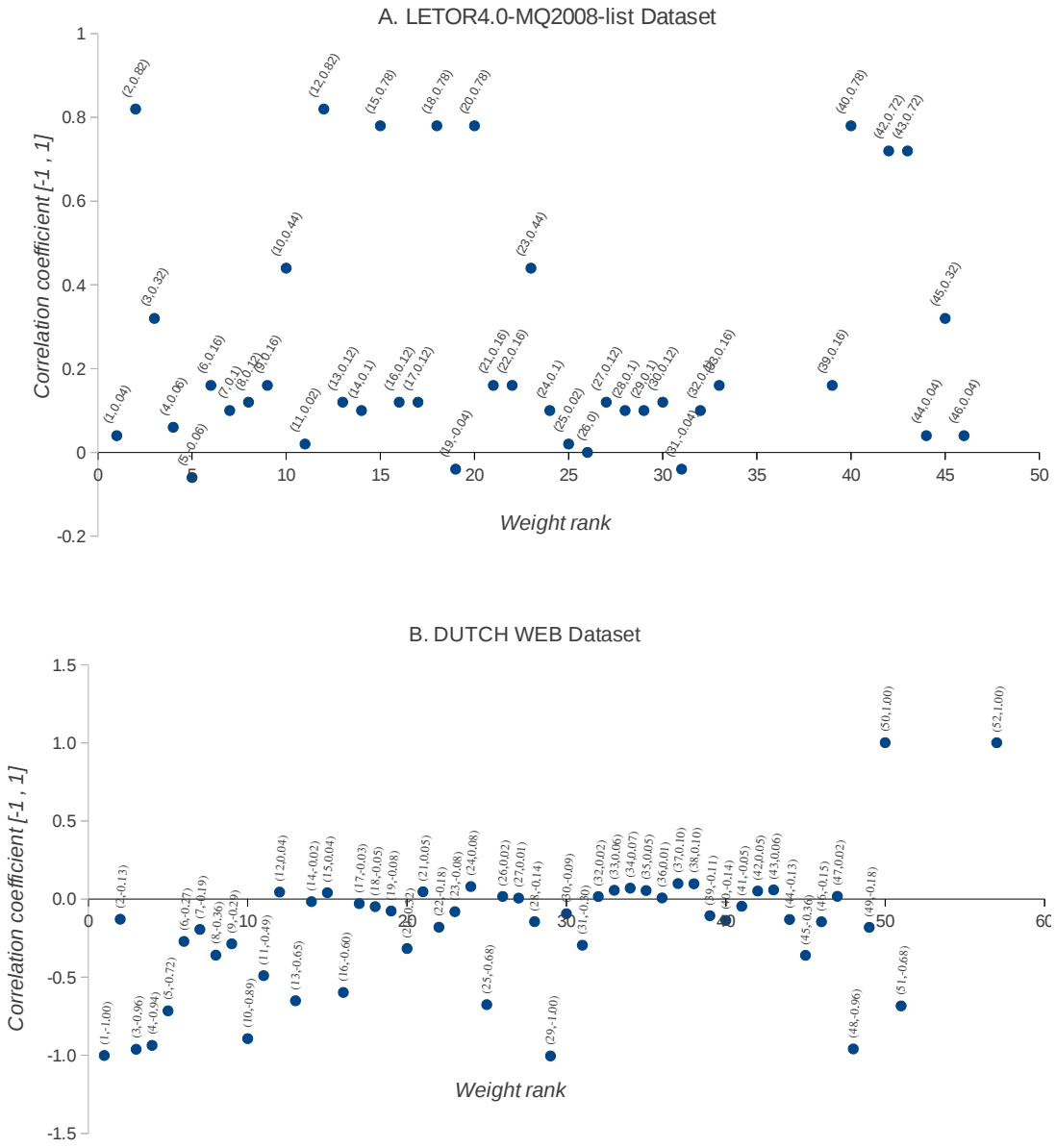
[7] RankLib is a library of learning to rank algorithms. Currently it has implemented 8 popular algorithms, among them 4 are listwise approaches (AdaRank, ListNet, Coordinate Ascent and LambdaMART)

search in each direction was set to 25. We train the ranker until observing a drop on the NDCG@10 between two consecutive iteration that is below a tolerance value of 0.001 (see Appendix B for details).

Fifth, the final step was to plot a graph [Figure 7.1(A)] with the mean Spearman rank correlation coefficients of each feature on Y-axis (i.e. S), against the ranks of weights of the features obtained from the previous step on X-axis (i.e. $rank(W)$, highest weight $= 1^{st}$ rank). A single dot represents the intersection of X-Y pair of a single feature and is labeled by (x,y) values. We plot the rank of a feature's weight instead of the actual value in order to improve the readability of the graph. Since 6 features turn out to have NaN correlation coefficient the graph shows only 40 dots. Full list of feature id, mean Spearman rank correlation coefficient, Coordinate Ascent feature weight is provided on Appendix B.

The steps discussed above are repeated on the DUTCH WEB dataset, and the results are provide on Figure 7.1(B).

FIGURE 7.1: **Weights** of features assigned by Coordinate Ascent sorted in descending order (highest weight assigned 1^{st} rank) **versus** corresponding **mean Spearman rank correlation coefficients** of features, computed for LETOR4.0 - MQ2008-list(A) and DUTCH WEB(B) datasets, each point is **labeled with (x,y)**.



In general it appears that for the LETOR4.0 dataset, the correlation as well as the weight is positive for most features. For the DUTCH WEB on the other hand, the intersection of X-Y pair for most of the features lie under zero line. In addition the graph from the LETOR4.0 dataset analysis shows, only one feature (i.e. LMIR.JM of URL) which have very strong positive correlation is also assigned a weight which ranked 2^{nd} (i.e. the dot labeled (2,0.82)). On the opposite, three features which have very strong positive correlation (LMIR.JM of title = 0.78, LMIR.JM of anchor = 0.72 and BM25 of body = 0.72) are assigned weights that ranked very low (40, 42 and 43 respectively). In the middle, four features with approximately same positive strong correlation coefficient are matched with weights that rank in between 12 and 20. When we look into the DUTCH WEB graph, a feature with a perfect negative correlation (i.e -1) happen to have the highest weight. In contrary to this, two features that showed a perfect positive correlation (i.e. +1) with Google.nl's ranking are assigned the smallest weight in the Coordinate Ascent trained model (see Appendix B for the exact values of weights).

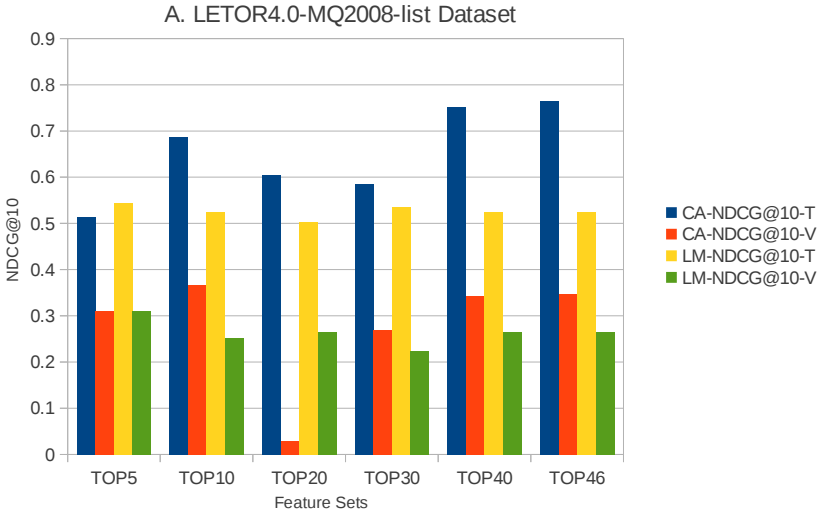
Based on this graphs we do not have enough evidence to draw strong conclusion on the matter. However, one can clearly see that, strong positive and strong negative correlated features does not necessarily get highest and lowest weight respectively on trained ranker particularly Coordinate Ascent. Hence, *since features with high weight are the most contributing factors for well ranking, we can deduce to some degree that strong correlation is not always a cause for well ranking, in other words the result provides a proof for the theory : correlation* $\neq$ *causation*. Further more, we believe a more conclusive result could be obtained by carefully choosing the correlation technique as well as the learning algorithm.

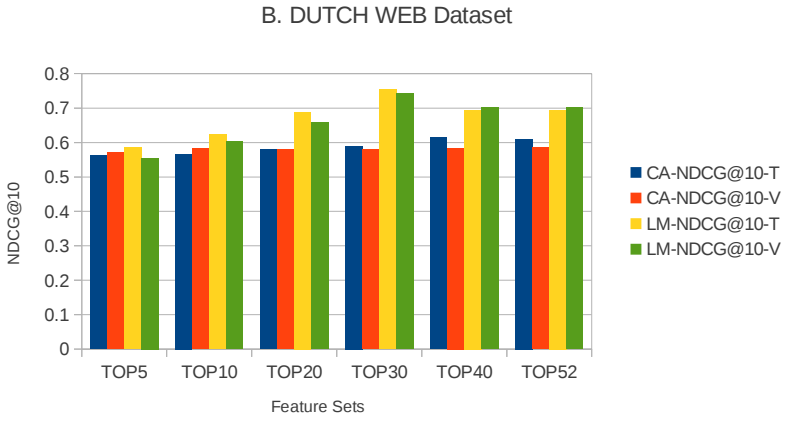
Strategy-2 : Performance of Rankers Trained On Various Subsets of Features

On this part of the evaluation, we provide an answer to SRQ-3.2: *"Does considering highly correlated ranking factors give a better performing ranker, compared to using the whole set of ranking factors?"*.

To answer this question, we train 6 rankers for LambdaMART and Coordinate Ascent each on the LETOR4.0-MQ2008-list dataset. The configuration of the training is conducted by utilizing different set of features. The total feature sets are categorized into 6 subsets after ordering them in descending order according to their mean Spearman rank correlation. Note that the NDCG@10 values are normalized for presentation purpose.

FIGURE 7.2: Features ordered according to their Spearman/Biserial rank correlation coefficient (Descending), divided into 6 sets, used to train a ranking model with LambdaMART (LM) and Coordinate Ascent (CA) on the LETOR4.0-MQ2008-list(A) and DUTCH WEB(B) datasets, the NDCG@10 measurement on the training data (NDCG@10-T) and the validation data (NDCG@10-V) is presented in this two graphs.

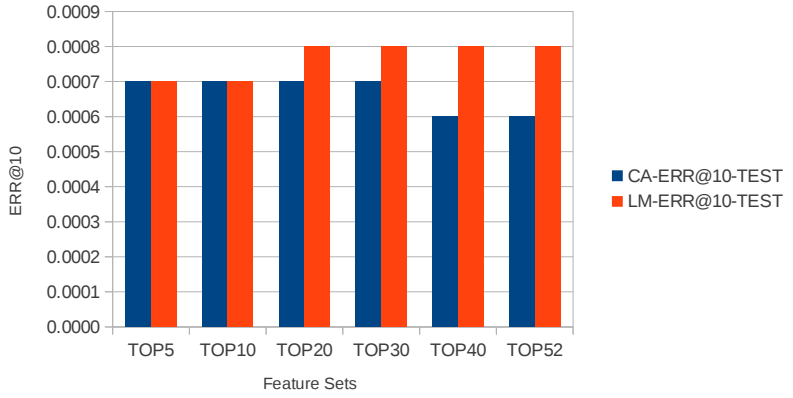




As overall trend, the graph on figure 7.2(A) shows the value NDCG@10 on the validation data (i.e. NDCG@10-V) for both Coordinate Ascent and LambdaMART does not really vary in consistent manner as expected. If we look into NCDG@10 value of Coordinate Ascent (i.e. CA-NDCG@10-V) it first raises from TOP5 to TOP10, then it drops with TOP20, later it starts to increment with TOP30,TOP40 and TOP46 feature sets. Whereas for LambdaMART (i.e. LM-NDCG@10-V) it falls down when the feature set changes from TOP5 to TOP10 and remained approximately same with TOP20 and TOP30 before incrementing on the TOP40 and TOP46 feature sets. The change in the value of NDCG@10 with the different feature sets in LambdaMART is more steady compared to Coordinate Ascent, each perform best when using TOP5 and TOP10 feature sets respectively.

Unlike the results on the LETOR4.0 dataset, the value of NDCG@10 with the DUTCH WEB dataset, as shown on figure 7.2(B), remained roughly constant on different feature sets with Coordinate Ascent. With LambdaMART however, it fist slowly grows from TOP5 to it's peak value at TOP30 and then falls back with TOP40 and TOP52 feature sets.

FIGURE 7.3: Features of the DUTCH WEB dataset ordered according to their Spearman/Biserial rank correlation coefficient (Descending), divided into 6 sets, used to train a ranking model with LambdaMART (LM) and Coordinate Ascent (CA), the ERR@10 measurement on the test data (ERR@10-TEST) is presented in this graph.



The bar graph on Figure 7.3 shows the performance of the models trained on the DUTCH WEB dataset, evaluated on the test data measured with ERR@10. For Coordinate Ascent the ERR@10 value remained same except for the TOP40 and TOP52 feature sets which gives lower value. On the other hand, with LambdaMART it gives same value for TOP5 and TOP10, and a bit higher value for the rest feature sets.

Concluding, pre-filtering feature sets according to their mean Spearman/Biserial correlation coefficient indeed give a better performing ranker when performance is measured with ERR@10 in Coordinate Ascent. Where as for LambdaMART, a better performing ranker is achieved by using the whole set of features. Therefore the answer to the question asked earlier on this section is : it is dependent on the learning to rank algorithm used for training.

7.3 Summary

In this chapter we have discussed evaluation measures as well as suitable learning to rank algorithms that can be utilized to conduct such experiment. Most appealing, we have showed in our result that strong correlation does not always establish well ranking which goes in line to the theory of *correlation* $\neq$ *causation*. Moreover, we have provided a basic proof that demonstrate, pre-filtering feature sets according to their correlation coefficients to improve performance of a ranker is heavily dependent on the learning to rank algorithm applied.

Chapter 8

Conclusion and Future Work

8.1 Conclusions

In this thesis, we presented work related to "Identifying the most influential ranking factors for well ranking on search engines". Broadly, we investigated on the DUTCH WEB and LETOR dataset and presented the correlation results of identified factors and ranking. We focus on two approaches that are used to select set of ranking factors that have higher influence in well ranking. The first approach is, to calculate correlation coefficient (e.g. Spearman rank) between a factor and the rank of it's corresponding webpages (ranked document in general) on a particular search engine. The second approach is, to train a ranking model using machine learning techniques, on datasets and select the features that contributed most for a better performing ranker.

8.1.1 Main Contribution

Using the results obtained from previous chapters, we presented the main contribution of this research in Chapter 6 and Chapter 7. Three core research questions are addressed in these two chapters.

As an answer to **RQ-1:** *Which ranking factors influence organic search results?*, we recommend Webmasters to give primary focus to the following list of points.

1. Backlink related factors yield relatively stronger positive correlation, hence we encourage Webmasters to continuously build backlinks.
2. Putting more content (text) on various tags(such as body, description tags etc.) of a page is well associated to high ranked webpages on Google.nl.

3. Linking from a site to it's Google+ and Facebook fan pages shows positive correlation to well ranking on Google.nl.

4. The length of a URL has negative correlation with well ranking, and should be regarded as bad practice of URL structuring.

5. Pagerank turns out to be negatively correlated to well ranked pages on the LETOR dataset, this could be an indication that Pagerank should not be the primary focus.

6. Optimizing textual content in different tags (body, title, anchor) could benefit from weighting schemes such as BM25[1] ans LMIR.

More specifically to the LeadQualifier.nl we recommend the following improvement ideas.

1. The LeadQualifier should add checks for more ranking factors to give a better advice on how to improve a website's visibility on search engines.

2. The LeadQualifier should revise it's score calculation for a webpage.

3. The LeadQualifier should be less dependent on external tools to gather factors such as the Pagerank and Backlinks. We suggest these factors should be produced internally.

To address the research question **RQ-2:** *Is it better to use only the top well ranked pages (e.g top 40) while computing correlation coefficients instead of using all ranked pages per search term?*, we computed Spearman rank correlation over the LETOR dataset, a dataset with relatively larger number of ranked relevant documents, and showed on our results that *considering few from the bottom and few from the top (40 for each in our case) of the ranked webpages gives stronger correlation coefficient.* This could bring a huge difference on the SEO today: assume a Webmaster decides to implement ranking factors which showed a correlation coefficient of 0.5 and above, more factors will make through his filter if the correlation coefficients are stronger. Therefore, we urge SEO companies which are publishing correlation studies regarding "ranking factors and well ranking" to reconsider their methodologies which only utilizes the top few ranked pages.

- **SRQ-3.1:** *Is there any sensible relationship between the calculated correlation coefficient of a ranking factor (first approach) and it's corresponding weight assigned by a ranker(second approach)?*

[1] BM25 weighting scheme (BM stands Best Match) is used to measure term weights, recent TREC tests have shown BM25 to be the best of the known probabilistic weighting schemes.(`http://xapian.org/docs/bm25.html`, July 29, 2014)

- **SRQ-3.2:** *Does considering highly correlated ranking factors give a better performing ranker, compared to using the whole set of ranking factors?*

Sub research question **SRQ-3.1** and **SRQ-3.2** are formulated to find an answer for the third research question **RQ-3:** *How can we evaluate the importance of ranking factors?*. We are not able to answer **SRQ-3.1** affirmatively, regardless we have deduced indirectly that *strong positive (if not just positive) correlation is not always a cause for well ranking*. By showing performance of rankers trained on Coordinate Ascent and LambdaMART, we are able to answer **SRQ-3.1** conditionally. We concluded that *prefiltering feature sets according to their mean Spearman/Biserial correlation coefficients might improve performance of trained ranker depending on the learning to rank algorithm used for training.*

8.2 Future Work

In this section of the chapter, we will present some possible research area that could be conducted, by extending this research or independently. Some of the research ideas included here were part of this research, but excluded later for various reasons.

8.2.1 Local Search Ranking Factors

We believe identifying ranking factors that influence local search is something that requires separate and further investigation. From the perspective of a business owner, Local Search is about being found online when people are looking for their company or services they offer. On the other hand, Local Search is mostly used to refer to getting businesses to rank in the listings that appear in the Search Engine Results Pages (SERPs) accompanied by Map pins[2]. Users usually expect to see local results, i.e. results that are optimized to their geographical location (neighbourhood, city, county, state)[3]. One example is when people search for "Amsterdam boot" which means "Amsterdam boat" they should be presented with result of boat rentals located in Amsterdam. Moz published major factors for local search ranking, the author of the blog pointed out that the primary factors for local search ranking seem to have remained largely the same for the past couple of years[4]. It would be very interesting to see similar research focused in The Netherlands by taking the results of this thesis as starting point.

[2] `http://localu.org/blog/what-is-local-search/`
[3] http://www.ngsmarketing.com/
[4] http://moz.com/blog/local-search-ranking-factors-2013

8.2.2 Long-Tail Keywords

As described in Chapter 4 a majority of the search terms used in this research contains two or three words (keywords). A similar research could be conducted by constructing set of search terms which are long-tail, with a goal of discovering a relationship between long-tail search terms and the ranking factors used in well ranked pages on a certain search engine. Long-tail keywords are longer, more specific keywords that are less common, individually, but add up to account for the majority of search-driven traffic[5]. Long-tail keywords are the opposite of "head" terms, which are more popular or more frequently searched on. For example, "fish tanks" is a head term, but "compare prices whisper aquarium filters" is a long-tail keyword. Long-tail keywords can offer incredible ROI[6] because they are less competitive to rank well on organic search result and less expensive to bid on for Pay Per Click (PPC).

8.2.3 Social Signals

Social signals also know as social metrics are measurement tools that can be used to define and articulate social value, social outcomes and the results generated by investment and activities in the social sector [39]. In other words social metric is an indicator that measures the social media activities of a website. For instance the number of likes on Facebook fan page of a given company is a good example of social signals. Social indicators that can be extracted from the site of a webpage such as existence of a link to Facebook fan page from a webpage, are included in this research. However social metrics that are external to the page were ignored, because it was difficult to collect the data. It would be nice to see a result that shows any direct/indirect relationship between these social signals and well ranking webpages.

8.2.4 Ground-Truth

While constructing a dataset suitable for LETOR we made an assumption and use the position of each webpage on Google.nl as ground-truth. It would be nice to see a research that finds a way to avoid this assumption and come up with more concrete result.

[5] `http://www.wordstream.com/long-tail-keywords`
[6] Return Over Investment

8.2.5 Special Sample

Considering only the top 40 search results (i.e. the approach of this research) of every search query could introduce some bias on the conclusion to be drawn from the experiment as it does not contemplate the pages ranking low on the result list. To avoid this, we suggest next researchers to collect a special dataset containing search terms with total number of search result in a defined range (e.g. in a range of 100 to 125 search terms). For instance, the search terms could be constructed based on the following scheme :
search terms = "restaurants in" + [Netherlands city name]
e.g. *"restaurants in Enschede", "restaurants in Hengelo", "restaurants in Lochem"* etc..

The main reason to follow this scheme for constructing the search terms is based on the following assumptions and criteria:

- Most cities have restaurants, therefore result is expected for all queries.
- Restaurants are very well engaged with social media activities, which suits with the desire to investigate on social media activities and ranking.
- A total of 143 cities (small, medium-sized and big cities[7][8]) from 12 provinces could be included in the dataset .
- An "exact match" search could be used to narrow down the search results, therefore analyze all ranking pages. Based on our quick search on Google.com, the search term "restaurants in Amsterdam" gives 149 results.
  N.B. The number of results Google display on the first result page is an estimation which is normally way bigger than the actual number. To see actual number of results found on a particular query, one should go to the very last of the result pages.

[7]The general consensus is that a city constitutes a population of more than 30,000-50,000 inhabitants. Cities with between 100,000 and 250,000 inhabitants are mostly called 'middelgrote steden' (medium-sized cities), while the use of 'grote steden' (big cities) is usually reserved for Amsterdam, Rotterdam, The Hague and Utrecht

[8]`http://nl.wikipedia.org/wiki/Nederlandse_gemeente`

Appendix A

LETOR Dataset

A.1 LETOR

LETOR [1] is a package of benchmark datasets for research on LEarning TO Rank, which contains standard features, relevance judgments, data partitioning, evaluation tools, and several baselines.

A.1.1 Documents

LETOR4.0 is the latest release of LETOR package and it uses the Gov2 webpage collection (25M pages) and two query sets from Million Query track of TREC 2007 (MQ2007) and TREC 2008(MQ2008).

A.1.2 Queries

There are about 1700 queries in MQ2007 with labeled documents and about 800 queries in MQ2008 with labeled documents.

A.1.3 Features

A query-document pair is represented by a 46-dimensional feature vector. Full list of the features is provided in Appendix B.

[1] `http://research.microsoft.com/en-us/um/beijing/projects/letor/letor4dataset.aspx`

A.1.4 Dataset Format

The dataset is formated according to SVMLight format, which is shown below.

```
<line> .=. <relevance> qid:<qid> <feature>:<value> ... <feature>:<value>
<relevance> .=. 0 | 1 | 2 | 3 | 4
<qid> .=. <positive integer>
<feature> .=. <positive integer>
<value> .=. <float>
```

Each row is a query-document pair. In following example, the first column is relevance label of this pair, the second column is query id, the following columns are features, and the end of the row is comment about the pair, including id of the document.

```
1 qid:10 1:0.004356 2:0.080000 3:0.036364 4:0.000000 ... 46:0.000000 #docid =
    GX057-59-4044939 inc = 1 prob = 0.698286
```

A.1.5 Relevance Judgment/Ground-Truth

The relevance of each document to the query has been judged by a professional editor who could give one of the 5 relevance labels of set (1). Each of these relevance labels is then converted to an integer ranging from 0 (for bad) to 4 (for perfect). There are some specific guidelines given to the editors instructing them how to perform these relevance judgments. However, in the listwise version of LETOR4.0 the ground truth is a permutation for a query instead of multiple level relevance judgments. As shown in the following example, the first column is the relevance degree of a document in ground truth permutation. Large value of the relevance degree means top position of the document in the permutation. The other columns are the same as that in the setting of supervised ranking.

```
1008 qid:10 1:0.004356 2:0.080000 3:0.036364 4:0.000000 ... 46:0.000000 #docid
    = GX057-59-4044939 inc = 1 prob = 0.698286
```

A.1.6 Data Partition

The 5-fold cross validation strategy is adopted and the 5-fold partitions are included in the package. In each fold, there are three subsets for learning: training set, validation set and testing set.

Appendix B

Correlation Coefficients Vs Weights

B.1 NDCG Calculation Example

Assume the the search term is $Query = abc$, then the NDCG is calculated as shown on the table[1].

Table B.1: Example of NDCG calculation explained.

#	URL	Gain	DCG	Max DCG	NDCG
1	http://abc.go.com/	Perfect: $31=2^5-1$	31	31	1 = 31/31
2	http://www.abcteach.com/	Fair: $3=2^2-1$	32.9	40.5	0.81=32.9/40.5
3	http://abcnews.go.com/sections/	Excellent: $15=2^4-1$	40.4	48.0	0.84=40.4/48.0
4	http://www.abc.net.au/	Excellent: 15	46.9	54.5	0.86=46.9/54.5
5	http://abcnews.go.com/	Excellent: 15	52.7	60.4	0.87=52.7/60.4

[1] `http://research.microsoft.com/en-us/people/tyliu/learning_to_rank_tutorial_-_www_-_2008.pdf`

B.2 Training on LETOR4.0 Dataset

Linux Terminal Command :

```
$java -jar RankLib-2.3.jar -train MQ2008-list/Fold1/train.txt -test
  MQ2008-list/Fold1/test.txt -validate MQ2008-list/Fold1/vali.txt -ranker 4
  -metric2t NDCG@10 -metric2T ERR@10 -gmax 1831 -norm zscore -save
  MQ2008-list/Fold1/Coordinate-Ascent-LETOR-FOLD1-MODEL.txt
```

General Parameters:

Training data: MQ2008-list/Fold1/train.txt

Test data: MQ2008-list/Fold1/test.txt

Validation data: MQ2008-list/Fold1/vali.txt

Feature vector representation: Dense.

Ranking method: Coordinate Ascent

Feature description file: Unspecified. All features will be used.

Train metric: NDCG@10

Test metric: ERR@10

Highest relevance label (to compute ERR): 1831

Feature normalization: zscore

Model file: Coordinate-Ascent-LETOR-FOLD1-MODEL.txt

Coordinate Ascent's Parameters:

No. of random restarts: 5

No. of iterations to search in each direction: 25

Tolerance: 0.001

Regularization: No

Training file : 471 ranked lists, 540679 entries

Validation file : 157 ranked lists, 180664 entries

Test file : 156 ranked lists, 180877 entries

NDCG@10 on training data: 89932.6968

NDCG@10 on validation data: 57694.4007

Table B.2: Mean of Spearman Rank Correlation Coefficient Vs Coordinate Ascent Feature Weight, LETOR4.0-MQ2008-list

Feature Id	**Mean Spearman Rank Correlation Coefficient**	**Weight Rank**	**Coordinate Ascent Feature Weight**	**Feature Description**
1	0.32	3	0.00074842260518108800	TF(Term frequency) of body
2	0.16	21	0.00007457482014089923	TF of anchor
3	0.12	17	0.00001457753149420800	TF of title
4	0.1	24	0.00000454607800768632	TF of URL
5	0.32	45	-0.19316287010457300000	TF of whole document
6	NaN	34	0.00000169788160884643	IDF(Inverse document frequency) of body
7	NaN	35	0.00000169788160884643	IDF of anchor
8	NaN	36	0.00000169788160884643	IDF of title
9	NaN	37	0.00000169788160884643	IDF of URL
10	NaN	38	0.00000169788160884643	IDF of whole document
11	0.44	10	0.00004733269455061640	TF*IDF of body
12	0.16	6	0.00019766829638811900	TF*IDF of anchor
13	0.12	8	0.00011352500239933000	TF*IDF of title
14	0.1	28	0.00000229214017194268	TF*IDF of URL
15	0.44	23	0.00007046208676712690	TF*IDF of whole document
16	0.04	44	-0.01716329632794800000	DL(Document length) of body
17	0.06	4	0.00044433870539862900	DL of anchor
18	0	26	0.00002689882271357710	DL of title
19	-0.04	19	0.00001104047516152390	DL of URL
20	0.04	46	-0.77589986883533300000	DL of whole document
21	0.72	43	-0.00000859382534984109	BM25 of body
22	0.78	15	0.00001684723026377870	BM25 of anchor
23	0.82	12	0.00003839830683167090	BM25 of title
24	0.78	20	0.00001059065367712920	BM25 of URL

Continued on next page

Table B.2 – continued from previous page

25	0.16	9	0.00008854452590134140	BM25 of whole document
26	0.16	33	0.00000178277568928875	LMIR.ABS of body
27	0.16	39	0.00000169788160884643	LMIR.ABS of anchor
28	0.16	22	0.00000739851911054832	LMIR.ABS of title
29	0.12	27	0.00000263596119773408	LMIR.ABS of URL
30	0.12	30	0.00000195256385017340	LMIR.ABS of whole document
31	0.12	13	0.00002334587212163840	LMIR.DIR of body
32	0.12	16	0.00001684723026377870	LMIR.DIR of anchor
33	0.1	29	0.00000226907503828958	LMIR.DIR of title
34	0.1	7	0.00011495082962292600	LMIR.DIR of URL
35	0.1	32	0.00000178277568928875	LMIR.DIR of whole document
36	0.1	14	0.00002161403250964790	LMIR.JM of body
37	0.72	42	0.00000107816079119115	LMIR.JM of anchor
38	0.78	40	0.00000169788160884643	LMIR.JM of title
39	0.82	2	0.00139251760149540000	LMIR.JM of URL
40	0.78	18	0.00001335931217524360	LMIR.JM of whole document
41	-0.06	5	0.00020975301036626300	PageRank
42	-0.04	31	0.00000195256385017340	Inlink number
43	NaN	41	0.00000169788160884643	Outlink number
44	0.02	11	0.00004033352842567750	Number of slash in URL
45	0.02	25	0.00000392953474847396	Length of URL
46	0.04	1	0.01013896935046290000	Number of child page

B.3 Training on DUTCH WEB Dataset

Linux Terminal Command :

```
$java -jar RankLib-2.3.jar -train TRAIN.txt -test TEST.txt -validate
  VALIDATION.txt -ranker 4 -metric2t NDCG@10 -metric2T ERR@10 -gmax 39 -norm
  zscore -save Coordinate-Ascent-DUTCH_WEB-Dataset-Model.txt
```

General Parameters:

Training data: TRAIN.txt

Test data: TEST.txt
Validation data: VALIDATION.txt
Feature vector representation: Dense.
Ranking method: Coordinate Ascent
Feature description file: Unspecified. All features will be used.
Train metric: NDCG@10
Test metric: ERR@10
Highest relevance label (to compute ERR): 39
Feature normalization: zscore
Model file: Coordinate-Ascent-Our-Dataset-Model.txt
Coordinate Ascent's Parameters:
No. of random restarts: 5
No. of iterations to search in each direction: 25
Tolerance: 0.001
Regularization: No
Training file : 4541 ranked lists, 180389 entries
Validation file : 1513 ranked lists, 60103 entries
Test file : 1514 ranked lists, 60147 entries

TABLE B.3: Mean of Spearman Rank Correlation Coefficient Vs Coordinate Ascent Feature Weight, DUTCH-WEB Dataset

Feature Id	**Mean Spearman Rank Correlation Coefficient**	**Weight Rank**	**Coordinate Ascent Feature Weight**	**Feature Description**
1	0.06	43	0.00018829836110803100	ALL LINKS COUNT ON PAGE
2	0.04	15	0.00108117989466699000	B, I, STRONG WORD COUNT ON PAGE
3	0.05	42	0.00029250150463424000	BODY WORD COUNT ON PAGE
4	0.02	32	0.00034248642913522300	EXTERNAL LINK COUNT ON PAGE

Continued on next page

Table B.3 – continued from previous page

5	0.06	33	0.00034248642913522300	IMAG TAGS COUNT ON PAGE
6	0.07	34	0.00034248642913522300	INTERNAL LINK COUNT ON PAGE
7	0.05	35	0.00034248642913522300	META DESC WORD COUNT ON PAGE
8	0.01	36	0.00034248642913522300	META KEYWORD WORD COUNT ON PAGE
9	-0.13	44	0.00004318878525543400	NO FOLLOW COUNT ON PAGE
10	0.01	27	0.00035961075059198500	TITLE WORD COUNT ON PAGE
11	-0.15	46	-0.00012454434460463300	URL DEPTH
12	0.08	24	0.00039385939350550700	BACKLINKS, NUMBER OF EXTERNAL LINKS
13	0.04	12	0.00163451648304785000	BACKLINKS, NUMBER OF INDEXED URLS
14	0.10	37	0.00034248642913522300	BACKLINKS, NUMBER OF REFFERING IP
15	0.10	38	0.00034248642913522300	BACKLINKS, NUMBER OF REFFERING DOMAINIS
16	-0.14	28	0.00035961075059198500	FACEBOOK, DOMAIN INSIGHT ON PAGE
17	-0.18	49	-0.00105051139872602000	GOOGLE ADSENSE SLOTS ON PAGE
18	-1.00	29	0.00035961075059198500	GOOGLE ANALYTICS TRACKING CODE ON PAGE
19	-0.11	39	0.00034248642913522300	LINKEDIN LINK ON PAGE
21	-0.03	17	0.00069203100617059800	OPEN GRAPH MARKUP ON PAGE
21	-0.68	25	0.00039385939350550700	PIN IT LINK ON PAGE
22	-0.08	19	0.00058621047480875900	SEARCH TERM IN <A>
23	-0.30	31	0.00035961075059198400	SEARCH TERM IN <B>OR <I>OR <STRONG>

Continued on next page

Table B.3 – continued from previous page

24	0.02	26	0.00037759128812158400	SEARCH TERM IN <BODY>
25	-0.08	23	0.00041355236318078200	SEARCH TERM IN <H1>
26	-0.29	9	0.00316865002644975000	SEARCH TERM IN <H2>
27	-0.49	11	0.00203043413544997000	SEARCH TERM IN <H3>
28	-0.68	51	-0.00631156192996317000	SEARCH TERM IN <H4>
29	-0.89	10	0.00222701800545179000	SEARCH TERM IN <H5>
30	-0.96	3	0.13096169079702800000	SEARCH TERM IN <H6>
31	-0.32	20	0.00053171018123243400	SEARCH TERM IN IMAGE ALT
32	-0.19	7	0.02318718746852750000	SEARCH TERM IN META DESC
33	-0.27	6	0.02343494084036740000	SEARCH TERM IN META KEYWORD
34	-0.18	22	0.00041629439515404600	SEARCH TERM IN <P>
35	-0.05	18	0.00059935125098664100	SEARCH TERM IN <TITLE>
36	-0.13	2	0.16669697775609000000	SEARCH TERM IN <UL>OR <OL>
37	-0.36	45	0.00000727917629081841	TWITTER CARD MARKUP ON PAGE
38	-0.14	40	0.00034248642913522300	TWITTER LINK ON PAGE
39	-0.05	41	0.00034248642913522300	SHEMA.ORG ON PAGE
40	0.02	47	-0.00040040649618066900	FACEBOOK, FOLLOW US LINK ON SITE
41	-0.65	13	0.00120619439261061000	GOOGLE PLUS, AUTHOR MARKUP ON SITE
42	-1.00	1	0.17186148750055900000	GOOGLE PLUS, COMMUNITIES LINK ON SITE
43	0.05	21	0.00050974823896413800	GOOGLE PLUS, FOLLOW LINK ON SITE
44	-0.96	48	-0.00077435223063660400	GOOGLE PLUS, PHOTO LINK ON SITE
45	-0.09	30	0.00035961075059198400	GOOGLE PLUS PLUSONE BUTTON ON SITE

Continued on next page

Table B.3 – continued from previous page

46	-0.02	14	0.00110203356227410000	GOOGLE PLUS, PUBLISHER MARKUP ON SITE
47	-0.72	5	0.04850178368068210000	GOOGLE PLUS, SHARE LINK ON SITE
48	-0.60	16	0.00072920170912033800	GOOGLE PLUS, USER PROFILE LINK ON SITE
49	-0.94	4	0.07739991164603140000	URL EXACT MATCH TO DOMAIN
49	1.00	50	-0.00419888345201863000	URL DOMAIN IS VALID
51	1.00	52	-0.31797042121401600000	URL HAS PUBLIC SUFFIX
52	-0.36	8	0.00327771717826669000	URL PARTIAL MATCH TO DOMAIN

Appendix C

Statistics of DUTCH WEB Dataset

C.1 Introduction

To give the reader a more deeper view of the DUTCH WEB dataset, we conducted this analysis by considering two subsets of it. As summarized in table C.1 the first set, named SET1, contains the top 10 ranked webpages for each search term, while the second set, named SET2 is just the whole dataset which contains top 40 ranked webpages for each search term. The reason for conducting the analysis this way is discussed in chapter 6. It should also be noted that, sometimes we present analysis results based on larger dataset (the original dataset before cleaning).

TABLE C.1: Basic statistics on the SET1 and SET2 sets

SET	# WEBPAGES	DESCRIPTION
SET1	75280	Prepared by taking the top 10 ranked webpages for each search term.
SET2	300639	Is just the whole dataset which contains top 40 ranked webpages for each search term.

In order to help the reader quickly see and understand the results depicted on bar graphs through out this section we provide the following information:

Legend: The percentage (maximum 100%) is plotted on the X-axis, which measures size/share/count of a particular factor in the analyzed dataset. On left side of the Y-axis, the name of each factor is written in a descriptive way. The length of the bar represents the percentage of a particular factor (with the longer the best). The bars are sorted in descending order, with the longer bar on the top.

Color Code: In each bar graphs which depicts two independent bars for each factors, the blue bar always represents value from SET1 and the red bar always represents value from SET2. However, the color code is not applicable for bar graphs which depicts only one bar for each factor.

C.2 URL Protocol Type

Based on the motivation elaborated in Section 4.2.1, in this section we discuses the survey result of of SSL implementation of the Dutch websites. We analyze URL protocol type used by websites on our dataset and find that the distribution is heavily skewed towards HTTP. First we used SET2 and the statistics shows that over 96% of the URLs uses HTTP protocol, and the rest 3% have implemented HTTPS. We wondered if this would change when we use SET1, unfortunately the results was same as the results from SET2 with only a negligible difference. It was striking to see almost similar proportion in both the sets, hence we performed statistical significance test which is presented in the next section. Then we wanted to see how many of the URLs with HTTPS protocol was ranked at position 1 on Google.nl (for SET1) and found out 2.99% got position 1.

The graph on Figure C.1 shows the distribution of URL protocol usage in our dataset, both for top 40 and top 10 ranked webpages of the search terms used on Google.nl.

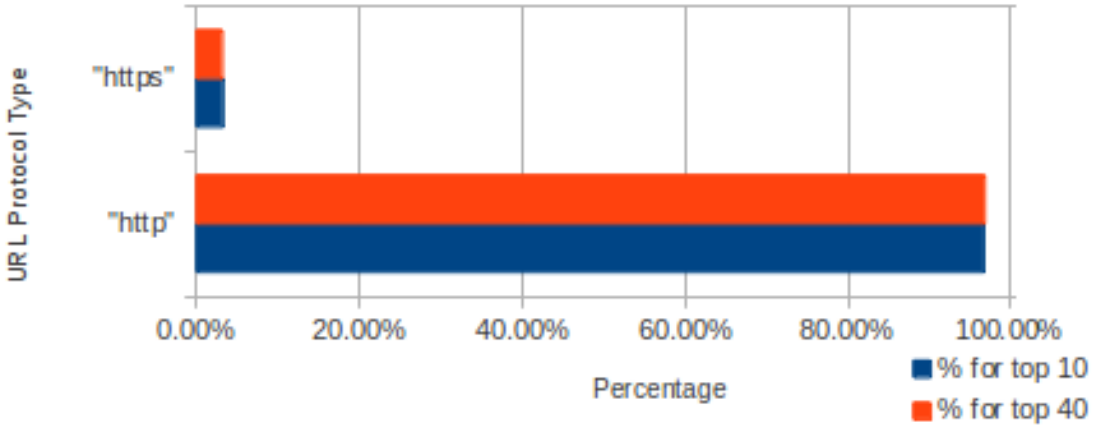


FIGURE C.1: Percentage of URLs categorized according to the their URL protocol type (HTTP or HTTPS), for top 10 webpages and for top 40 webpages.

C.2.1 Test for URL Protocol Type

We calculated P values to determine whether or not the proportion of HTTP to HTTPS observed on SET1 is equal to the proportion observed on SET2. The steps followed to conduct the significance test are clearly presented below.

1. Determine Expected Values

As we are trying to show that URLs with HTTPS protocol are preferred by search engines and as a result ranked higher, we expect to see more URLs that implemented HTTPS protocol on SET1. In other words the distribution we observed on SET2 (which is HTTP = 96% and HTTPS = 3%) will be reversed (percent of URLs with HTTPS will be greater than percent of URLs with HTTP on SET1).

H0(null hypothesis) : the percentage of URLs with HTTPS protocol and URLs with HTTP protocol on SET1 will remain the same as the percentage on SET2. In other words, the proportion will remain the same because there is no any kind of relationship between the data source used and the results observed.

H1(alternate hypothesis) : the percentage of URLs with HTTPS protocol is higher than URLs with HTTP protocol on SET1. In other words this hypothesis claims that there is a relationship between the data source used and the observed results.

We take a random sample of the dataset with 5000 URLs of SET2 and the proportion of HTTP to HTTPS URLs is 4885 by 115. Treating this proportion as

expected value, we wish to see same proportion on another random sample of the same size taken from SET1.

2. Determine Observed Result

Next we analyze the proportion of HTTP to HTTPS on another sample of the same size (5000 URLs) take from SET1. That will give us our actual (or "observed") values. Since we have changed the data source, the observed results might differ from the expected results. There are two possibilities to explain this: either this happened by chance, or changing the data source caused the difference.

We randomly select 5000 URLs from SET1. We find that 4867 were HTTP and 133 were HTTPS. These differ from our expected results of 4885 and 115, respectively. P value will help us determine whether our experimental manipulation (in this case, changing the source of our data from SET2 to SET1) cause this change in results, or is this a negligible difference and we're just observing a chance variation?

3. Determine Degrees of Freedom

The equation for degrees of freedom is $Degrees of Freedom = n - 1$, where n is the number of categories or variables being analyzed in our analysis. Our analysis has two categories of results: URLs with HTTP protocol and URLs with HTTPS protocol. Thus, we have 2-1 = 1 degree of freedom.

4. Calculate Chi Square

We calculated chi square to get the difference between the observed and expected values of the analysis. The formula for chi square is provided in Section 3.2.7. Putting the values we found in to the formula [3.5] we will get a chi square value of 2.88. $x^2 = ((4867 - 4885)^2/4885) + (133 - 115)^2/115) = 2.88$.

5. Determine the Significance Level

As explained in previous chapter(Section 3.2.6) significance level is a measure of how confident we want to be with our result. For our analysis we choose a significance level of 0.05. Which means there is a 5% probability that the difference occurred was due to pure chance. In other words, there is 95% probability that the difference in the result was caused by the data source used rather than chance.

6. Calculate the P Value

We used chi square distribution table provided by Medcal.org[1], and the degree of freedom (i.e. 1) to approximate the P value. According to this read, the P value lie between 0.05 and 0.1.

7. Decision

Our P value is between 0.05 and 0.1 which is greater than our significance level

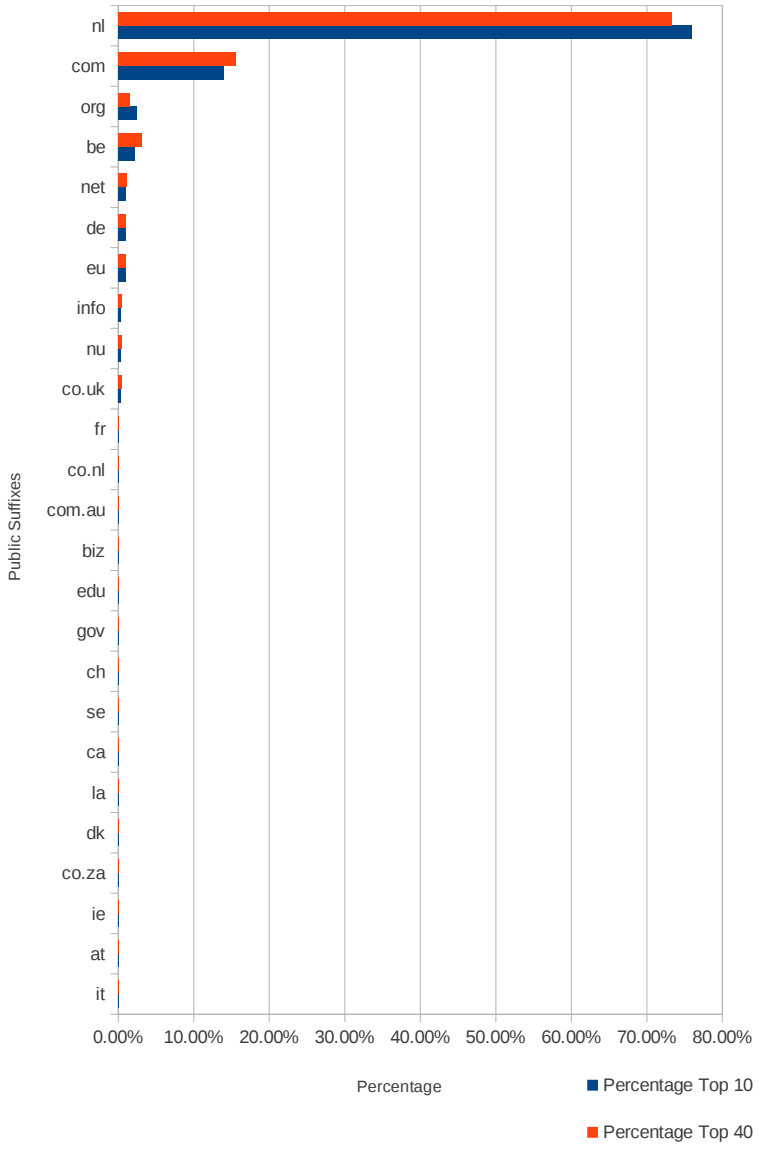
[1] `http://www.medcalc.org/manual/chi-square-table.php`

(i.e. 0.05), in this case we can't reject the null hypothesis were we claimed there is no significant relationship between the data source used and the observed results. This leads as to make the following decision : *at the 5% significance level the data do provide sufficient evidence to conclude that there is no noticeable relationship between the type of protocol implemented in a URL (HTTP or HTTPS) and it's rank on our dataset.*

C.3 Public Suffixes

The graph on Figure C.2 depicts the distribution of public suffixes also know as effective top-level domains(eTLDs) in the DUTCH WEB dataset, both for SET1 and SET2. The .nl TLD was found to be the most abundant in both SET1 and SET2, with 76% and 73% coverage respectively. The second most abundant TLD is .com with 14% share on SET1 and 16% share on SET2. The remaining percent, for both sets, is shared among other different TLDs and eTLDs, all of them holding less than 2.5% share.

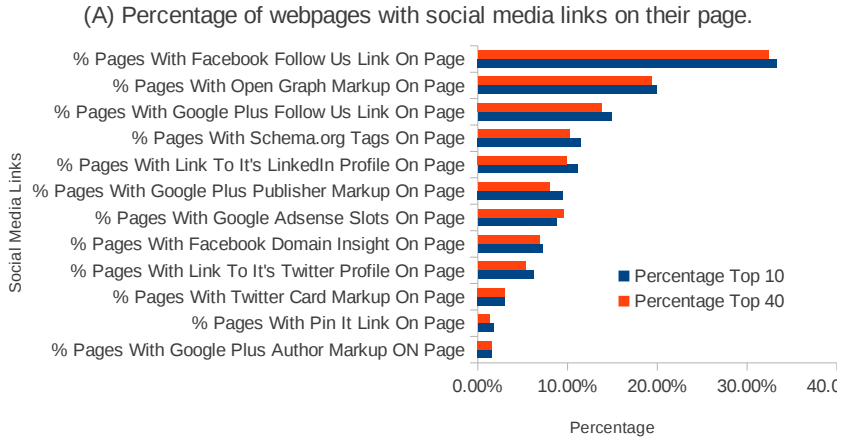
FIGURE C.2: Top 25 eTLDs found in our dataset, both for top 10 and top 40 ranked webpages.

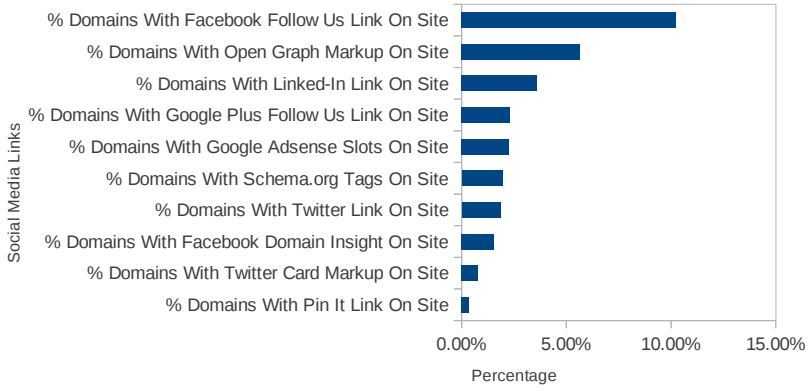


C.4 Social Media Links On Page

We extracted several social media links from each of the webpages that was downloaded, but only 12 factors are deemed as important and presented here. For instance Figure C.3(A) shows, on average 33% of the webpages (for both SET1 and SET2) have a "Follow Us On Facebook" link on their page. Separate analysis is conducted to find out how many of the websites (based on domain name) have a link to their profile/company pages of major social media sites like Facebook, Linkedin, Twitter etc. As depicted on Figure C.3(B) the percentage of domain names with a "Follow Us On Facebook" link is the largest (around 10%) coverage. It is important to note that the percentages displayed in this bar graphs do not add up to 100%, because the factors are analyzed independently.

FIGURE C.3: Percentage of webpages(A) and domain names(B) with social media links on their page.





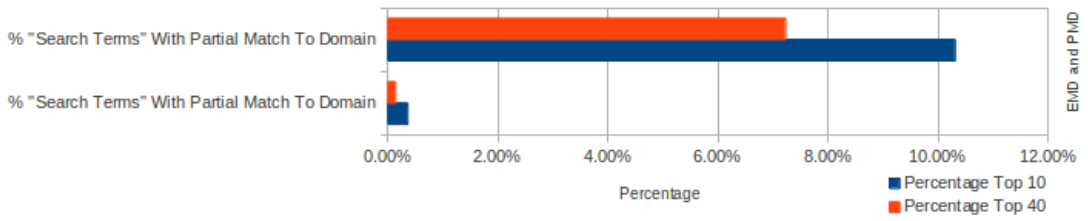


FIGURE C.4: Percentage of search terms which have either exact math or partial match with domain name of ranked webpages (EMD and PMD).

C.5 EMD and PMD

The rules used to determine whether or not a URL is EMD or PMD is clearly explained in Chapter 4, Figure C.4 we show the percentage of search terms which have either EMD or PMD.

C.6 Top Domain Names

Figure C.5 depicts the top 20 domain names in the DUTCH WEB dataset. The www.marktplaats.nl comes first on our dataset, similarly Alexa.com[2] puts this site as the most visited Dutch originated website in The Netherlands. Inaddition, 12 of the domains are with .nl TLDs.

[2] `http://www.alexa.com/topsites`(June 19, 2014)

FIGURE C.5: Percentage of top 20 domains in the TOP40 set.

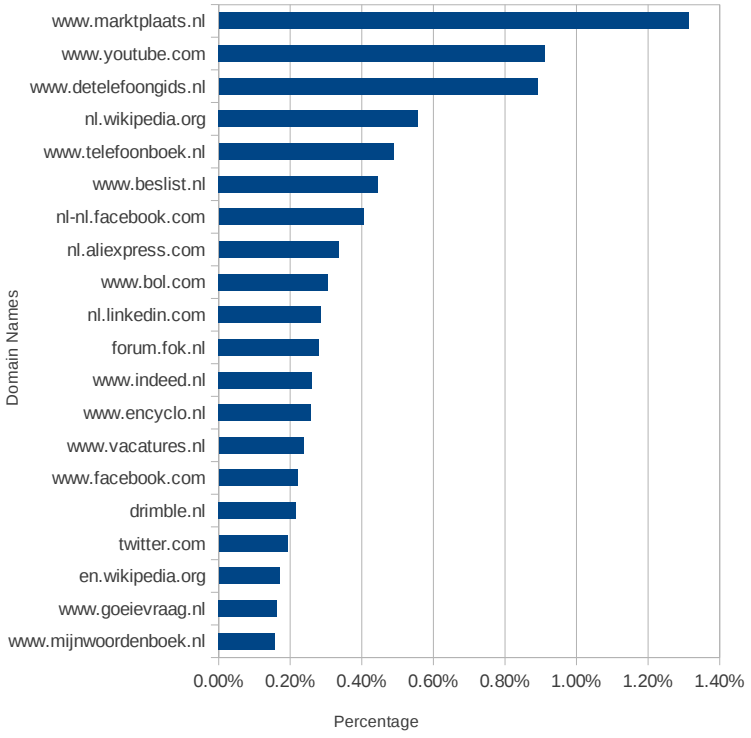


Table C.2: Backlink related data

FACTORS	AVG	MAX	# WEBPAGES
External Backlinks to a Page	221364987	29374101965	300639
Referring IPs to a Page	29656	2087209	300639
Referring Domains to a Page	175925	16151331	300639
Indexed URLs per Domain of a Page	17954677	2944202343	300639

C.7 Backlinks

Table C.2 shows, the maximum number of backlinks for a webpage found in our dataset is around 29 billion. The average number of pages index per domain is around 18 million, which reflect the influence of the globally popular websites.

C.8 List of All Ranking Factors

Description of a database table storing all ranking factors analyzed in this research is given in the table below. The reader should be reminded of the abbreviations provided in the beginning of this document:

Field	Type	Null	Key	Default
ID	int(10)	NO	PRI	NULL
SEARCHTERM	varchar(350)	NO		NULL
POSITION	int(2)	NO		NULL
FB_FOLLOW_US_LINK_ON_SITE	bit(1)	NO		b'0'
GP_PLUSONE_BUTTON_ON_SITE	bit(1)	NO		b'0'
GP_FOLLOW_LINK_ON_SITE	bit(1)	NO		b'0'
GP_SHARE_LINK_ON_SITE	bit(1)	NO		b'0'
GP_COMMUNITIES_LINK_ON_SITE	bit(1)	NO		b'0'
GP_PHOTO_LINK_ON_SITE	bit(1)	NO		b'0'
GP_USER_PROFILE_LINK_ON_SITE	bit(1)	NO		b'0'
GP_PUBLISHER_MARKUP_ON_SITE	bit(1)	NO		b'0'
GP_AUTHOR_MARKUP_ON_SITE	bit(1)	NO		b'0'
URL_EXACT_MATCH_TO_DOMAIN	bit(1)	NO		b'0'
URL_PARTIAL_MATCH_TO_DOMAIN	bit(1)	NO		b'0'
URL_HAS_PUBLIC_SUFFICS	bit(1)	NO		b'0'
URL_DOMAIN_IS_VALID	bit(1)	NO		b'0'
CONTENT_ST_IN_TITLE	bit(1)	NO		b'0'
CONTENT_ST_IN_BODY	bit(1)	NO		b'0'
CONTENT_ST_IN_META_DESC	bit(1)	NO		b'0'
CONTENT_ST_IN_META_KEYWORD	bit(1)	NO		b'0'
CONTENT_ST_IN_P	bit(1)	NO		b'0'
CONTENT_ST_IN_A	bit(1)	NO		b'0'
CONTENT_ST_IN_UL_OL	bit(1)	NO		b'0'
CONTENT_ST_IN_B_I_STRONG	bit(1)	NO		b'0'
CONTENT_ST_IN_IMAGE_ALT	bit(1)	NO		b'0'
CONTENT_ST_IN_H1	bit(1)	NO		b'0'
CONTENT_ST_IN_H2	bit(1)	NO		b'0'
CONTENT_ST_IN_H3	bit(1)	NO		b'0'
CONTENT_ST_IN_H4	bit(1)	NO		b'0'
CONTENT_ST_IN_H5	bit(1)	NO		b'0'
CONTENT_ST_IN_H6	bit(1)	NO		b'0'
CONTENT_GOOGLE_ADSENSE_SLOTS_ON_PAGE	bit(1)	NO		b'0'
CONTNET_SHEMADOTORG_ON_PAGE	bit(1)	NO		b'0'
CONTENT_OPEN_GRAPH_MARKUP_ON_PAGE	bit(1)	NO		b'0'
CONTENT_TWITTER_CARD_MARKUP_ON_PAGE	bit(1)	NO		b'0'
CONTENT_PIN_IT_LINK_ON_PAGE	bit(1)	NO		b'0'
CONTENT_FB_DOMAIN_INSIGHT_ON_PAGE	bit(1)	NO		b'0'
CONTENT_LINKEDIN_LINK_ON_PAGE	bit(1)	NO		b'0'
CONTENT_TWITTER_LINK_ON_PAGE	bit(1)	NO		b'0'
CONTENT_GOOGLE_ANALYTICS_TRAKING_CODE_ON_PAGE	bit(1)	NO		b'0'
URL_ST_OCCURANCE_COUNT_IN_URL_PATH	int(11)	NO		NULL
URL_DEPTH	int(11)	NO		NULL
CONTENT_TF	int(11)	NO		NULL
CONTENT_TITLE_WORD_COUNT	int(11)	NO		NULL
CONTENT_BODY_WORD_COUNT	int(11)	NO		NULL
CONTENT_META_DESC_WORD_COUNT	int(11)	NO		NULL
CONTENT_META_KEYWORD_WORD_COUNT	int(11)	NO		NULL
CONTENT_B_I_STRONG_WORD_COUNT	int(11)	NO		NULL
CONTENT_IMAG_TAGS_COUNT	int(11)	NO		NULL
CONTENT_NO_FOLLOW_COUNT	int(11)	NO		NULL
CONTENT_EXTERNAL_LINK_COUNT	int(11)	NO		NULL
CONTENT_INTERNAL_LINK_COUNT	int(11)	NO		NULL
CONTENT_ALL_LINKS_COUNT	int(11)	NO		NULL
BL_NUMBER_OF_EXTERNAL_LINKS	bigint(20)	NO		NULL
BL_NUMBER_OF_REF_DOMAINIS	bigint(20)	NO		NULL
BL_NUMBER_OF_INDEXED_URLS	bigint(20)	NO		NULL
BL_NUMBER_OF_REF_IP	bigint(20)	NO		NULL
URL_PROTOCOL	varchar(100)	NO		NULL
URL_PUBLIC_SUFFICS	varchar(100)	NO		NULL
URL	varchar(2000)	NO		NULL
FILENAME	varchar(350)	NO		NULL

FIGURE C.6: Table description of raking factors database table

Bibliography

[1] Albert Bifet, Carlos Castillo, Paul-Alexandru Chirita, and Ingmar Weber. An analysis of factors used in search engine ranking. In *First International Workshop on Adversarial Information Retrieval on the Web*, April 2005.

[2] Ao-Jan Su, Y Charlie Hu, Aleksandar Kuzmanovic, and Cheng-Kok Koh. How to improve your google ranking: Myths and reality. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pages 50–57. IEEE, August 2010.

[3] Michael P Evans. Analysing google rankings through search engine optimization data. *Internet research*, 17(1):21–37, 2007.

[4] Marcus Tober, Dr. Leonhard Hennig, and Daniel Furch. Seo ranking factors – rank correlation 2013 bing usa. *SEO Ranking Factors – Rank Correlation 2013 Bing USA*, 1(1), August 2013.

[5] Netmark.com. Google search engine rankings correlational study. June 2013. URL `http://www.netmark.com/google-ranking-factors/google-ranking-factors-methodology`.

[6] Matthew Peters. 2013 search engine ranking factors @ONLINE, July 2013. URL `http://moz.com/blog/ranking-factors-2013`.

[7] Google. What is webmaster tools? @ONLINE, 2014. URL `https://support.google.com/webmasters/answer/4559176?hl=en&ref_topic=3309469`.

[8] V Dang and W Bruce Croft. Feature selection for document ranking using best first search and coordinate ascent. In *Sigir workshop on feature generation and selection for information retrieval*, 2010.

[9] Hamed Valizadegan, Rong Jin, Ruofei Zhang, and Jianchang Mao. Learning to rank by optimizing ndcg measure. In *NIPS*, volume 22, pages 1883–1891, 2009.

[10] Google Inc. Search engine optimization starter guide. 2010.

[11] Nick Craswell and David Hawking. Web information retrieval. In Ayse Göker and John Davies, editors, *Information Retrieval: Searching in the 21st Century*, pages 85–101. Wiley, UK, 2009.

[12] Junaidah Mohamed Kassim and Mahathir Rahmany. Introduction to semantic search engine. In *Electrical Engineering and Informatics, 2009. ICEEI'09. International Conference on*, volume 2, pages 380–386. IEEE, 2009.

[13] Zoltan Gyongyi and Hector Garcia-Molina. Web spam taxonomy. In *First international workshop on adversarial information retrieval on the web (AIRWeb 2005)*, 2005.

[14] Fen Xia and Jue Wang. Listwise approach to learning to rank - theory and algorithm. *Analysis*, pages 1192–1199, 2008. URL `http://portal.acm.org/citation.cfm?id=1390306`.

[15] Olivier Chapelle. Yahoo ! learning to rank challenge overview. 14:1–24, 2011.

[16] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM, 2007.

[17] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[18] Yining Wang, Wang Liwei, Yuanzhi Li, Di He, Wei Chen, and Tie-Yan Liu. A theoretical analysis of ndcg ranking measures. In *26th Annual Conference on Learning Theory*, 2013.

[19] Jen-yuan Yeh, Jung-yi Lin, Hao-ren Ke, and Wei-pang Yang. Learning to rank for information retrieval using genetic programming. (2).

[20] Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13 (4):346–374, 2010. ISSN 1386-4564. doi: 10.1007/s10791-009-9123-y. URL `http://dx.doi.org/10.1007/s10791-009-9123-y`.

[21] Jerrold H Zar. Significance testing of the spearman rank correlation coefficient. *Journal of the American Statistical Association*, 67(339):578–580, 1972. URL `http://www.jstor.org/discover/10.2307/2284441?uid=3738736&uid=2&uid=4&sid=21103350012761`.

[22] Hervé Abdi. The kendall rank correlation coefficient. *Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, CA*, pages 508–510, 2007.

[23] Helena Chmura Kraemer. *Biserial Correlation*. John Wiley and Sons, Inc., 2004. ISBN 9780471667193. doi: 10.1002/0471667196.ess0153.

[24] Peter J Huber. *Robust statistics*. Springer, 2011.

[25] Steven N. Goodman. Toward evidence-based medical statistics. 1: The p value fallacy. *Annals of Internal Medicine*, 130(12):995–1004, 1999. doi: 10.7326/0003-4819-130-12-199906150-00008. URL `http://dx.doi.org/10.7326/0003-4819-130-12-199906150-00008`.

[26] Stephen Stigler. Fisher and the 5 *CHANCE*, 21(4):12–12, 2008. ISSN 0933-2480. doi: 10.1007/s00144-008-0033-3. URL `http://dx.doi.org/10.1007/s00144-008-0033-3`.

[27] Larry Masinter, Tim Berners-Lee, and Roy T Fielding. Uniform resource identifier (uri): Generic syntax. 2005.

[28] Paul V Mockapetris. Domain names: Implementation specification. 1983.

[29] Zakir Durumeric, James Kasten, Michael Bailey, and J Alex Halderman. Analysis of the https certificate ecosystem. In *Internet Measurement Conference*, 2013.

[30] Trustworthy internet movement @ONLINE, May 2014. URL `https://www.trustworthyinternet.org/ssl-pulse/`. Last Checked : May 19, 2014.

[31] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *In Proceedings of the Nineth Annual ACM-SIAM Symposium on Discrete Algorithms.*, 1(1), May 1998.

[32] Bruce Ratner. *Statistical and machine-learning data mining: Techniques for better predictive modeling and analysis of big data*. CRC Press, 2011.

[33] Introducing Letor Datasets, Tao Qin, and Tie-yan Liu. Introducing LETOR 4.0 Datasets. 2009.

[34] Chengxiang Zhai and John Lafferty. A Study of Smoothing Methods for Language Models Applied to Information Retrieval. 22(2):0–33, 2004.

[35] Stephen Robertson and Hugo Zaragoza. On rank-based effectiveness measures and optimization. *Information Retrieval*, 10(3):321–339, 2007.

[36] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 621–630. ACM, 2009.

[37] Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11:23–581, 2010.

[38] Donald Metzler and W Bruce Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274, 2007.

[39] Tessa Hebb. Report on social metrics : Key informant interviews. April 2011.