Anna Paananen

Comparative Analysis of Yandex and Google Search Engines

Helsinki Metropolia University of Applied Sciences
Master's Degree
Information Technology
Master's Thesis
26 May 2012

PREFACE

Working in NetBooster Finland as an International Project Manager specialized in Russian market I've been asked many times about differences between the search engines Yandex and Google. This Master's Thesis is the outcome of my professional experience in the Search Engine Optimisation field in Russia and Finland. I would like to thank all the people from NetBooster Finland and Helsinki Metropolia University of Applied Sciences who has helped me in the development of the study.

Special thanks to my instructors Timo-Pekka Jäntti and Ville Jääskeläinen for all the support, both in technical and non-technical matters. I would like to thank also my collegues from NetBooster Finland for their help and support while writing the thesis.

Last but not least I would like to thank my mother Tamara Kapitonova, who always has been my prior motivator for the education, and of course to my lovely husband Jukka Paananen for his inconditional support and patience.

Helsinki, May 26, 2012

Anna Paananen

Author(s) Title Number of Pages Date	Anna Paananen Comparative Analysis of Google and Yandex Search Engines 51 pages + 1 appendix 26 May 2012
Degree	Master's Degree
Degree Programme	Degree Programme in Information Technology
Specialisation option	
Instructor	Timo-Pekka Jäntti, Supervisor

This thesis presents a comparative analysis of algorithms and information retrieval performance of two search engines: Yandex and Google in the Russian language.

Comparing two search engines is usually done with user satisfaction studies and market share measures in addition to the basic comparison measures. Yandex is the most popular search engine in Russia, while Google is the most popular search engine in the world and well known for the quality of the results. The most common opinion about the reason for the popularity of Yandex s that it retrieves better quality results specifically in the Russian language.

The comparison of the performance of some search engines in the English language has been studied mostly by comparing the relevancy of the results retrieved. There is a number of studies having been done on understanding the mathematical aspects of Google's algorithm and the ranking factors. No studies on comparing algorithms and the quality of retrieved results of Yandex and Google have been done.

This study is the comparison of the algorithms and the retrieved results of Yandex and Google search engines in the Russian language. The comparison can be divided in three main tasks, description of web information retrieval, comparison of PageRank and MatrixNet algorithms, and the comparison of the quality of the retrieved results for selected queries.

The main contributions of this thesis are the comparison of the ranking methods of both of the search engines, the quality of the results, and the main ranking factors of Yandex and Google.

Key words	World Wide Web, Web search, Search Engines, Google, Yandex, information retrieval, algorithms

Contents

1. Introduction 1

2. Web Search Engines 3

2.1. Traditional Information Retrieval 3
2.1.1. Boolean Search Engines 3
2.1.2. Vector Space Model Search Engines 4
2.1.3. Probabalistic Model Search Engines 4
2.2. Web Information Retrieval 5
2.2.1. History of Web Search Engines 6
2.2.2. Elements of Web Search Process 6
2.2.3. Crawling, Indexing and Query Processing 8

3. Google and Yandex Algorithms 12

3.1. Google Search Engine 12
3.1.1. History of Google Inc. 13
3.1.1. Mathematics of Google's PageRank 16
3.2. Yandex Search Engine 23
3.2.1. History of Yandex 24
3.2.1. Description of MatrixNet 25

4. Search Engine Optimisation 35

5. Results and Analysis 39

5.1. Test Queries 39
5.2. Test Enviroment 41
5.3. Response Time 41
5.4. Precision 42

Discussions and Conclusions 48

References 50

List of Figures

Figure 1: Elements of search engine 7
Figure 2: Estimated size of Google's index 16
Figure 3: Directed graph representing the Web of six pages 18
Figure 4: Simple graph with rank sink 22
Figure 5: Simple graph with cycle 22
Figure 6: Example of the decision tree 32
Figure 7: Greedy split for 1-region tree 33
Figure 8: The structure of the split conditions for one layer 33
Figure 9: Search Ranking Factors 2011 by SEOmoz 37

List of Tables

Table 1: Example of calculation of PageRank 20
Table 2: The description of basics SEO factors 36
Table 3: Selected queries for the test and their popularity 40
Table 4: Response time during off-peak hours 42
Table 5: Response time during peak hours 42
Table 6: Precision scores for Group A 43
Table 7: Precision scores for Group B 44
Table 8: Precision scores for Group C 45
Table 9: Precision scores for Group D 46
Table 10: Mean precision scores for each query and groups 47

Abbreviations

FTP	File transfer protocol
GDN	Discounted Cumulative Gain
IP	Internet Protocol
MAP	Mean Average Precision
nGDN	Normalized Discounted Cumulative Gain
PR	PageRank
SE	Search Engines
SEO	Search Engine Optimisation
SERP	Search Engine Results Page
TCI	Thematic Citation Index
URI	Uniform Resource Indetifier
URL	Uniform Resource Locator
WWW	World Wide Web

Appendices

Appendix 1: The list of all retrieved results and their relevancy

1. Introduction

Searching on the World Wide Web has become a part of our daily life as the Web is now a necessary tool for collecting information and it provides convenience in information retrieval because it can combine information from many different web sites. The ultimate goal in designing and publishing a web page is to share information. However, the high number of web pages added to the Web on a daily basis has made the Web a space of all kinds of data and information, which provides a challenge for information retrieval. The amount of information on the Web, as well as the number of hosts and domain names registered worldwide, are growing rapidly. To overcome these retrieval problems, more than 20 companies and institutions have developed search tools, such as Yahoo, AltaVista, Google, Yandex and many others.

Google is the most popular search engine in the World. In the first quater of 2012 Google had 91.7% of the overall search engine market share in the World. Google is also the most popular search engine in Europe with the 94.51% of the market share. But in some countries the local search engines perform better. For instance in China Baidu shares 67.4% of the search, while Google has only 16.1% of the market share. In South Korea local search engine called Naver shares 61.9% of the market, and Google is the third popular search engine with only 7.2% of the market share.

In Russia the most popular search engine is Yandex, it shares 60.4% of the market, while Google.ru has 26.2%. The reason for Yandex being the most popular search engine in Russia in the opinion of Internet Marketers is that Yandex retrieve better results compared to Google, but no studies have yet been published on that subject. There is a number of studies having been done on understanding the mathematical aspects of Google's algorithm and the ranking factors. No studies on comparing the algorithms and the quality of the retrieved results of Yandex and Google have been done so far.

The main contributions of this thesis are the comparison of ranking methods of both of the search engines, the quality of the results, and the main ranking factors of Yandex and Google.

The present study shows the comparison of Google and Yandex, the most popular search engine in Russia. The objective is to analyse which search engine performs better in the Russian language by comparing their algorithms and search results. The steps to achieve this goal were the comparison of mathematical aspects of Google's and Yandex' formulas, described in Chapter three and comparing the relevancy of the results retrieved.

2. Web Search Engines

Information retrieval is the process of finding material within large document collections for a particular query. Information retrieval used to be an activity that only a few people engaged in: reference librarians and similar professional searchers. Now the world has changed, and hundreds of millions of people engage in information retrieval every day when they use a web search engine. Traditional information retrieval is search within limited, controlled, nonlinked collections, whereas web information retrieval is search within the world's largest and linked document collections. Web search engines practically became the most visible information retrieval applications. The next section explains the basic models of traditional information retrieval search.

2.1. Traditional Information Retrieval

There are three basic computer-aided techniques for searching traditional information retrieval collections: Boolean models, vector space models, and probabilistic models. These search models, which were developed in the 1960s, have had decades to grow into new search models. In fact, according to Langville and Meyer (2006), in February 2003, there were at least 150,000 different search engines, which means that there are possibly 150,000 search models. Manning *et al.* (2008) describes traditional information retrieval models and points, that nevertheless, most search engines rely on one or more of the three basic models, which are described below.

2.1.1. Boolean Search Engines

The Boolean model of information retrieval, one of the earliest and simplest retrieval models, uses the notion of exact matching to match documents to a user query. The adjective Boolean refers to the use of Boolean algebra, whereby words are logically combined with the Boolean operators *and*, *or* and *not*. Any number of logical statements can be combined using three Boolean operators. The Boolean model of information retrieval operates by considering which keywords are present or absent in a document. Thus, a document is judged as relevant and irrelevant; there is no

concept of a partial match between documents and queries. This can lead to poor performance. The main drawback of Boolean search engines is that they fall prey to two of the most common information retrieval problems, synonymy and polysemy.

2.1.2. Vector Space Model Search Engines

Another information retrieval model uses the vector space model, developed by Gerard Salton in the early 1960s. Manning et al. (2008) points, that this model was developed to sidestep some of the information retrieval problems mentioned above. Vector space models transform textual data into numeric vectors and matrices, then employ matrix analysis techniques to discover key features and connections in the document collection. Some advanced vector space models address the common text analysis problems of synonymy and polysemy. Two additional advantages of the vector space model are relevance scoring and relevance feedback. The vector space model allows documents to partially match a query by assigning each document a number between 0 and 1, which can be interpreted as the likelihood of relevance to the query. The group of retrieved documents can be then be sorted by degree of relevancy, which is not possible with the simple Boolean model. Thus, vector space models return documents in an ordered list, sorted according to a relevance score. A drawback of the vector space model is its computational expense. At query time, similarity measures must be computed between each document and the query. Golub and Van Loan (1996) analyzed matrix computation and draw a conclusion that advanced models require an expensive singular value decomposition of a large matrix that numerically represents the entire document collection. As the collection grows, the expense of this matrix decomposition becomes prohibitive.

2.1.3. Probabalistic Model Search Engines

Probabilistic models attempt to estimate the probability that the user will find a particular document relevant. Langville and Meyer (2006) describe probabilistic models in their work. Retrieved documents are ranked by their differences of relevance. The relevance in this model is the ratio of the probability that the document is relevant to the query divided by the probability that the document is not relevant to the query. The probabilistic model operates recursively and requires that the underlying algorithm

guess at initial parameters then iteratively tries to improve this initial guess to obtain a final ranking of relevancy probabilities. Unfortunately, probabilistic models can be very hard to build and program. Their complexity grows quickly, limiting for many researchers their scalability. Probabilistic models also require several unrealistic simplifying assumptions, such as independence between terms as well as documents. On the other hand, the probabilistic framework can accommodate preferences, and thus, these models do offer promise of tailoring search results to the preferences of individual users. For example, a user's query history can be incorporated into the probabilistic model's initial guess, which generates better query results than a demographic guess. Web search engines practically became the most visible information retrieval applications, which have even more challenges than any of traditional information retrieval models. An introduction to web information retrieval and its challenges is given in the next section.

2.2. Web Information Retrieval

World Wide Web entered the information retrieval world in 1989 and created challenge for many web search engines built on the techniques of traditional search engines, because they differ in many ways. The main difference is that Web is a unique document collection, because it is huge, dynamic, self-organized and hyperlinked.

An additional information retrieval challenge for any document collection, especially to the Web, concerns accuracy. Although the amount of accessible information continues to grow, a user's ability to look at documents does not. sers rarely look beyond the first 10 or 20 documents retrieved. This user impatience means that search engine accuracy must increase just as rapidly as the number of documents is increasing. Edosomwan and Edosomwan (2010) mentioned, that another dilemma to web search engines concerns their performance measurements and comparison. While traditional search engines are compared by running tests on familiar, well studied, controlled collections, this is not realistic for web engines. Even small web collections are too large for researchers to create estimates of the precision and recall numerators and denominators for many queries.

2.2.1. History of Web Search Engines

Web search engines began to appear in 1994 when the number of Internet resources increased. However, Internet search engines were in use before the emergence and growth of the Web. The first pre-Web search engine was Archie, created in 1990 by Alan Emtage, a student at McGill University in Montreal. Archie allowed keyword searches of a database of names of files available via FTP. Bill Slawski (2006) points out that Archie allowed users to look around the Internet by the file name, and did not index the content of text files like most search engines do. The first robot and search engine of the Web was Wandex, which was developed by Matthew Gray in 1993. Since the appearance and exponential growth of the Web, hundreds of search engines with different features have appeared.

Primary search engines were designed based on traditional information retrieval methods. AltaVista, Lycos and Excite made huge centralized indices of Web pages. To answer a query, they simply retrieved results from their indexed databases and showed the cached pages based on keyword occurrence and proximity. While traditional indexing models have been successful in databases, it was revealed that these methods are not sufficient for a tremendously unstructured information resource such as the Web. The completeness of the index is not the only factor in the quality of search results. Since then the quality of search has been dramatically increased by many other search engines, including Google's innovative ranking system PageRank. Levy (2011) points out, that nowadays there are more than 100 web search engines, which are using different algorithms. In order to analyse how search engines work, the following sections describe the basics of the web search process.

2.2.2. Elements of Web Search Process

There are different ways to organise web content but every search engine has the same basic parts which include a crawler or spider, an index or catalogue, and an interface or query module. Users enter a search term through a predefined query module, specific to each search engine. Typically, the search engine works by sending out a spider to fetch as many documents as possible. Then another program called an

indexer reads these documents and creates an index based on the words contained in each document. Basic elements of the web information retrieval process and their relationship one to another are shown in Figure 1.

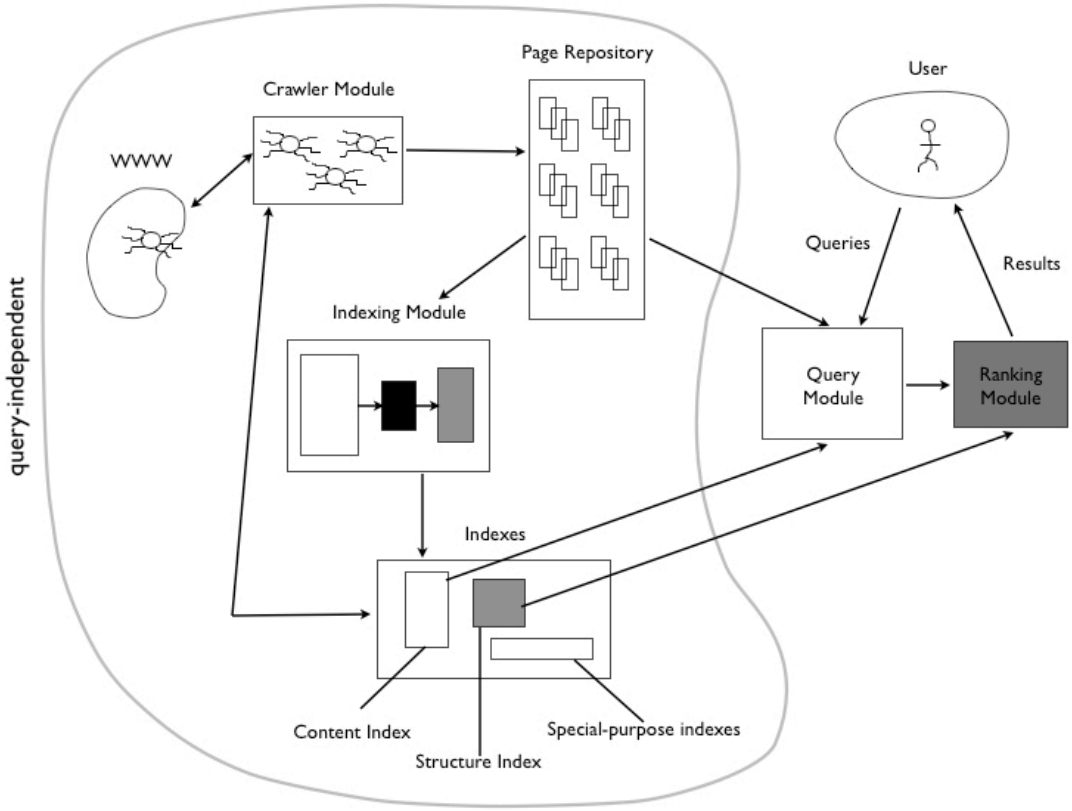

Figure 1. Elements of search engine

The basic elements of the web information retrieval process have been studied and described by Langville and Meyer (2006). Crawler module contains the software that collects and categorizes the web's documents. The crawling software creates virtual robots, called spiders that constantly scour the Web gathering new information and web pages and returning to store them in a central repository. The spiders return with new web pages, which are temporarily stored as full, complete web pages in the page repository. The new pages remain in the repository until they are sent to the indexing module. The indexing module takes each new uncompressed page and extracts only the vital descriptions, creating a compressed description of the page that is stored in various indices. The indices hold the valuable compressed information for each web page. There are three main indices. The first is called the content index. The content, such as keywords, title, and anchor text for each web page, is stored in a compressed form using an inverted file structure. Further valuable information regarding the

hyperlink structure of pages in the search engine's index is gleaned during the indexing phase. This link information is stored in compressed form in the structure index. The crawler module sometimes accesses the structure index to find uncrawled pages. Special-purpose indices are the final type of index (image index, pdf index etc).
The four modules described above (crawler, page repository, indexers, indices) and their corresponding data files exist and operate independent of users and their queries. Spiders are constantly crawling the Web, bringing back new and updated pages to be indexed and stored. In Figure 1 these modules are circled and labeled as query-independent. Unlike the preceding modules, the query module is query-dependent and is initiated when a user enters a query, to which the search engine must respond in real-time.

The query module converts a user's natural language query into a language that the search system can understand (usually numbers), and consults the various indices in order to answer the query. For example, the query module consults the content index and its inverted file to find which pages use the query terms. These pages are called the relevant pages. Then the query module passes the set of relevant pages to the ranking module. The ranking module takes the set of relevant pages and ranks them according to some criterion. The outcome is an ordered list of web pages such the pages near the top of the list are most likely to be what the user desires. This ranking which carries valuable, discriminatory power is arrived at by combining two scores, the content score and the popularity score. Many rules are used to give each relevant page a relevancy or content score. The popularity score is determined from an analysis of the Web's hyperlink structure. The content score is combined with the popularity score to determine and overall score for each relevant page. The set of relevant pages resulting from the query module is then presented to the user in order of their overall scores.

2.2.3. Crawling, Indexing and Query Processing

Spiders are the building blocks of search engines. Decisions about the design of the crawler and the capabilities of its spiders affect the design of the modules, such as the indexing and query processing modules.

According to Manning *et al.* (2008), the crawler module contains a short software program that instructs robots or spiders on how and which pages to retrieve. The crawling module gives a spider a root set of URLs to visit, instructing it to start there and follow links on those pages to find new pages. Every crawling program must address several issues. For example, which pages should the spiders crawl? Some search engines focus on specialized search, and as a result, conduct specialized crawls, through only .gov page, or pages with images, or blog files, etc. According to Ashmanov and Ivanov (2010), Yandex crawls Russian Internet, therefore only the following domains are taken into index: .su, .ru, .am, .az, .by, .ge, .kg, .kz, .md, .ua, .uz. Yandex' robot also can visit other servers, if the Russian text there is found.

In addition it should be mentioned that Yandex has more than 16 different specialized crawls for different kind of data, but the most important one is the main indexing robot, whose function is to search and index information to maintain a base of the main search. There is a fast robot that assists the main one; its task is to index fresh, important up-to-date information promptly. Since the Web is dynamic, the information in last month's pages may contain different content from this month. Therefore, the crawling is a never-ending process.

In fact, back in 2000, Google was struggling about keeping the updated information. There were factors which prevented the crawl and were so onerous that after several attempts it looked as though Google would never build its next index. The web was growing at an amazing pace, with billions of more documents each year. The presence of search engines such as Google and Yandex actually accelerated the pace, offering an incentive to people as they discovered that even the uncommon piece of information could be accessed. Levy (2011) points out that Google was trying to contain such flow with more machines – cheap ones, thus increasing the chance of a breakdown. The updates would work for a while, then fail. In 2000 it took weeks before the Google's indices were updated. It is hard to overestimate the seriousness of this problem. One of the key elements of good search is freshness – making sure that the indices have recent results. Levy (2011) shows as an example September 11. 2001 terrorist attacks. If this problem occurred an year later after the attacks, the results for search query "World Trade Center" that November or December, would have found no

links to the event. Instead, the suggestions for a fine dining experience at Windows on the World, on the 107th floor of the no longer existent North Tower.

Each new or refreshed page that a spider brings back is sent to the indexing module, where software programs parse the page content and strip it of its valuable information, so that the only essential skeleton of the page is passed to the appropriate indices. Valuable information is contained in title, description, and anchor text as well as in bolded terms, terms in large font, and hyperlinks. One important index is the content index, which stores the textual information for each page in compressed form. An inverted file, which is used to store this compressed information, is similar to the index in the end of most of the non-literature books. Next to each term there is a list of all locations where the term appears. In the simplest case, the location is the page identifier. It is clear that an advantage of the inverted file is its use as a quick lookup table.

The simple inverted file, a main element in traditional information retrieval, does pose some challenges for web collections. This challenge is explained in Manning *et al.* (2008), because multilingual terms, phrases, and proper names are used, the number of terms, and thus the file size, is huge. Also, the number of web pages using popular broad terms such as "weather" or "sports" is large. Therefore, the number of page identifiers next to these terms is large and consumes storage.

Furthermore, page identifiers are usually not the only descriptors stored for each term. Other descriptors such as location of the term in the page (title, description, or body) and the appearance of the term (bolded, large font, or in anchor text) are stored next to each page identifier. Any number of descriptors can be used to aid the search engine in retrieving relevant documents. In addition, as pages change content, so must their compressed representation in the inverted file. Thus, an active area of research is the design of methods for efficiently updating indices. Lastly, the enormous inverted file must be stored on a distributed architecture, which means strategies for optimal partitioning must be designed.

Unlike the crawler and indexing modules of a search engine, the query module's operations depend on the user. The query module must process user queries in real-

time, and return results in milliseconds. In order to process a query this quickly, the query module accesses precomputed indices such as the content index and the structure index. When the user enters the query of two words, the query module consults the inverted lists both words and assumes the Boolean AND is used. The resulting set of relevant pages is the list of pages, which uses both words. Many traditional engines stop here, returning this list to the user. However, for broad queries on the vast web collection, this set of relevant pages can be huge, containing hundreds of thousands of pages. Therefore, rankings are placed on the pages in this set to make the list of retrieved pages more manageable. Consequently, the query modules passes its list to relevant pages to the ranking module, which creates the list of pages ordered from most relevant to least relevant. The ranking module accesses precomputed indices to create a ranking at query-time. Search engines combine content scores for relevant pages with popularity scores to generate an overall score for each page. Relevant pages are the sorted by their overall scores. How Google and Yandex compound their ranking is discussed in the next section.

3. Google and Yandex Algorithms

This section consists of three parts and describes the history of the two search engines, explains the mathematical aspects of their algorithms and shows the comparison of the results. In the first part the history of Google search engine and its algorithm is described. The second part shows the history of Yandex and description of MatrixNet algorithm. For the results comparison of Yandex and Google, ten competitive queries were selected in the Russian language and the analysis of the retrieved results is presented in the "Results and Analysis" section.

Google uses link analysis with the formula of PageRank, while many modern search engines on the Internet, such as Yandex, Yahoo and Bing, using models based machine learning methods. The latest ranking algorithm for machine learning, developed and applied in a search engine Yandex is called MatrixNet.

In November 2009 Yandex announced that it had significantly increased its search quality due to deployment of a new proprietary MatrixNet algorithm, a variant of a gradient boosting method which uses obivious decision trees.

In an interview in 2008, Peter Norvig, the director of research at Google, denied that their search engine exclusively relied on machine-learned ranking, pointed out that their search engine was not yet ready to entrust the final ranking to machine learning algorithms, citing the fact that the automatically generated models may behave unpredictably in the new classes of queries, which are not similar to the requests of the learning set, compared with the models created by human experts.

3.1. Google Search Engine

Google Search is a web search engine owned by Google Inc and is the most used search engine in the world. Google receives several hundred million queries each day through its various services. As mentioned above, Google has 91.7% of the overall search engine market share in the world. The order of search results on Google is

based, in part, on a priority rank called PageRank. The history of Google Inc. and the mathematical aspects of PageRank are shown in the following subsections.

3.1.1. History of Google Inc.

Sergey Brin and Larry Page had been collaborating on their Web search engine since 1995. By 1998, things were really starting to accelerate for these two scientists, a PhD students at Stanford university. Larry Page, at the time, was working on a PhD research project involving the mathematical properties of the link structure on the Internet. The research project, "BackRub", used an algorithm to follow the links in a web page and analyze all the connections. The PageRank algorithm, which was described by Bring and Page (1998), generated a popularity index for each web page based on the quantity and quality of incoming links. By 1998 Google's web crawler had indexed 60 million URLs and the company had been formally incorporated. In the next few years Google became the gateway to the Internet for the masses, as well as a traffic director that could make or break a company with its search rankings.

Larry Page understood that web links were like citations in a scholarly article. It was widely recognized that it is possible to identify which papers were really important without reading them – simply tally up how many other papers cited them in notes and bibliographies. Page believed that this principle could also work with web pages. But getting the right data would be difficult. Web pages made their outgoing links transparent: built into the code were easily identifiable markers for the destinations user could travel to with a mouse click from that page. But it was not obvious at all what linked to a page. To find that out, a database of links that connected to another page should be collected, then it would go backward. That is why Page called his system BackRub. "The early versions of hypertext had a tragic flaw: you couldn't follow links in the other direction," Page once told a reporter. "BackRub was about reversing that." (Levy 2011) A year later, their unique approach to link analysis was earning BackRub a growing reputation among those who had seen it.

Since Page was not a world-class programmer, he asked Scott Hassan for help. Page's program "had so many bugs in it, it wasn't funny", says Hassan. Part of the problem was that Page was using relevantly new computer language Java for his ambitious

project, and Java kept crashing. He decided to take his code and just rewrite it into another language. He wrote a program in Python – a more flexible language that was becoming popular for web-based programs – that would act as a spider, it would crawl the Web for data. The program would visit a page, find all the links, and put them into a queue. Then it would check to see if it had visited those link pages previously. If it had not, it put the link on a queue of future destinations to visit and repeated the process. Brin, the math professional, took on the huge task of crunching the mathematics that would make sense of the mess of links uncovered by their survey of the growing Web.

Steven Levy, a technology reporter from New York, in his book Levy (2011), describes the history of Goolge corporation and points out that in 1998 no one at the web search companies mentioned using links. The links were the reason that a research project running on a computer in a Stanford dorm room had become the top performer. Larry Page's PageRank was powerful because it cleverly analyzed those links and assigned a number to them, a metric on a scale of 1 to 10, which allowed user to see the page's prominence in comparison to every other page on the web.

"The idea behind PageRank was that you can estimate to importance of a web page by the web pages that link to it," Brin would say. "We actually developed a lot of math to solve that problem. Important pages tended to link to important pages. We convert the entire Web into a big equation with several hundred million variables, which are the PageRanks of all the web pages, and billions of terms, which are all the links."

The PageRank score would be combined with a number of more traditional information retrieval techniques, such as comparing the keyword to text on the page and determining relevance by examining factors such as frequency, font size, capitalization, and position of the keyword. Such factors are knows as signals, and they are critical to search quality. There are few crucial milliseconds in the process of a web search during which the engine interprets the keyword and then accesses the vast index, where all the text on billions of pages is stored and ordered just like an index of a book. At that point the engine needs some help to figure out how to rank those pages. So it looks for signals – traits that can help the engine figure out which pages will satisfy the query.

Though PageRank was was the combination of that algorithm with other signals that created the mind-blowing results. If the keyword matched the title of the web page or the domain name, that page would go higher in the rankings. For queries consisting of multiple words, documents containing all of the search query terms in close proximity would typically get the nod over those in which the phrase match was "not even close." Another powerful signal was the "anchor text" of links that led to the page. For instance, if a web page used the words "Bill Clinton" to link to the white House, "Bill Clinton" would be the anchor text. Because of the high values assigned to anchor text, a BackRub query for "Bill Clinton" would lead to www.whitehouse.gov as the top result because numerous web pages with high PageRanks used the president's name to link the White House site. When a user did a search, the right page would come up, even if the page did not include the actual words he/she was searching for. It was also something other search engines failed to do.

PageRank had one other powerful advantage. To search engines that relied on the traditional IR approach of analyzing content, the Web presented a challenge. There were millions and millions of pages, and as more and more were added, the performance of those systems inevitably degraded.

In September 1997, Page and Brin renamed BackRub to something they hoped would be suitable for a business. The Page's dorm roommate suggested the call it "googol." The word was a mathematical term referring to the number 1 followed by 100 zeros. Sometimes the word "googolplex" was used generically to refer to an insanely large number. "The name reflected the scale of what we were doing," Brin explained a few years later, "It actually became a better choice of name later on, because now we have billions of pages and images and groups and documents, and hundreds of millions of searches a day." Page misspelled the word, which was just a well since the Internet address for the correct spelling was already taken. Google.com was available. In 1998, Google was launched.

In a public presentation at the Seventh International World Wide Web conference in Brisbane, Australia, the paper "The anatomy of a large-scale hyper textual Web engine" made small ripples in the information science community that quickly turned

into waves. Since that evenful year, PageRank has emerged as a dominant link analysis model, partly due to its query-independence, its virtual immunity to spamming, and Google's huge business success.

While having a larger index of web pages accessed does not necessarily make one search engine better than another, it does mean the "bigger" search engine has a better opportunity to return a longer list of relevant results, especially for unusual queries. As a result, search engines are constantly battling for the title of "The World's Largest Index." Nowadays Google is officially the search engine in the world. Figure 2 shows the size of Google's index.

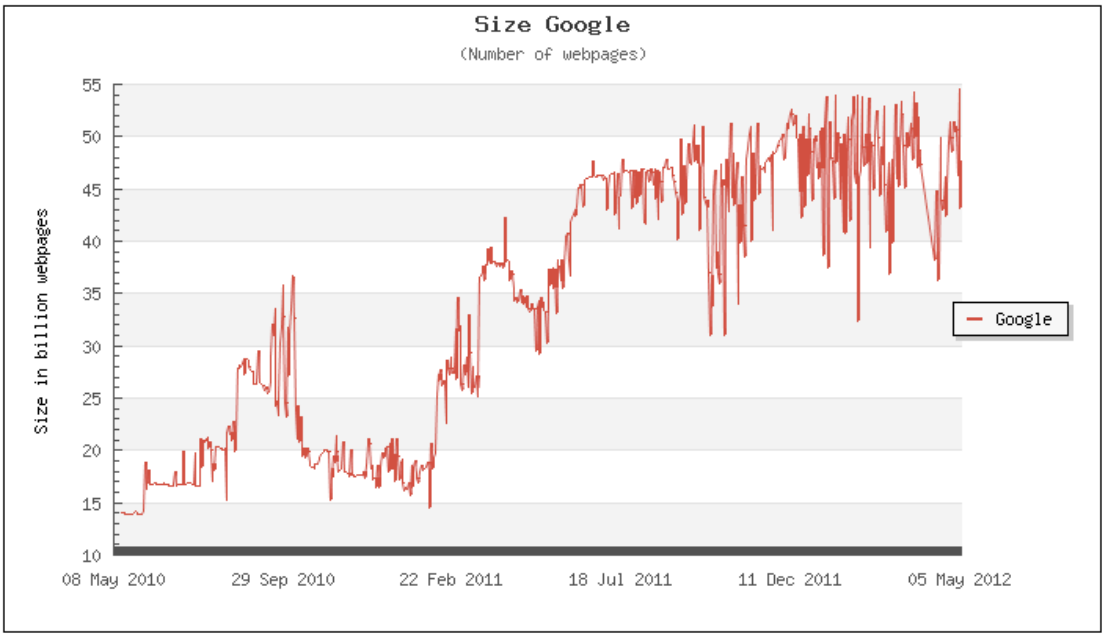

Figure 2. Estimated size of Google's index

,Google is the biggest search engine in the world and has over 50 billion pages in its index. The algorithm of a PageRank is described in the next subsection.

3.1.1. Mathematics of Google's PageRank

PageRank is Google's method of measuring the "importance" of a page's When all other factors such as Title tag and keywords are taken into account, Google uses PageRank to adjust results so that sites that are deemed more "important" will move up in the results page of a user's search accordingly.

The basic idea of a PageRank (Brin and Page, 1998) says that if a page links to another page, it is casting a vote, which indicates that the other page is good; if lots of pages link to a page, then it has more votes and its worth should be higher. PageRank is determined by the links pointing to a page. But if PageRank itself has an influence on the number of links to a page, it is influencing the quality of that page. The links are no longer based solely on human judgement. If a webmaster picks their outbound links by searching on Google, then there is a corresponding increase in a page's PageRank. This increase is not solely because it is a good page, but because its PageRank is already high.

According to Ridings and Shishigin (2002), with ranking factors other than PageRank, there is a score beyond which the slow down in the rate that any factor adds to this score is so insignificant that it is not worthwhile. This is the Non-PageRank Factor Threshold. If for the query the results are Page A and Page B, then Page A and B have scores for that query which are the total scores for all ranking factors (including PageRank). If page A's score is higher than page B's score, obviously, page A will be listed first. These are both below our hypothetical Non-PageRank Factor Threshold, thus without any change in PageRank, it is possible for page B to improve their optimisation to beat page A for this particular query. Generally, when querying Google, the group of pages in the search results will contain some pages that have a score above the Non-PageRank Factor Threshold, and some that do not.

To be competitive the site owners must raise their page's search engine ranking score beyond the Non-PageRank Factor Threshold. To fail to do so means that they can easily be beaten in the search results for query terms. The quickest way to approach the Non-PageRank Factor Threshold is through "on the page factors," however it is impossible to move above the Non-PageRank Factor Threshold without PageRank.

The keyword competition should be also taken into account. There are some queries where competition is so intense that sites must do everything possible to maximize their ranking score. In such situations it is impossible to rank highly through Non-PageRank factors alone. That is not to say that Non-PageRank factors are

notimportant. The final rank score is: Final Rank Score = (score for all Non-PageRank factors) x (actual PageRank score).

Improving either side of the equation can have a positive effect. However, because the Non-PageRank factors have a restricted maximum benefit, the actual PageRank score must be improved in order to compete successfully. Under really heavy competition – it holds true that sites cannot rank well unless their actual PageRank score is above a certain level. For queries that do not have heavy competition, this level is easy to achieve without even trying. However, where heavy competition exists, Non-PageRank factors are just as important until they reach the Non-PageRank factor threshold.

The Web's hyperlink structure forms a massive directed graph. The nodes in the graph represent web pages and the directed arcs or links represent the hyperlinks. Thus, hyperlinks into a page, which are called inlinks, point into nodes, while outlinks point out from nodes (Langville and Meyer, 2006). Figure 3 shows a tiny, artificial document collection consisting of six web pages.

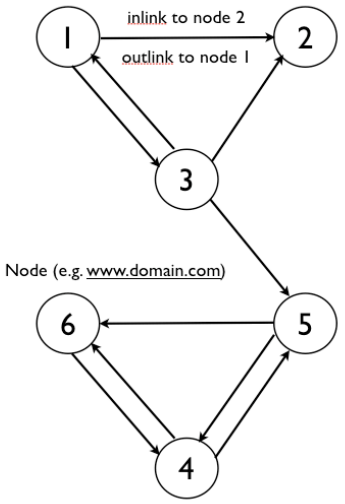

Figure 3. Directed graph representing the Web of six pages

Before 1998, the web graph was largely an untapped source of information. While researches like Kleinberg and Brin and Page recognized this graph's potential, most people wondered just what the web graph had to do with search engine results. The connection is understood by viewing a hyperlink as a recommendation. A hyperlink

from one homepage to another homepage is an endorsement of another page. Thus, a page with more recommendations (which are realized through inlinks) must be more important that a page with a few inlinks. However, similar to other recommendation systems such as bibliographic citations or letters of reference, the status of the recommender is also important.

Manning *et al.* (2008) points out, that academic citation literature has been applied to the Web, largely by counting citations or backlinks to a given page. This gives some approximation of a page's importance or quality. PageRank extends this idea by not counting links from all pages equally, and by normalizing by the number of links on a page.

Brin and Page, the inventors of PageRank, began with a simple summation equation, the roots of which actually derive from bibliometrics research, the analysis of the citation structure among academic papers. The PageRank of a page P_i, denoted $r(P_i)$ is the sum of the PageRanks of all pages pointing into P_i (Brin *et al.*, 1999).

$$r(P_i) = \sum_{P_j \in B_{P_i}} \frac{r(P_i)}{|P|} \quad (1)$$

In the equation (1), B_{P_i} is the set of pages pointing into P_i (backlinking to P_i in Brin and Page's words) and $|P_i|$ is the number of outlinks from P_i. The PageRank of inlinking pages $r(P_i)$ in equation (1) is tempered by the number of recommendations made by P_i, denoted $|P_i|$. The problem with equation (1) is that the $r(P_i)$ values, the PageRanks of pages inlinking to page P_i, are unknown. To sidestep this problem, Brin and Page used an iterative procedure. That is, they assumed that, in the beginning, all pages have equal PageRank (of say $1/n$, where n is the number of pages in Google's index of the Web). The rule is equation (1) is followed to compute $r(P_i)$ for each page P_i in the index and is successively applied, substituting the values of the previous iterate into $r(P_i)$. Let $r_{k+1}(P_i)$ be the PageRank of page P_i at iteration $k+1$. Then,

$$r_{k+1}(P_i) = \sum_{P_j \in B_{P_i}} \frac{r_k(P_i)}{|P|} \quad (2)$$

This process is initiated with $r_0(P_i) = 1/n$ for all pages P_i and repeated with the hope that the PageRank scores with eventually converge to some final stable values. Applying equation (2) to the tiny Web shown in Figure 3 gives the following values for the PageRanks after ten iterations.

These calcuations continue on until the value for each page no longer changes. In practice, Google probably does not wait for this convergence, but instead runs a number of iterations of the calculation which is likely to give them fairly accurate values. In Ridings (2002) the convergence is described as an important mathematical aspect of PageRank, which allows Google to provide unprecedented search quality at comparably low costs. Provided the dampening factor is less than one, then convergence will occur. Once the limiting values have been reached, Google no longer needs to expend processing power on calculating the PageRank.

The calculations of PageRank using equation (1) for the simple graph of six web pages in figure 3 are presented in Table 1. Table 1 shows the first 10 ietrations using equation (1) for the graph presented in Figure 3.

Table 1. Example of calculation of PageRank

Iteration	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6
0	0.150000	0.150000	0.150000	0.150000	0.150000	0.150000
1	0.192500	0.256250	0.213750	0.341250	0.256250	0.277500
2	0.210563	0.292375	0.231813	0.494781	0.355594	0.403938
3	0.215680	0.305169	0.239489	0.644474	0.425962	0.511409
4	0.217855	0.309519	0.241664	0.765732	0.491757	0.604935
5	0.218471	0.311060	0.242588	0.873192	0.543908	0.684433
6	0.218733	0.311584	0.242850	0.962929	0.589840	0.752267
7	0.218808	0.311769	0.242962	1.040109	0.628052	0.809927
8	0.218839	0.311832	0.242993	1.105360	0.660886	0.858969
9	0.218848	0.311855	0.243007	1.161000	0.688626	0.900654
10	0.218848	0.311855	0.243007	1.161000	0.688626	0.900654

Equations (1) and (2) compute PageRank one page at a time. Using matrices, the tedious Σ symbol can be replaced, and at each iteration, compute a PageRank vector, which uses a single 1 x n vector to hold the PageRank values for all pages in the index. In order to do this, an n x n matrix H and a 1 x n row vector π^T could be

used. The matrix H is a row normalized hyperlink matrix with $H_{ij} = 1/|P_i|$ if there is a link from node i to node j, and 0, otherwise. Although H has the same nonzero structure as the binary adjacency matrix for the graph, its nonzero elements are probabilities. Consider once again a tiny web graph of Figure 3. The H matrix for tiny web of Figure 3 is shown in matrix (3).

$$H = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (3)$$

The nonzero elements of row i correspond to the outlinking pages of page i, whereas the nonzero elements of column i correspond to the inlinking pages of page i. A row vector $\pi^{(k)T}$ is the PageRank vector at the k^{th} iteration. Using this matrix notation, equation (2) can be written compactly as shown in equation (4).

$$\pi^{(k+1)T} = \pi^{(k)T} H \quad (4)$$

Langville and Meyer (2006) points out, that matrix equation (4) yields some immediate observations.

1. Each iteration of equation (3) involves one vector-matrix multiplication, which generally requires $O(n^2)$ computation, where n is the size of the square matrix H.

2. H is a very sparse matrix (a large proportion of its elements are 0) because most web pages link to only a handful of other pages. Sparse matrices are welcome for several reasons. First, they require minimal storage, since sparse storage schemes, which store only the nonzero elements of the matrix and their location, exist. Second, vector-matrix multiplication involving a sparse matrix requires much less effort than the $O(n^2)$ dense computation. In fact, it requires $O(nnz(H))$ computation, where $nnz(H)$ is the number of nonzeros in H. Estimates show that the average web page has about 10 outlinks, which means that H has about $10n$ nonzeros, as opposed to the n^2 nonzeros in a completely dense matrix. This means that the vector-matrix multiplication of equation (3) reduces to $O(n)$ effort.

3. The iterative process of equation (2) is a simple linear stationary process of the form studied in most numerical analysis classes.

4. H looks a lot like a stochastic transition probability matrix for Markov chain. The dangling nodes of the network, those nodes with no outlinks, create 0 rows in the matrix. All the other rows, which correspond to the nondagling nodes, create stochastic rows. Thus, H is called substohastic.

These four observations are important to the development and execution of the PageRank model. Figure 4 illustrates a simple graph with rank sink.

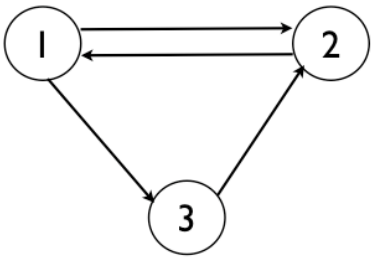

Figure 4. Simple graph with rank sink

Brin and Page originally started the iterative process with $\pi^{(0)T} = 1/ne^T$, where e^T is the row vector for all 1s. They immediately ran into several problems when using equation (4) with this initial vector. For example, there is the problem of rank sinks, those pages that accumulate more and more PageRank at each iteration, monopolizing the scores and refusing to share. In the simple example of Figure 4, the dangling node 3 is a rank sink. In the more complicated example of Figure 4, the cluster of nodes 4, 5, and 6 conspire to hoard PageRank. After just 13 iterations of equation (4), $\pi^{(13)T} = \begin{pmatrix} 0 & 0 & 0 & 2/3 & 1/3 & 1/5 \end{pmatrix}$. This conspiring can be malicious or inadvertent. The example with $\pi^{(13)T}$ also shows another problem caused by sinks. As nodes hoard PageRank, some nodes may be left with none. Thus, ranking nodes by their PageRank values if tough when a majority of the nodes are tied with PageRank 0. Figure 5 illustrates a simple graph with cycle.

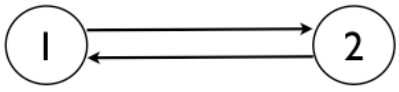

Figure 5. Simple graph with cycle

There is also problem of cycles. Consider the simplest case in Figure 5, page 1 only point to page 2 and vice versa, creating an infinite loop or cycle. Suppose iterative process of equation (4) is run with $\pi^{(13)T} = \begin{pmatrix} 1 & 0 \end{pmatrix}$. The iterates will not converge no matter how long the process is run. The iterates $\pi^{(k)T}$ turns indefinitely between $\begin{pmatrix} 1 & 0 \end{pmatrix}$ when k is even and $\begin{pmatrix} 0 & 1 \end{pmatrix}$ when k is odd.

The question that arises from all this is how, and when can, or will Google influence the results of the PageRank calculation. Google has shown that they can, and will modify the data on which PageRank is based. The primary example of this is what has become known as PageRank Zero (PR0). Basically speculation says that when Google wants to penalize a page, it is assigned a PageRank of zero. As PR is a multiplier, this will obviously always list PR0 pages as the very last entry in the search results. To stop its voting power, the second penalty must also be applied. This is the same penalty that is applied to link farms. Google has shown that they are capable of ignoring links they believe have been artificially created. The analysis of Google's search results is shown in the "Results and Conclusions" section.

3.2. Yandex Search Engine

Yandex is a Russian IT company, which operates the largest search engine in Russia with 60.4% of the market share in that country and also develops a number of Internet-based services and products. Yandex ranked as the 5th largest search engine worldwide with more than 3 billion searches, or 1.7% of global seacrh as of September 2011. Yandex is well-positioned within this large and rapidly expanding Internet market. It is currently the most visited web property overall in Russia, with more than 80 million Internet visitors in April 2012, making it more popular than Google, Microsoft and Facebook combined (34.6 million unduplicated visitors visited at least one of these sites). The web site is also present in Belarus, Kazakhstan, Ukraine and Turkey. The history of Yandex company and their latest ranking algorithm for machine learning, called MatrixNet, is described in the following subsections.

3.2.1. History of Yandex

The history of Yandex began in 1990, when a fresh graduate mathematitican and programmer Arkady Volozh start working on his first search technology at the company Arkadia. At that time, several key programmers developed a handful of search programs. These included The International Classifier of Patents and Search through the Bible, which took into account the Russian-language morphology. Both systems were running under DOS and allows to search by selecting words from a given dictionary, using the standard logical operators.

In 1993, Arkadia has become a subsidiary of CompTek, when the software technology has been significantly enhanced by cooperation with a team of experts of structural linguistics directed by Yuri Apresyan. In fact, the dictionary, which provides search and takes into account the Russian-language morphology, had the size of only 300 Kb that is entirely loaded into memory and worked rapidly. At this point the user could use any form of the queries in Russian language.

The word "Yandex" was invented by the company's two principal founders, Ilya Segalovich, Chief Technology Officer of Yandex, and Arkady Volozh, Yandex's Chief Executive Officer. At that time, Ilya was experimenting with different derivatives of words that described the essence of the technology. As a result, the team invented "yandex" – with "Ya" standing for the Russian "I". The full name originally stood for "Yet Another Index". Today the word Yandex has become synonymous with Internet search in Russian-speaking countries, just the same as Google in the rest of the world. Millions of people use Yandex each day for Internet search and other valuable services.

In early 1996 an algorithm for construction of hypothesis was developed. From that time, a morphological analysis was no longer tied to the dictionary – if the word was not excited in the dictionary, then the most similar words were found and thus the model of inflexion were build. In summer 1996 the CompTek and search engine developers have come to the conclusion that the development of the technology itself is more important and interesting than the creation of applications based on search. Market research has shown great possibilities of search technologies.

The official launch date of the yandex.ru search engine was September 23, 1997. On this date the system was publicly displayed at the Softool exhibition in Moscow. The Yandex search engine of 1997 took into account Russian language morphology and distance between words, and computed the relevance of a document using a complex algorithm. Within three years, Yandex became the largest search engine in Russia.

Nowadays Yandex is the largest search engine in Russian-speaking countries and is the largest Russian Internet company developing its world class proprietary technologies and creating a wide range of services for large audiences.

Yandex's innovative approach was manifested in 2009 when the company implemented a new method of machine learning which was called MatrixNet. This breakthrough technology takes into account thousands of search factors and their combinations. That has enabled Yandex to make search more precise as well as to refine the quality of search results for several classes of search queries.

3.2.1. Description of MatrixNet

Compared to Google, which built its technology based on links, Yandex from the beginning positioned itself as a search engine, based on the Russian language morphology. Therefore Yandex's approach is very different. Yandex's search engine processes more than 120,000,000 queries every day. Almost half of these queries are unique. To deal with this load of questions successfully, a search engine has to be able to make decisions based on the previous experience, that is, it has to learn. That is where machine learning is used.

Machine learning is essential not only in search technology. Speech or text recognition, for instance, is also impossible without a machine being able to learn. The term ”machine learning” coined in the 1950s, basically, means the effort to make a computer perform the tasks natural to human behavior, but difficult for breaking down into algorithmic patterns ”understandable” by machines. A machine that can learn is a machine that can make its own decisions based on input algorithms, empirical data and experience.

Decision making, however, is a human quality, which a machine cannot really master. What it can do, though, is learn to create and apply a rule that would help to decide whether a particular web page is a good answer to user's question or not.

This rule is based on properties of web pages and user's queries. Some of these properties, like the number of links leading to a particular page, are static – describing a web page, while others, like whether a web page has words matching a search query, how many and where on a page, are dynamic – describing both a web page and a search query. There are also properties specific only to search queries, such as geolocation. For a search engine, this means that to give a good answer to a user's question it has to factor in where this question has come from.

These quantifiable properties of web pages and search queries are called ranking factors. These factors are the key in performing exact searches and making the decision on which results are the most relevant. For a search engine to return relevant results for a user's query, it needs to consider a multitude of such factors. To approximate the users' expectations, a search engine requires sample user queries and matching results, which have already been considered satisfactory by the users. Assessors – people, who decide whether a particular web page offers a 'good' response to a certain search query – provide their evaluations. A number of search responses, together with corresponding queries, make up a learning sample for a search engine 'to learn to find' certain dependencies between these web pages and their properties. To represent real users' search patterns truthfully, a learning sample has to include all kinds of search queries in the same proportion as they occur in real life.

After a search engine has found dependencies between web pages in the learning sample and their properties, it can choose the best ranking formula for the search results it can deliver to a specific user's query and return the most relevant of them on top of all the rest.

Machine learning has been implemented in search technologies since the early 2000s. When a computer uses a large number of factors (properties of web pages and search queries) on a relatively small learning sample ("good" results as estimated by assessors), it begins to find dependencies that do not exist. For example, a learning

sample might accidentally include two different pages each having the same particular combination of factors, like they both are 2 KB, with purple background and feature text, which starts with "A". And, by sheer chance, these pages both happen to be relevant to the search query. A computer may deem this accidental combination of factors to be essential for a search result to be relevant to the search query. At the same time, all web pages offering really relevant and useful information about queries, but lacking this particular combination of factors, will be considered less important.

In 2009 Yandex launched MatrixNet, a new method of machine learning. A key feature of this method is its resistance to overfitting, which allows the Yandex search engine take into account a very large number of factors when it makes the decision about relevancy of search results. But now, the search system does not need more samples of search results to learn how to tell the ”good” from the ”not so good”. This safeguards the system from making mistakes by finding dependencies that do not exist.

MatrixNet allows to generate a very long and complex ranking formula, which considers a multitude of various factors and their combinations. Alternative machine learning methods either produce simpler formulas using a smaller number of factors or require a larger learning sample. MatrixNet builds a formula based on tens of thousands of factors, which significantly increases the relevance of search results.

Another important feature of MatrixNet is that allows customize a ranking formula for a specific class of search queries. Incidentally, tweaking the ranking algorithm for commercial searches will not undermine the quality of ranking for other types of queries. Commonly, any single turn of any single switch in a mechanism will result in global change in the whole machine. MatrixNet, however, allows to adjust specific parameters for specific classes of queries without causing a major overhaul of the whole system. In addition, MatrixNet can automatically choose sensitivity for specific ranges of ranking factors.

For each user's query, a search engine has to evaluate properties of millions of pages, assess their relevancy and rank them accordingly with the most relevant on top. Scanning each page in succession either would require a huge number of servers or

would take a lot of time – but a searcher cannot wait. MatrixNet solves this problem as it allows checking web pages for a very large number of ranking factors without increasing processing power.

Producing the final list of top results is based on all those lists of the most relevant pages produced by each server. These results are ranked using MatrixNet formula, which allows to consider a multitude of ranking factors and their combinations. Thus, the most relevant web sites find their way to the top of search results for the user to receive an answer to their question almost instantly.

The difficulty of the analysis of MatrixNet algorithm is that the formula has never been published, unlike Google's PageRank. But in the 'Internet Mathematics' contest, started by Yandex in 2004, in 2009 the real relevance tables that were used for learning ranking formula at Yandex, were distributed. The tables contained computed and normalized features of query-document pairs as well as relevance judgments made by Yandex assessors. The task of the 'Internet Mathematics 2009' contest was to obtain a document ranking formula using machine learning methods. As a result, a greedy algorithm was used for MatrixNet modification. The description of this modification using greedy algorithm is shown below.

A greedy algorithm is an algorithm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum. Greedy algorithms performed well in solving the practical problems of machine learning. This algorithm is used to solve the problem of improving the ranking quality and sorting the most relevant documents to the particular query in MatrixNet.

In the greedy algorithm the functions of the relevance of document d with respect to query q as follows:

$$fr(q,d) = a_1 h_1(q,d) + a_2 h_2(q,d) + \ldots + a_n h_n(q,d) \quad (5)$$

According to Gulin and Karpovich (2009), MatrixNet is using the method of weak leaners algorithms, which in equation (5) are shown as $h_k(q,d)$. It should be

mentioned that amount of functions $h_k(q,d)$ is sufficiently large, tens of thousands and coefficients a_k are small quantities. According to Zyabrev and Pozharkov (2010) it is possible that a_k could be larger quantities, but in practice they are not. Coefficients a_k may be less than zero, which means that some of the terms give a negative contribution to relevancy. The learning is based on the estimated pair $(query,document)$, whose number is likely to have more than 5 million.

Gulin and Karpovich (2009) described several metrics, which are commonly used to evaluate and compare the quality of ranking algorithms on a sample of assessors estimate. Often ranking model parameters tend to adjust in order to maximize the value of one of these metrics. Examples of this metrics are: GDN, nGDN and MAP.

The main goal is to rank documents according to their quality of conformance to the search query. Prerequisites includes set of search queries $Q=\{q_1,...,q_n\}$, set of documents corresponding to each query $q\in Q$, $q\to\{d_1,d_2,...\}$ and relevance judgments for each pari $(query,document)$ in the form of numbers from 0 to 1 - $rel(q,d)\in[0,1]$.

Evaluation mark for ranking will be an average value of evaluation measure over the set of search queries Q:

$$\frac{\sum_{q\in Q} EvMeas(ranking\_for\_query\_q)}{n} \quad (6)$$

An example of evaluation measure $EvMeas$: Precision-10 – percent of documents with relevance judgments greater than 0 in top-10 and MAP – mean average precision:

$$MAP(ranking\_for\_query\_q)=\frac{1}{k}\sum_{i=1}^{k}\frac{i}{n_r(i)} \quad (7)$$

In equation (7) k is the number of documents with the positive relevance judgments corresponding to the query q, $n_r(i)$ is the position of i-th document with relevance judgment greater than 0.

The main quality metrics is Discounted Cumulative Gain (DCG) averaged over all queries. The following initial formula for DCG was used:

$$DCG(ranking\_for\_query\_q) = \sum_{j=1}^{N_q} \frac{rel_j}{\log_2 j + 1} \quad (8)$$

In the equation (8) N_q is a total number of documents in ranked list, rel_j is relevance judgment for document on position j.

Normalized DCG (nDCG) is calculated with the following formula:

$$nDCG(...) = \frac{DCG(ranking\_for\_query\_q)}{DCG(ideal\_ranking\_for\_query\_q)} \quad (9)$$

Each pair $(query, document)$ is described by the vector of features.

$$(q,d) \rightarrow (f_1(q,d), f_2(q,d), ...) \quad (10)$$

Search ranking is the sorting by the value of relevance function. Relevance function is a combination of features:

$$fr(q,d) = 3.14 \cdot \log_7(f_9(q,d) + e^{f_{66}(q,d)} + ... \quad (11)$$

The main question of optimisation is how to get a good relevance function. Based on the learning set of examples P_i - set of pairs (q,d) and with relevance judgments $rel(q,d)$ and use learning to rank methods to obtain fr.

According to Gulin and Karpovich (2009), solve direct optimisation problem:

$$\arg\max_{fr \in F} = \frac{\sum_{q \in Q_l} EvMeas(ranking\_for\_query\_with\_fr)}{n} \quad (12)$$

In equation (12) F is the set of possible ranking functions, Q_l is the set of different queries in learning set P_l. In this case the difficulty in solving is that most of evaluation measures are non-continuous functions.

Simplify optimisation task to regression problem and minimize sum of loss functions:

$$\arg\min_{fr \in F} L_t(fr) = \frac{\sum_{(q,d) \in P_i} L(fr(q,d), rel(q,d))}{n} \quad (13)$$

In the equation (13) $L(fr(q,d), rel(q,d))$ is the loss function, F is the set of possible ranking functions.

In order to solve the regression problem in equation (13), the relevance function needs to be found in the following form:

$$fr(q,d) = \sum_{k=1}^{M} \alpha_k h_k(q,d) \quad (14)$$

Relevance function will be linear combination of functions $h_k(q,d)$, where terms $h_k(q,d)$ belong to simple weak learning family. The final function of relevance needs to be constructed by iterations. On each iteration the term $\alpha_k h_k(q,d)$ needs to be added to the relevance function:

$$fr_k(q,d) = fr_{k-1}(q,d) + \alpha_k h_k(q,d) \quad (15)$$

The value of parameter α_k and weak learner $h_k(q,d)$ can be a solution of natural optimisation task:

$$\arg\min_{\alpha, h(q,d)} \frac{\sum_{(q,d) \in P_i} L(fr_{k-1}(q,d) + \alpha h(q,d), rel(q,d))}{n} \quad (16)$$

This problem can be solved directly for quadratic loss function and simple classes H, but it can be very difficult to solve for other loss functions. The weak learners H is the set of decition-tree functions is shown in Figure 6.

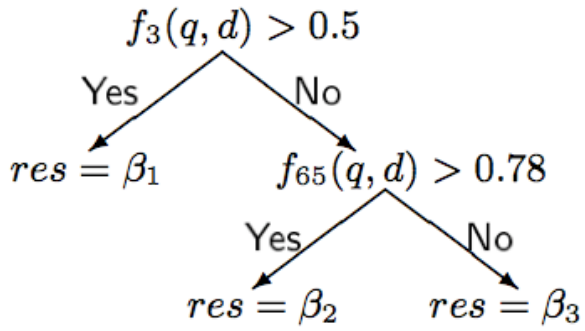

Figure 6. Example of the decision tree

The function splits feature space on 3 regions by conditions in the form $f_j(q,d) > \alpha$ (f_j - split feature, α- split bound). It has a constant value for feature vectors in one region.

In this model, decision trees are used by dividing the space into six areas. The optimisation problem to be solved:

$$\arg\min_{h(q,d)\in H} \sum_{(q,d)\in P_l} (g_{(q,d)} - \beta h(q,d))^2 \quad (17)$$

If tree-structure of weak learner $h(q,d)$ is known, then the split conditions and regions also do. Then the region constant values should be found. Optimisation problem reduces to ordinary regression problem:

$$\arg\min_{h(q,d)\in H} \sum_{(q,d)\in P_l} (g_{(q,d)} - \beta\beta_{ind(q,d)})^2 \quad (18)$$

In equation (18) $ind(q,d)$ is the number of region, which contains features vector for pair $ind(q,d)$ ($ind(q,d) \in \{1,...,6\}$).

Weak learner selection in form of tree structure includes $bestTree$, which is a constant funtion (1-region tree) and greedy split of $bestTree$, which is shown in Figure 7.

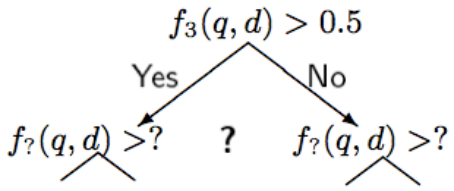

Figure 7. Greedy split for 1-region tree

In MatrixNet weak learners set is full decision trees with depth and regions: a constant number of layers and the same split conditions for one layer as shown in Figure 8.

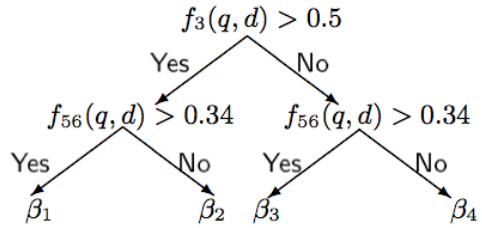

Figure 8. The structure of the split conditions for one layer

This can also be applied to the analysis of algorithm factors, which are important for Search Engine Optimisation, to maximize the relevance function of the site. Zyabrev and Pozharkov (2010) points out, that the function is a piecewise constant function, which together with the limited depth of the tree can provide the following effect. Documents with slightly different values for the algorithm can be perceived as equivalent. At the same time a document can be slightly more relevant with respect to another, but the value of their relevance will be the same. On the other hand, documents with slightly different properties can have very different values of the functions. Although in general such jumps are smoothed out by a large number of function of $h_k(q,d)$ terms. It is partly confirmed by Zyabrev and Pozharkov (2010) that the indirect dependence of the functions on the properties of the document, which makes the behavior of relevance depending on the function $f_i(q,d)$ is difficult predictable in terms of external analysis.

The conclusion can be drawn, i.e. that the algorithm has quite flexible features, and has no structural constraints allows additional learning if necessary at minimal resource cost, simply by adding new data in the learning set and not changing an existing structure.

In practice, when MatrixNet was launched, Yandex results had poor quality. This problem was widely discussed by SEO experts at forums and seminars, and the main conclusion was that the MatrixNet, in practice, promotes the doorways growing and spam, and as a quality of the result is measured by assessors. Assessors measure quality based on the quality of the information, relevancy for particular query, page speed, usability and user-friendly design.

The assessors are given random SERPs from real search queries and rate the documents according to the scale: Vital (the best answer possible; usually official sites of organizations), Useful (very good and informative answer), Relevant ”+” (answers the question), Relevant ”-” (partly relevant, but does not answer the question fully), Irrelevant (does not answer the question). The assessors are given tasks like to evaluate a specific document, evaluate search results for a particular query, evaluate site snippet in a SERP, compare two documents and pick the most relevant to a specific search query, compare two search results pages and pick the best. Mainly the human assessments are used on top 10 results, but can be also applied to further positions, depending on Yandex’s goal. There are two main ways the human assessments are used at Yandex: for evaluating quality of search results and for “teaching” MatrixNet.

Yandex has many different metrics to measure the quality of search results, one of them being pFound. pFound measures probability of that the user will find the answer he / she is looking for, based on hypotheses that a) the user will browse the SERP from the top to the bottom and b) the user will click on every document until he / she finds the answer or leaves the SERP without the answer. Similar analysis of the quality of the results is presented in the “Results and Analysis” section.

4. Search Engine Optimisation

This section explains Search Engine Optimisation (SEO) and describes the SEO factors, which influence the search results. Since Google and Yandex have different algorithms, SEO factors also various.

Many online companies have become aware of the importance of ranking well in the search engines. A user behaviour study by iProspect (2006) reveals that 62% of search engine users click only on results that appear on the first search engine results page (SERP) and less than 10% of users click on the results that appear after the third page.

In order to place well in SERPs companies have begun to use search engine optimisation techniques. That is they manipulate the site's content and meta tags, as well as attempt to attract incoming links from other sites. However, certain SEO techniques directly violate the guidelines published by the search engines. While the specific guidelines vary a bit, they can all be summed up as: 'show the same content to search engines as you show to users.'

Search Engine Optimisation is the active practice of optimising a web site by improving internal and external aspects in order to increase the traffic the site receives from search engines. Firms that practice SEO can vary; some have a highly specialised focus, while others take a broader and more general approach. According to Dover (2011), optimising a web site for search engines can require looking at so many unique elements that many practitioners of SEO consider themselves to be in the broad field of web site optimisation. Search engines have been known to occasionally modify their algorithms and, as a result, turn the SERPs upside down. For example this includes Yandex's new algorithm MatrixNet launched in 2009.

As mentioned above, there are many factors influencing the rankings of the web site. Such factors as page title, quality of content, meta description, inbound links and many other are the basic factors and not all described in this work in details. But it is important to mention most important of them in order to analyze and make conclusions. The most important SEO factors are listed in Table 2.

Table 2. The description of basics SEO factors

Factor	Description
Title tag	The title element of a web page is meant to be an accurate and concise description of a page's content. This element creates value in three specific areas: browser, search results, external results and is critical to both user experience and search engine optimisation.
Meta description	The meta description tag serves the function of advertising copy, drawing readers to a web site from the results and thus, is an extremely important part of search marketing. Crafting a readable, compelling description using important keywords can draw a much higher click-through rate of searchers to the given web page.
On-Page factors	Content pages are the main properties of web sites and are almost always the reason visitors come to a site. Ideal content pages should be very specific to a given topic (usually a product or an object) and be relevant. The purpose of the given web page should be directly stated in all of the following areas: title tag, domain name, content of page and image alt texts.
External links	External Links are hyperlinks that point at any domain other than the domain the link exists on. Many top SEOs believe that getting external links is the single most important objective for attaining high rankings. This stems from the idea that external links are one of the hardest metric to manipulate and thus one of the best ways for search engines to determine the popularity of a given web page.
Internal links	Internal Links are hyperlinks that point at the same domain as the domain that the link exists on. Internal Links are most useful for establishing site architecture and spreading link juice.
Anchor text	Anchor text is the visible characters and words that hyperlink display when linking to another document or location on the Web. As search engines have matured, they have started identifying more metrics for determining rankings. One metric that stood out among the rest was link relevancy. Link relevancy is determined by both the content of the source page and the content of the anchor text.
Domain	Domain names are the human readable Internet addresses of web sites. The domain name itself is a key ranking factor that the engines consider when calculating ranking order. Also having relevant keywords in a domain name is beneficial because the domain name is the text that other Internet users will use as anchor text when linking. Since keywords in anchor text are an important ranking factor, having these keywords in a domain name has a significantly positive impact on ranking.
URL	URL, or Uniform Resource Locator, is a subset of the Uniform Resource Identifier (URI) that specifies where an identified resource is available and the mechanism for retrieving it. URLs describe a site and page to visitors and search engines. Thus, keeping them relevant, compelling and accurate are key to ranking well.
Redirection	Redirection is process of forwarding one URL to a different URL. There are three main kinds of redirects online; 301, 302 and meta refresh. 301 redirect states for 'Moved Permanently' and it is recommended for SEO.

SEOmoz, a software developers company, but also one of the most respected SEO communities, has published the survey "Search Ranking Factors 2011", which is correlation-based analysis - comparing the aggregated opinions of 132 SEOs around the world with correlation data from over 10,000 results in Google. Rather than showing the old 0-5 importance scale along with the "degree of consensus" calculated on standard deviation, they have tried this new format, which highlights relative importance of metrics in a single section based on the aggregation of the voters' ordering. Those elements that are very high on the "influence value" tended to be consistently rated as more important that features below them. The degree of difference between influence values shows, on the 100-point scale, how much the average of the votes differed. The averages of voters' opinions are illustrated, which are most important ranking factors by SEOmoz are shown in Figure 9.

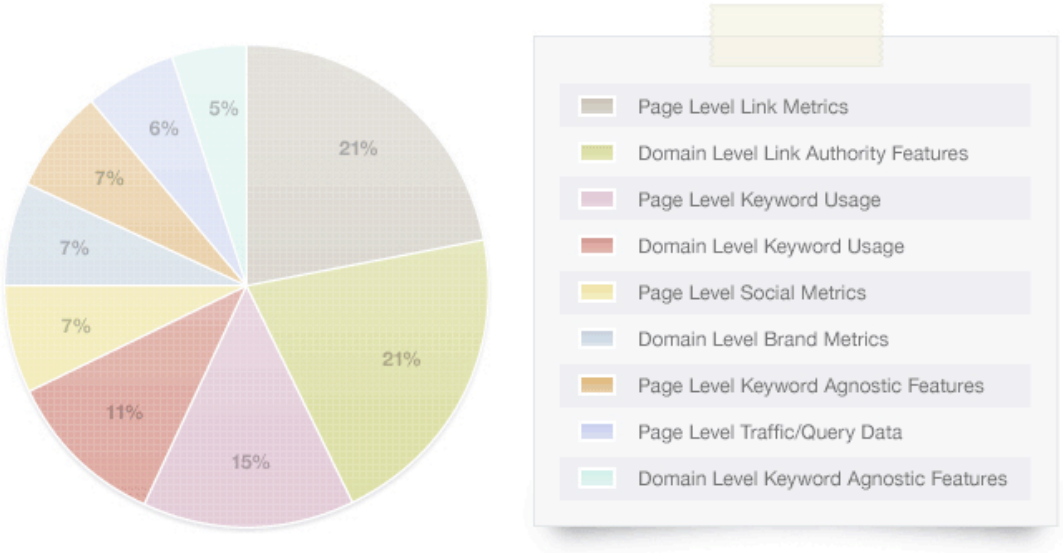

Figure 9. Search Ranking Factors 2011 by SEOmoz

As shown in Figure 9, the page level metrics as well as domain level link authority features are the most important factors of Search Engine Optimisation process. Page level metrics are the on-page factors, most of them were presented in Table 2. These on-page factors are similar to Google and Yandex. Domain level link authority features is another factor, but it can various for selected search engines, since Yandex has machine learning algorithm and the link authority is calculated in a different way.

Yandex also has its own "PageRank" and called thematic citation index (TCI). It is determined by the quantity and quality of inbound links to the web site, but TIC also takes into account the thematic proximity of the linked site. The TCI does not take into account links from forums, bulletin boards, sites on a free hosting and links from those web sites, which Yandex do not index.

The information in the title (tag <title>) Yandex maps in search results. Words that are in the tag <title>, carry more weight than others. In addition to the above methods, the relevance of the words affects the frequency of its use in the headers (<h1>, <h2> ...), the attribute alt, in tooltips (tag <acronym>) and the percentage of occurrence of the word in the document, i.e. how often the word is used on the page. But it is necessary to preserve the meaning of the document, or Yandex may find the page as a spam.

Whatever the search engine, Yandex or Google, no matter what they do with their filtering algorithms, it is still the main criteria for assessing the relevance of the resource with respect to a particular query is the presence of high-quality text content. For SEO a priority in promoting the resource is, above all, the optimisation of site content and its internal link structure and ease of navigation for the user directly, rather than a direct optimisation for a specific search engine.

5. Results and Analysis

This section presents the comparison analysis of the retrieved results from Yandex and Google. Ten search queries in the Russian language were used to test the search engines; the precisions of the search results retrieved were compared amongst the search engines. The first ten documents on each retrieval output were evaluated as being ”relevant” or ”non-relevnat” for evaluation of the seach engine’s precision.

5.1. Test Queries

Ten search queries were designed for use on both search engines. These queries were designed to test various features that each search engine claims to have, as well as to represent different levels of searching complexity. The queries also were designed to fall within the domain of Travel in Finland for the purpose of familiarity, such that investigators could judge the search results for relevance. The queries were designed based on their popularity, number of monthly searches in Yandex and Google. The data for monthly searches were taken from Yandex Keyword Stats tool and
Google Keyword tool, which show the number of times the particular query was searched per month. Ten queries were classified into four groups as follows:

Group A. Local searches:

1. Visa to Finland St. Petersburg (Виза в Финляндию в Петербурге)
2. Tours to Finland from St. Petersburg (Туры в Финляндию из Петербурга)

Group B. Descriptive searches:

3. Visit to Finland (Поездка в Финляндию)
4. Holidays in Finland (Отдых в Финлянди)
5. Finland for weekend (Финляндия на выходные)

Group C. Commercial searches:

6. Hotels in Finland (Отели в Финляндии)
7. Hotels in Finland (Гостиницы в Финляндии)
8. Hotels in Helsinki (Отели в Хельсинки)

Group D. Informational searches:

9. Map of Finland (Карта Финляндии)
10. Shopping in Finland (Магазины Финляндии)

The popularity of the selected queries in both Yandex and Google is shown in Table 3.

Table 3. Selected queries for the test and their popularity

Query - Group	Monthly searches in Yandex	Monthly searches in Google	Monthly searches summary
Query 1	34277	22200	56477
Query 2	23119	27100	50219
Total: Group A	57396	49300	106696
Query 3	9925	4400	14325
Query 4	6445	9900	16345
Query 5	1956	1600	3556
Total: Group B	18326	15900	34226
Query 6	5473	3600	9073
Query 7	781	1900	2681
Query 8	3940	2400	6340
Total: Group C	10194	7900	18094
Query 9	21042	12100	33142
Query 10	22792	9900	32692
Total: Group D	44834	22000	66834

In Table 3 the queries are sorted by groups, the popularity in Yandex and Google is given. SEO companies for the semantic optimisation, estimating the number of visitors to their web site and strategy planning, normally use these data. For instance, for local searches, Group A, the Query 2 has more searches in Google compare to Yandex, whereas query 1 has more searches in Yandex. There is an opinion that Yandex has better quality for commercial queries, while Google provides better information for general searches. There are no specific studies having been done on this subject, but some SEO professionals claim this based on their experience with their clients' projects.

For each query, only the first ten results were evaluated. For most users, the first ten retrieved results are the most important, i.e. almost all users hope that the first ten search results will provide what they are looking for and if this is not the case, they become frustrated and since usually search engines display results in descending order of relevance, it is believed that this metholodogy did not critically affect the validity of the results.

5.2. Test Enviroment

Mozilla Firefox version 12.0 was chosen as the Web browser for the study because it is compatible with all the search engines selected and is the most widely used browser locally. The current location in both search engines was set as St. Petersburg, this parameter is important for the local searches results. Two computers with different configurations: Mac OS X version 10.5.8 with the 2 GHz Inter Core 2 Duo processor and 2 GB 1067 MHz DDR3 memory and an Acer computer with an Intel Celeron M Processor 440, 1.86 GHz with 80 GB hard disk and 1 GB RAM were used. One computer was used for the entire experiment, which was repeated for validity on the second computer, i.e. each query was run twice. The results shown are those obtained from the Mac computer. The results from the repeated exercise are not presented because they were comparable and do not add to the outcomes of the study.

Ideally, each query should be executed on all search engines at the same time, so that if a relevant page is added, none should have an advantage of being able to index the new page over the other. For this study, that was not practically possible and so each query was searched on the search engines within a few hours of each other the same day. Those results returning an error "404" (i.e. path not found) or "603" (i.e. server not responding) were noted in order to be returned to. Returned visits were made at different times of the day to allow for the possibility that the site might have a regular down time for maintenance.

5.3. Response Time

The response time was calculated as the period between entering a search query and the retrieval of the first search results and was measured by stopwatch. One query from each group was selected to assess response time. The queries selected were: Query 1 (Group A), Query 4 (Group B), Query 8 (Group C) and Query 10 (Group D). The average response times for each search engine and for each selected query were then calculated.

The individual and mean response times, measured in seconds, for each search engine and for each query during off-peak and peak hours are shown in Tables 4 and 5, respectively.

Table 4. Response time during off-peak hours

Query	Yandex	Google
1	3	2
4	2	4
8	4	2
10	6	3
Mean	3.75	2.75

Table 5. Response time during peak hours

Query	Yandex	Google
1	18	12
4	25	9
8	18	23
10	17	18
Mean	20	16

The mean response times for both search engines were within the range of 2 s – 6 s during off-peak hours. During peak hours, mean response time increased to 9 s and went as high as 25 s.

5.4. Precision

For this study, precision was defined as the relevance of a search result to a search query and was determined separately for the first ten search results. The content of each retrieved result was checked to determine whether it satisfied the expected result. A precision score was calculated based on the number of the results within the first ten retrieved deemed to be relevant. The precision score of each result was based on the quality of the information, relevancy for particular query, page speed, usability and user-friendly design. In order to assess the overall performance of each search engine, not only the average precision score for each query was computed, but also the average precision score was calculated, based on all ten queries, for each search engine. The full list of all retrieved results and their precision score is listed in Appendix 1. The precision scores for Group A are listed in Table 6.

Table 6. Precision scores for Group A

Position for Q1	Yandex	Google
1	9	9
2	10	10
3	9	9
4	10	9
5	10	9
6	9	10
7	9	8
8	9	7
9	10	10
10	10	9
Mean Q1	9.5	9.0
Position for Q2	Yandex	Google
1	10	6
2	10	10
3	9	10
4	7	9
5	10	8
6	6	7
7	10	6
8	10	5
9	10	10
10	10	6
Mean Q2	9.2	7.7
Mean Group A	9.4	8.4

For the local searches (Group A) the mean precision scores in Yandex ranged from 9.2 to 9.5, in Google from 7.7 to 9.0. In average, Yandex retrieved better quality results. For the Query 1 the precision is approximately similar in both Yandex and Google, but for the Query 2 the precision differs to 1.5 scores. Query 2 (Tours to Finland from St. Petersburg) in the Russian language in some context might be considered as a synonym for the Query 3 (Visit to Finland). In Yandex most of the results with the precision score 10 were the sites of the companies, which offer trips or transfer to Finland from St. Petersburg. 40% of the retrieved results for Group A included the same web sites. This test shows that Yandex recognizes synonyms in the Russian language better than Google and that for local searches Yandex retrieved better results.

For the descriptive searches, which do not include the location name (Group B), the precision scores in Yandex ranged from 6.1 to 7.4, in Google from 6.6 to 8.8. In total,

Google retrieved better quality results with the precision score of 8.0. 13% of the sites, retrieved for the Group B were the same, which means that Google and Yandex have more diffirences for the descriptive searches compare to local searches. The precision scores for Group B are listed in Table 7.

Table 7. Precision scores for Group B

Position for Q3	Yandex	Google
1	10	10
2	10	10
3	9	9
4	6	9
5	4	10
6	9	8
7	5	8
8	3	6
9	4	10
10	5	8
Mean Q3	6.5	8.8
Position for Q4	Yandex	Google
1	10	10
2	10	10
3	6	9
4	6	7
5	5	6
6	3	9
7	4	10
8	5	10
9	5	10
10	7	7
Mean Q4	6.1	8.8
Position for Q5	Yandex	Google
1	9	10
2	7	5
3	8	6
4	7	7
5	10	4
6	7	9
7	9	5
8	7	4
9	4	7
10	6	9
Mean Q5	7.4	6.6
Mean Group B	6.6	8.0

While the precision score for Queries 3 and 4 in Google is higher when compared to Yandex, for the Query 5 the situation is the opposite. Here again the conclusion about synonyms can be drawn, because for the Query 5 (Finland for weekend) the results for

the Query 3 (Visit to Finland) can be relevant. Google retrieved better results for the Query 3, but it did not recognize the similarity with the Query 5. The conclusion that Google retrieves more relevant results for the descriptive searches, can be drawn, but this test confirms the statement mentioned above that Yandex recognizes synonyms in the Russian language better than Google.

For the commercial searches (Group C) Yandex and Google have similar precision scores of 7.8. The precision scores for Group C are shown in the Table 8.

Table 8. Precision scores for Group C

Position for Q6	Yandex	Google
1	6	8
2	8	6
3	7	6
4	9	10
5	10	10
6	10	7
7	7	4
8	6	8
9	5	7
10	3	5
Mean Q6	7.1	7.1
Position for Q7	Yandex	Google
1	10	7
2	7	5
3	6	7
4	8	9
5	7	6
6	8	7
7	6	8
8	5	6
9	7	6
10	9	5
Mean Q7	7.3	6.6
Position for Q8	Yandex	Google
1	10	10
2	10	10
3	10	10
4	10	9
5	7	9
6	10	10
7	10	10
8	9	10
9	8	9
10	8	10
Mean Q8	9.2	9.6
Mean Group C	7.8	7.8

In Group C Queries 6 and 7 are synonyms in the Russian language, which both can be translated into English as "Hotels in Finland", but Yandex has a better precision score for the Query 7, which proves that synonyms can be recognized by Yandex better than by Google. 60% of retrieved results for the Group C were similar in Yandex and in Google.

For informational searches (Group D) the test shows that Yandex retrieve more relevant results with the precision score of 9.0. The precision scores for Group D are tabulated in Table 9.

Table 9. Precision scores for Group D

Position for Q9	Yandex	Google
1	9	8
2	8	10
3	10	9
4	9	6
5	7	10
6	10	7
7	10	5
8	8	10
9	10	8
10	10	4
Mean Q9	10.0	7.7
Position for Q10	Yandex	Google
1	10	10
2	9	9
3	8	8
4	7	7
5	10	10
6	8	10
7	6	10
8	6	7
9	7	6
10	9	6
Mean Q10	8.0	8.3
Mean Group D	9.0	8.0

For informational searches, precision scores in Yandex ranged from 8 to 10, which makes the highest mean precision score of 9 within the test. 20% of the sites, retrieved for the Group D were the same, which means that Google and Yandex have more diffirences for the informational searches comparedto commercial searches.

In order to finalize the results of Yandex and Google performance, the overall precision scores should be compared. The comparison is shown in Table 10.

Table 10. Mean precision scores for each query and groups

Query	Yandex	Google
1	9.5	9.0
2	9.2	7.7
3	9.4	8.8
4	6.5	8.8
5	6.1	6.6
6	7.4	8.1
7	7.1	7.1
8	7.3	6.6
9	9.2	8.3
10	10.0	8.0
Group	Yandex	Google
A	9.4	8.4
B	6.6	8.1
C	7.8	7.8
D	9.0	8.0
Mean	8.2	8.0

The precision score for each query on each search engine, as shown in Table 10, ranges from 6.1 to 10.0 for different queries. Although the ranking of the precision scores varied amongst Yandex and Google depending on the query, Yandex obtained a slightly higher mean precision score of 8.2 while Google obtained the score of 8.0. The average similarity of the results is 38%, which means that Yandex and Google do not retrieve the same sites.

These results show that Yandex retrieves more relevant information for local searches and informational searches, while Google, on average, shows better results for descriptive and commercial searches. The test also shows that Yandex recognises synonyms in the Russian language better than Google.

Discussions and Conclusions

The thesis presents the comparison of search engines Yandex and Google using the Russian language. The effort was divided in two main tasks, the understanding of both algorithms, their formulas and approach, and the comparison of the relevancy of the results retrieved. In the first part ofthe study, the basics of information retrieval and main principles of search engines were presented. Then the search techniques of Google's PageRank and Yandex' MatrixNet were discussed and some main factors of Search Engine Optimisation were presented. Finally, the comparison of the quality of the retrieved search results was reported.

The main objective of the analysis of search engine performance in the Russian language was fully achieved. The analysis shows that machine learning algorithms retrieve better results for local services and informational searches, while the importance of the page, based on a link analysis leads to better results for commercial searches. For both response time and precision, Yandex proved to be a better performer than Google. Also the fact that machine learning algorithm based on the morphology of the Russian language, used by Yandex, can recognise synonyms in Russian language much better than Google. On the literature no similar analysis of both comparison of the search engine algorithms and the precision of the results were found. Other search engines, such Yahoo, Bing, Baidu and other are not addressing to the same topic, and cannot be compared to this project.

Google's PageRank measures the importance of the page based on the link analysis and this formula works for all types of searches. Yandex uses machine learning algorithms for their MatrixNet, which measures different factors for different type of queries. The conclusion can be drawn that Yandex, unlike Google, pays great attention to regional sites and gives a more influential role to the geographical dependence. In the analysis of the retrieved results most of the web sites in Yandex were located in St.Petersburg. Yandex's ranking of the sites includes the territorial basis. But on the other hand, this feature of Yandex leads to the fact that new sites that promote services on the territory of the former Soviet Union get relevant result of the search in the "foreign" geographic area.

The study also shows that there are also differences between search the engines Google and Yandex in terms of perception of the value of content. Google does not pay almost any attention to what is essentially promoted on the site in terms of enhanced citation index. The main attention is paid to both text and graphical content online. That might be the reason why the retrieved results in this study have better quality for the commercial searches in Google. Therefore it is comparibly easy to promote a web site in Google by increasing the amount and quality of the content, and incoming links. Yandex is inherent to have a different method of perception of the value of content. The textual content of the site is one measure of the value of the site for Yandex, but not the most important criterion.

For the site owners and companies the outcome of the study can be summarized as Yandex retrieves more relevant information for local searches and informational searches, while Google, on average, shows better results for descriptive and commercial searches.

As future work, the most obvious tasks would be deeper analysis of Search Engine Optimisation factors and differences of the techniques for site optimisation, testing different methodology and strategy of Search Engine Optimisation on similar sites and comparing the importance of the factors. Futhermore, a similar study could be done for other local search engines which have bigger local market shares than Google; in China for Baidu, in Korea for Naver, in Czech Republic Seznam and others.

References

1. Ashmanov, I. And Ivanov, A. Site promoting in Search Engines. Viliams, Moscow. 2010

2. Bradely P. *Multi-search engines – a comparison.* http://www.philb.com/msengine.htm Accessed in May 2012

3. Brin, S., Page, L., Motwani, R. and Winograd, T. *The PageRank Citation Ranking: Bringing Order to the Web.* Computer Networks and ISDN Systems, Stanford, 1999

4. Brin. S., Page, L. *The anatomy of a large-scale hypertextual Web search engine.* Computer Networks and ISDN Systems, Stanford, 1998

5. Dover, D. *Search Engine Optimization Secrets*. Wiley Publishing, Inc, Indianapolis, 2011

6. Edosomwan, J and Edosomwan, T. O. South African Journal of Science, 2010: *Comparative analysis of some search engines.* Volume 106 (11/12). Art. #169.

7. Gulin, A., Karpovich, P. *Greedy function optimization in learning to rank.* http://download.yandex.ru/company/experience/GDD/Zadnie_algoritmy_Karpovich.pdf 2009 Accessed in March 2012

8. Golub, G.H. and Van Loan, C. F. *Matrix computations.* Johns Hopkins University, Baltimore, 1996

9. Langville, A.N. and Meyer C.D. *Google's PageRank and Beyond: The science of Search Engine Rankings.* Princeton University Press, New Jersey, 2006

10. Levy, S. *In the Plex: how Google thinks, works, and shapes our lives.* Simon & Schuster, New York, 2011

11. Manning, C.D., Raghavan P. and Schutze, H. *Introduction to Information Retrieval.* Cambridge University Press, Cambridge, 2008

12. Nilsson, N.J. *Introduction to machine learning.* Computer Networks and ISDN Systems, Stanford, 1998

13. Ridings, C. and Shishigin, M. *PageRank Uncovered.* www.voelspriet2.nl/PageRank.pdf Accessed in April 2012

14. Sirovich, J. and Darie, J. *Professional Search Engine Optimization with PHP.* Wiley Publishing, Inc, Indianapolis, 2007

15. Sonnenreich, W. *A History of Search Engines.* Wiley Publishing, Inc, Indianapolis, 1997

16. Wall, A. History of Search Engines http://www.searchenginehistory.com/ Accessed in May 2012

17. Zyabrev, I., Pozharkov, O., *Statistical methods for the study of algorithms text search engine rankings,* 2010. http://www.altertrader.com/publications18.html Accessed in May 2012

18. Zyabrev, I., Pozharkov, O., *Greedy algorithms in Yandex*, 2010 http://www.altertrader.com/publications20.html Accessed in May 2012

19. 2011 Search Engine Ranking Factors http://www.seomoz.org/article/search-ranking-factors Accessed in May 2012

20. comScore Voices http://blog.comscore.com/2011/06/yandex_russia_with_love.html Accessed in May 2012

21. iProspect Search Engine User Behavior Study. http://www.iprospect.co.th/about/whitepaper_seuserbehavior_apr06.htm Accessed in May 2012

22. The size of the World Wide Web (The Internet). http://www.worldwidewebsize.com/ Accessed in May 2012

23. How much Information. http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/ Accessed in May 2012

24. SE analyzers http://analyzethis.ru/ Accessed in May 2012

25. Search Engine market share by country http://www.them.pro/Search-engine-market-share-country Accessed in May 201

26. What is Yandex.Webmaster. http://help.yandex.com/webmaster/ Accessed in May 2012

The list of all retrieved results and their relevancy

Group A Query 1: Visa to Finland St. Petersburg (Виза в Финляндию в Петербурге)

Number of monthly searches in Yandex – 34277

Number of monthly searches in Google – 22200

Position	Retrieved results in Yandex	Precision	Retrieved results in Google.ru	Precision
1	http://www.finland.org.ru/public/default.aspx?nodeid=36986&contentlan=15&culture=ru-RU	9	http://www.finland.org.ru/public/default.aspx?nodeid=36986	9
2	http://www.finland.org.ru/Public/default.aspx?contentid=192699	10	http://www.viza-absolut.ru/visa/	10
3	http://www.visadom.ru/	9	http://www.slktour.ru/paga.html	9
4	http://www.visas.ru/price/finvisa.html	10	http://multiviza.ru/countries/finland.html	9
5	http://www.finland-visa.ru/	10	http://www.rvisa.ru/visa/finland/	9
6	http://www.vizas.ru/finland/visas/	9	http://www.letimili.net/visas/finland.html	10
7	http://www.rvisa.ru/visa/finland/	9	http://polis812.ru/finskaya_viza	8
8	http://multiviza.ru/countries/finland.html	9	http://etats-schengen.ru/schengen-visa/finland/	7
9	http://archive.travel.ru/finland/formalities/visas/	10	http://www.vizashengen.ru/visa/finland	10
10	http://tonkosti.ru/%D0%92%D0%B8%D0%B7%D0%B0_%D0%B2_%D0%A4%D0%B8%D0%BD%D0%BB%D1%8F%D0%BD%D0%B4%D0%B8%D1%8E	10	http://archive.travel.ru/finland/formalities/visas/	9
Mean		9.5		9

Group A Query 2: Tours to Finland from St. Petersburg (Туры в Финляндию из Петербург)

Number of monthly searches in Yandex – 23119

Number of monthly searches in Google – 27100

Position	Retrieved results in Yandex	Precision	Retrieved results in Google.ru	Precision
1	http://www.dsbw.ru/finland	10	http://www.avit-spb.ru/tury-v-finlyandiyu	6
2	http://www.viking-travel.ru/countries/finland/	10	http://www.viking-travel.ru/countries/finland/	10
3	http://www.jazztour.ru/tours/finland/	9	http://www.scantravel.ru/countries/finland.html	10
4	http://db.travel.ru/tours/finland/	7	http://www.tur-finland.ru/	9
5	http://www.turizm.ru/finland/	10	http://www.orienta-tour.ru/	8
6	http://www.tournet.ru/finland-tour.htm	6	http://www.holidaym.ru/finland.php3	7
7	http://www.holidaym.ru/finland.php3	10	http://www.holidaym.ru/finland.php3	6
8	http://www.ros-tur.ru/direction/20	10	http://west-travel.ru/tours.phtml?co10untry=1&page=12	5
9	http://west-travel.ru/tours.phtml?country=1&page=12	10	http://finland.grandtour.ru/	10
10	http://www.ayda.ru/finland/	10	http://www.orbita.travel/#Content	6
Average quality		9.2		7.7

Group B Query 3: "Поездка в Финляндию" (Eng.: Trip to Finland)

Number of monthly searches in Yandex – 9925

Number of monthly searches in Google – 4400

Position	Retrieved results in Yandex	Precision	Retrieved results in Google.ru	Precision
1	http://www.naotdix.ru/finland/	10	http://www.orienta-tour.ru/	10
2	http://archive.travel.ru/finland/hints/	10	http://finnish.ru/to_finland/index.php	10
3	http://www.stopinfin.ru/road/byauto/	9	http://www.fi1.ru/	9
4	http://100dorog.ru/club/stories/32095/	6	http://www.stopinfin.ru/road/byauto/	9
5	http://www.nevatravel.ru/tours/docs/4219/	4	http://www.orbita.travel/#Content	10
6	http://www.viking-travel.ru/countries/finland/	9	http://www.sapsantrans.ru/	8
7	http://www.tury.ru/country/id/finland	5	http://www.asb-tur.ru/	8
8	http://www.kurortmag.ru/region/finlyandiya/	3	http://turizm.inspb.ru/L89/index.html	6
9	http://paraisol.ru/documents/documents_5.html	4	http://fintaxi.ru/	10
10	http://www.finka.spb.ru/otz.htm	5	http://estur.ru/	8
Average quality		6.5		8.8

Group B Query 4: "Отдых в Финляндии" (Eng.: Holidays in Finland)

Number of monthly searches in Yandex – 22792

Number of monthly searches in Google – 9900

Position	Retrieved results in Yandex	Precision	Retrieved results in Google.ru	Precision
1	http://www.jazztour.ru/finland/	10	http://www.jazztour.ru/finland/	10
2	http://www.ayda.ru/finland/	10	http://www.ehtari.ru/	10
3	http://www.dsbw.ru/finland	6	http://www.viking-travel.ru/countries/finland/	9
4	http://www.turizm.ru/finland/	6	http://poiskvill.ru/Finlyandiya/	7
5	http://www.fincot.ru/	5	http://www.gotofinland.ru/	6
6	http://www.veditours.ru/	3	http://www.tur-finland.ru/	9
7	http://www.naotdix.ru/finland/	4	http://www.ayda.ru/finland/	10
8	http://www.rgb-tour.ru/country/finland/response/	5	http://www.tamirusu.ru/countries/64/42/	10
9	http://www.strana-suomi.ru/	5	http://www.cotfin.ru/	10
10	http://www.travelfinland.ru/	7	http://www.prostor-tour.ru/	7
Average quality		6.1		8.8

Group B Query 5: "Финляндия на выходные" (Eng.: Weekend in Finland)

Number of monthly searches in Yandex – 1956

Number of monthly searches in Google – 1600

Position	Retrieved results in Yandex	Precision	Retrieved results in Google.ru	Precision
1	http://finnish.ru/finland/holiday/index.php	9	http://vk.com/finland	10
2	http://www.velena-travel.ru/c_finland.php	7	http://adm.ru/scand/fin/stat/stat21.htm	5
3	http://www.virazh-tour.ru/weekend/	8	http://www.stopinfin.ru/info/	6
4	http://www.prozapad.ru/fin_weekend.html	7	http://www.orienta-tour.ru/	7
5	http://da.fi/1123.html	10	http://suomiclub.ru/	4
6	http://infinland.ru/?p=country/holiday	7	http://finnish.ru/finland/holiday/index.php	9
7	http://www.fincot.ru/weekends.html	9	http://www.finka.spb.ru/	5
8	http://www.busline.ru/shopping/art74.html	7	http://www.nicktour.spb.ru/	4
9	http://www.goweekend.ru/	4	http://www.orbita.travel/#Content	7
10	http://www.fi1.ru/	6	http://fintaxi.ru/	9
Average quality		7.4		6.6

Group B Query 6: "Отели в Финляндия" (Eng.: Hotels in Finland)

Number of monthly searches in Yandex – 5473

Number of monthly searches in Google – 3600

Position	Retrieved results in Yandex	Precision	Retrieved results in Google.ru	Precision
1	http://www.tophotels.ru/main/hotels/3	6	http://catalogue.horse21.ru/finland+hotels	8
2	http://catalogue.horse21.ru/finland+hotels	8	http://www.hoteldiscount.ru/hotels/finland/	6
3	http://hotels.turizm.ru/finland/	7	http://www.tophotels.ru/main/hotels/3/	6
4	http://www.votpusk.ru/hotels/hcity.asp?CN=FI	9	http://www.agoda.ru/europe/finland.html	10
5	http://fin-digest.ru/info/hotels/	10	http://www.rgb-tour.ru/country/finland/hotel/	10
6	http://www.city-of-hotels.ru/903/finliandiia/1.html	10	http://www.turpravda.com/fi/	7
7	http://www.turpravda.com/fi/	7	http://www.tournet.ru/finland-hotel.htm	4
8	http://www.hoteldiscount.ru/hotels/finland	6	http://hotels-turris.ru/leikari.html	8
9	http://www.tury.ru/country/id/finland	5	http://www.stopinfin.ru/	7
10	http://fi.otzyv.ru/	3	http://adm.ru/scand/fin/hotel/hotel_fin_glav.htm	5
Average quality		7.1		7.1

Group C Query 7: "Гостиницы Финляндии" (Eng.: Hotels in Finland)

Number of monthly searches in Yandex – 781

Number of monthly searches in Google – 1900

Position	Retrieved results in Yandex	Precision	Retrieved results in Google.ru	Precision
1	http://www.hotels.ru/hotels/finland/cities.htm	10	http://catalogue.horse21.ru/finland+hotels	7
2	http://www.tournet.ru/finland-hotel.htm	7	http://www.hoteldiscount.ru/hotels/finland/	5
3	http://www.votpusk.ru/hotels/hcity.asp?CN=FI	6	http://www.tournet.ru/finland-hotel.htm	7
4	http://www.tophotels.ru/main/hotels/3	8	http://www.viking-travel.ru/countries/finland/hotels/	9
5	http://catalogue.horse21.ru/finland+hotels	7	http://www.votpusk.ru/hotels/hcity.asp?CN=FI	6
6	http://www.hros.ru/country/finland.ru.html	8	http://hotels-turris.ru/leikari.html	7
7	http://www.norvica.ru/content/hotels	6	http://www.slktour.ru/finhotels.html	8
8	http://www.hoteldiscount.ru/hotels/finland	5	http://apartespoo.com/	6
9	http://paraisol.ru/paraisol/finland/hotel/	7	http://www.norvica.ru/content/hotels	6
10	http://hotels.turizm.ru/finland/	9	http://www.fi1.ru/%D0%BE%D1%82%D0%B5%D0%BB%D0%B8-%D1%84%D0%B8%D0%BD%D0%BB%D1%8F%D0%BD%D0%B4%D0%B8%D0%B8	5
Average quality		7.3		6.6

Group C Query 8: "Отели Хельсинки" (Eng.: Hotels in Helsinki)

Number of monthly searches in Yandex – 3940

Number of monthly searches in Google – 2400

Position	Retrieved results in Yandex	Precision	Retrieved results in Google.ru	Precision
1	http://www.helsinki-hotels.net/rus/	10	http://www.hoteldiscount.ru/hotels/finland/helsinki/	10
2	http://ru.hotels.com/de475103/gostinitsy-hel-sinki-finlandia/	10	http://content.oktogo.ru/%D0%A5%D0%B5%D0%BB%D1%8C%D1%81%D0%B8%D0%BD%D0%BA%D0%B8_t2547.aspx	9
3	http://www.agoda.ru/europe/finland/helsinki.html	10	http://ibooked.ru/hotels/finland/helsinki	10
4	http://www.hotelscombined.ru/City/Helsinki.htm	10	http://catalogue.horse21.ru/finland+hotels/helsinki+hotels	9
5	http://www.star-line.ru/finlandtours/hotelshelsinki	7	http://www.hotels.su/newDes/cities_5887_pFI_hotels_HELSINKI.html	9
6	http://www.hoteldiscount.ru/hotels/finland/helsinki	10	http://www.agoda.ru/europe/finland/helsinki.html	10
7	http://www.tripadvisor.ru/Hotels-g189934-Helsinki_Southern_Finland-Hotels.html	10	http://ru.hotels.com/de475103/gostinitsy-hel-sinki-finlandia/	10
8	http://catalogue.horse21.ru/finland+hotels/helsinki+hotels	9	http://www.helsinki-hotels.net/rus/hotels.htm	10
9	http://fi.otzyv.ru/?city=265	8	http://www.tourister.ru/world/europe/finland/city/helsinki/hotels	8
10	http://content.oktogo.ru/%D0%A5%D0%B5%D0%BB%D1%8C%D1%81%D0%B8%D0%BD%D0%BA%D0%B8_t2547.aspx	9	http://www.norvica.ru/content/hotels-finland-helsinki	10
Average quality		9.3		9.5

Group D Query 9: "Карта Финляндии" (Eng.: Map of Finland)

Number of monthly searches in Yandex – 21042

Number of monthly searches in Google – 12100

Position	Retrieved results in Yandex	Precision	Retrieved results in Google.ru	Precision
1	http://karta.fi/	9	http://www.finkarta.ru/	8
2	http://www.finkarta.ru/	8	http://www.tourister.ru/world/europe/finland/map	10
3	http://www.tourister.ru/world/europe/finland/map	10	http://karta.fi/	9
4	http://www.mapfi.ru/	9	http://infinland.ru/?p=country/maps	6
5	http://www.stokart.ru/index/finland/	7	http://www.evromap.ru/index.php/europe-maps/finland	10
6	http://finnish.ru/links/maps/index.php	10	http://www.stokart.ru/index/finland/	7
7	http://www.karta-finland.ru/	10	http://ski.spb.ru/sklons/finland/KARTA-FINLYaNDII.html	5
8	http://maps.turizm.ru/country_219.html	8	http://www.karta-finland.ru/	10
9	http://www.evromap.ru/index.php/europe-maps/finland	10	http://arvomedia.ru/content/view/2166/659/	8
10	http://www.tournet.ru/finland/carts/carta-all.htm	10	http://www.alvas.ru/finland.htm	4
Average quality		9.1		7.7

Group D Query 10: "Магазины Финляндии" (Eng.: Shoping in Finland)

Number of monthly searches in Yandex – 22792

Number of monthly searches in Google – 9900

Position	Retrieved results in Yandex	Precision	Retrieved results in Google.ru	Precision
1	http://www.magazin.fi/	10	http://www.magazin.fi/	10
2	http://www.go-shopping.fi/	9	http://www.go-shopping.fi/	9
3	http://www.rus-tourist.ru/node/28	8	http://www.infofinland.ru/index.php/what/shopping	8
4	http://www.orienta-tour.ru/magaziny-finlyandii	7	http://www.fi4ru.narod.ru/sur_kaup.htm	7
5	http://da.fi/28.html	10	http://e-finland.ru/travel/shoping/	10
6	http://www.infofinland.ru/index.php/what/shopping	8	http://fintour-spb.ru/aboutfin/shops.php	10
7	http://www.fi1.ru/%D0%BC%D0%B0%D0%B3%D0%B0%D0%B7%D0%B8%D0%BD%D1%8B-%D1%84%D0%B8%D0%BD%D0%BB%D1%8F%D0%BD%D0%B4%D0%B8%D0%B8	6	http://da.fi/28.html	10
8	http://www.to-finland.ru/index.php?id=73	6	http://finnish.ru/relax/shopping/shops.php	7
9	http://finnish.ru/relax/shopping/shops.php	7	http://www.travel.ru/news/2012/04/05/199805.html	6
10	http://marina-travel.ru/magaziny_finlyandii	9	http://tonkosti.ru/%D0%9C%D0%B0%D0%B3%D0%B0%D0%B7%D0%B8%D0%BD%D1%8B_%D0%A4%D0%B8%D0%BD%D0%BB%D1%8F%D0%BD%D0%B4%D0%B8%D0%B8	6
Average quality		8		8.3