Structuring a Modern Web Service for Users and Search Engines

Master of Science in Technology Thesis
University of Turku
Department of Computing
Software Engineering
2021
Eero Ruohola

Supervisor:
Tuomas Mäkilä

UNIVERSITY OF TURKU
Department of Computing

EERO RUOHOLA: Structuring a Modern Web Service for Users and Search Engines

Master of Science in Technology Thesis, 50 p., 2 app. p.
Software Engineering
September 2021

This thesis aims to study how search engines should be taken into account when developing an early stage web service. The thesis introduces practical methods for improving the presentation of a web service's pages in search engine results and a general overview of the working principles of search engines is provided. The thesis also goes over the history of web search engines in general before covering the history of Google in a bit more detailed manner.

The practical aspect of this thesis is conducted as a case study, where search engine legibility related improvements are made to the Skole application. Skole is a very early stage web service, were higher educations students can share knowledge and discuss their studies. The case study aims to get information about how and what kind of improvements can be made, and how they make a difference in practice. The case study also seeks to find out how important it is to follow the guidelines of search engine providers, when doing development for this kind of very early stage web service.

The results of the case study will mainly be measured by documenting how the application's pages show up in Google before and after changes have been made. Some aggregated data from analytics providers is also used to see how the made improvements have affected the usage of the application.

Keywords: Search Engines, Web Services, SEO, Google Search, PageRank, User Experience

Contents

1 Introduction

This thesis aims to study how a modern web service can be made in a way that it is pleasing to use for end users and at the same time easily processable by search engines. This covers how the visible and invisible contents of web pages can be structured in such a way that search engines get as much information from the site for their use as possible. The thesis will study this in practice by implementing search engine readability improvements in the Skole application. Because of its near monopoly like market share in the western world, the practical aspects will only be covered from the perspective of Google Search. It is important to state though that Google arguably does not have any real monopolistic powers, even though 90% of searches go through them [1].

The practical implementation of the thesis is done in collaboration with Skole Oy which the author is a co-founder of. The practical part will focus on improving the search engine legibility of Skole Oy's Skole application. In addition to practical implementing, the used research method will be literature study. This thesis will not be covering how the visibility of a web service's content can be increases via any paid advertisements or free social media marketing. It also does not cover how one can increase backlinking from other websites or other marketing techniques, but instead focuses just on the technical aspect of making a site more easily discovered and indexed by search engines.

The thesis has three main research questions:

- *RQ1*: How to structure the content of a website so that it is easy for search engines to read, and at the same time intuitive to use for humans?
- *RQ2*: How the search engine legibility of an existing application can be improved in practice?
- *RQ3*: How important is search engine legibility in an early stage web service?

The thesis's second chapter will go over search engines and their working principles in a general way. It starts with a short introduction about the history of search engines and then explains in a quite high level how they find and rank web pages. The last two sections will be about what kind of techniques can be used to manipulate search engines illegitimately and how unwanted content can be hidden from them. The third chapter will focus on Google Search. The chapter goes over the history of Google Search and Google as a whole, and then explains how users interact with the search engine in practice. The chapter ends with a section that covers Google's pioneering PageRank algorithm. The fourth chapter introduces the case study of the thesis. The chapter introduces the Skole application and studies how it works currently. The fifth chapter will focus on making changes to Skole which improve its search engine legibility while maintaining good usability. The case study concludes in the sixth chapter, where the results of the study are analyzed. The analyzing will go over the possible improvements that occurred based on the practical implementation and will tell about any flaws that could not be fixed and about any future considerations that could still be done. The seventh and final chapter will then conclude the whole thesis.

2 Search Engines

The basic idea of a search engine is to collect the information scattered across the World Wide Web into a single place, where it can be easily searched and discovered by humans. This chapter will briefly go over the history of search engines from the early days of the Web to the current time. This also goes through how search engines in general see and process web pages and what kind of information they look from them, aiming to answer *RQ1.* The last two sections will cover how website creators can hide content from search engines, and what malicious practices exist around search engine optimization.

2.1 History of Search Engines

A search engine is an application, usually hosted on a website, that allows an end user to find relevant webpages by providing some search terms which are then matched against the content indexed by the engine. The returned results are then shown the user usually in the order of relevance. This relevancy can be subjective and is one of the biggest differentiators between different search engines. Search engines aim to provide as relevant search results as possible, and they are constantly improving their proprietary algorithms to make them more useful, so that more people would end up using them. [2]

The first search engine on the Internet was Archie. Archie was created in 1990 by three computer science students: Alan Emtage, Bill Heelan and J. Peter Deutsch, who were studying at McGill University in Montreal at the time. Archie searched all FTP servers available on the Internet, downloaded all files from them, and allowed users to search those files by the file names. The name Archie was derived from the word *Archive*. Even though Archie was the first tool to be used to search the Internet, it only indexed the names of the pages and not their contents, and it did not allow searching the World Wide Web. [3] In the early days of the Web only way to find websites on it was through links from other websites. Though quite early on websites dedicated to listing other sites in a catalog like format appeared as well. One of these sites was the Open Directory Project, also known as DMOZ, which was created by two employees of Sun Microsystems [2]. The first primitive search engine for finding content from the Web was W3 Catalog [3] (originally called Jughead [4]). Oscar Nierstrasz — the creator of W3 Catalog — launched the tool in 1993 because he deemed the existing solutions problematic since they only listed the pages, but did not allow searching from the list. His implementation used a Perl script which would periodically go through some authoritative page lists and download their contents into a searchable file and exposed that as a search bar on a website. W3 Catalog got eventually shut down in 1996 because of lack of interest in maintaining it, its fragility and implementation overhead, and because competing tools had gotten much better at that point.

Even though searching from the complete contents of webpages is of course the standard today, this was not possible with the first search engines. The first search engine which indexed every word on the pages was Brian Pinkerton's WebCrawler, which went live in April 1994. It was initially a desktop application [3], but later turned into a web service, and is actually still accessible at its original URL at `www.webcrawler.com`.

The early web search engines mostly relied on the content of a website and especially keywords that appeared on the page to rank the search results. If the user's search term appeared many times on a certain web page that page would then be shown pretty high up in the results. This made it so that many webmasters would stuff their pages full of mostly irrelevant keywords to get their page be shown higher up in search results. These keywords were often stored as a block of text on the bottom of webpages, or sometimes they were written in the same color as the background, so they would be invisible for the end user. This showed that the keyword based system was clearly flawed and there was room for something better.

The first search engine algorithm to use links as the means to determine the importance of a website was RankDex. RankDex was invented in 1996 by Robin Li, who is nowadays a billionaire and one of the richest persons in China. [2] Many search engines implemented their own versions of this link based ranking of sites, but an obvious problem in it was that all links were treated as equally important. A website owner could just make other websites with thousands of links to their primary site and thus increase their site's ranking maliciously. This is what Google's founder Lawrence (Larry) Page wanted to combat with his PageRank algorithm. The algorithm was patented by Stanford University in 1998 [5], since Larry Page was studying in Stanford at that time. PageRank is at the core of Google Search. It works by weighting the links so that links from important and popular websites count more in the ranking than links from smaller and less known sites. PageRank was crucial in Google Search's rise to the most used search engine, since it prevented most of the unwanted spam content that troubled many other search engines at that time. [2] This thesis will go over PageRank more thoroughly in chapter three.

Many search engines emerged after Google Search, but it constantly increased in market share even with the ever stronger competition. Google Search had been launched 1996, and already at the end of 1998 it had risen to be the 7th most used search engine with a 5% market share. By 2003 it had become the most popular and had a 23% share. From there the lead just kept on increasing and increasing. [6]

In 2009 Google Search was used to perform over 70% of the searches in the world, and according to Data Is Beautiful the percentage had gone to 77% in 2019 [6]. The number cannot be exactly measured though, and different sources can give slightly different values. For example NetMarketShare shows Google's market share to have been 83% at that time [7], Statista shows it to have been 88% [8], and Statcounter tells that the number would have been almost 93% [9]. Regardless of which source one believes, it is extremely clear that Google has a massive share of the search market, and there does not seem to be any signs of that stopping.

2.2 How Search Engines See and Rank the Web

The purpose of a search engine is to offer search results for its users. Before being able to show any results, a search engine needs to have ways of finding websites and gathering information from them. Different search engines of course have different kind of approaches to this, but the basic principles in modern search engines are likely quite similar.

Google uses three steps to generate search results: crawling, indexing, and serving. Crawling is the process of finding and visiting pages. Indexing is the step where Googlebot (Google's web crawler) goes through the contents of found pages and processes key information from them and adds that information to Google's massive search index. Serving pages happens when a user searches something and Google's servers return the most relevant search results for the search. Serving is a complex process and the relevancy of the results is determined by hundreds of different parameters. [10]

The technique for improving a website's visibility and ranking in search engines is called search engine optimization (SEO). SEO can be divided into two subcategories: on-page SEO and off-page SEO. On-page SEO focuses on making the site easy to crawl and understand for search engines by considering how the HTML is structured and what kind of keywords and content is on the web page. Off-page SEO involves how the page interacts with other sites and the Web as a whole. Off-page SEO techniques involve for example getting links from other websites to the site and promoting the site in social media. [11] This thesis focuses almost solely on on-page SEO, and especially on how the website's general appearance on a search engine can be improved by implementing on-page SEO techniques, and not so much on how to get a page to rank higher in a search engine.

On-page SEO covers everything that a website creator can do on their own site to help their site get good visibility in a search engine and to make its content easily discoverable from search engines by potential users. Unlike off-page SEO, on-page SEO is fully controllable by the webmaster, and it can be improved practically endlessly by tweaking the structure of the website, and most importantly by getting valuable content on it that people actually want to consume. A big part of on-page SEO — like the actual webpage content — is directly visible for the end users, but a part of it is just making sure that the HTML that is not visibly rendered is also well-structured.

Some things to consider when doing on-page SEO according to Swapna and Anuradha [11] are: the URL of the webpage, meta tags, such as the description; the title and header tags; keywords; spelling and grammar; emphasizing text with bold or italics; page loading times; and the uniqueness of the content. Kumar and Paul [12] do not say anything about using bold or italics, nor about spelling and grammar, but they seem to otherwise agree about the techniques stated by Swapna and Anuradha. Keeping on eye on the spelling and grammar will also most certainly be appreciated at least by the end users of the website.

Modern search engines, like Google, generally find and index new websites automatically without any actions from the site owner. Nevertheless, if a site is completely new, or is not linked at all from other indexed sites, search engines might not find it. Google Search Console's help recommends in these situations to verify the ownership of the site through them and then submitting some of site's pages from Google Search Console either one by one, or through a sitemap. [13]

One way to help search engines better understand a website is to provide a sitemap file on it. A sitemap is a special page on a website that lists all other pages, images, and other files on the site that a search engine should be aware of, and possibly tells some extra information about them. This extra information can include details about how important the pages are relative to each other, which pages are translations of other pages, and when was the last time a page was updated [14].

Google recommends sitemaps for all large sites, sites that have a lot of pages that do not link to each other, sites that are new and are not linked much to from other websites, and for sites that have a lot of image or video content [15]. Google Search Console documentation tells that having a sitemap will never have any negative effects [15], so any website looking to improve their search engine legibility should likely have one. Google allows sitemaps to be submitted in three different formats: XML, text file, or syndication feed; and in addition, their documentation explains that all websites generated with Google Sites automatically get a suitable sitemap generated for them. The documentation also tells that regardless of the format used to submit the sitemap, it should adhere to the standard sitemap protocol. [15] The protocol defined in `www.sitemaps.org` contains the full specification for sitemaps, including the syntax of them, the different formats allowed, and how the protocol can be extended. The protocol also states that the maximum size of a single sitemap is 50 000 URLs or 50 megabytes, but websites with more pages can easily circumvent this by making a sitemap index file, which can then link to a number of other sitemaps. [16]

Another file on a website, which is used for directing the crawling of search engines is a `robots.txt` file. `robots.txt` is a text file located at the root of a website, which tells search engines about what pages or files they should not request from the site [17]. `robots.txt` is pretty much the opposite of a sitemap. A sitemap should be used to encourage crawling of certain pages and a `robots.txt` should be used to prevent crawling. Preventing crawling does not necessarily prevent from Google from indexing a site though. Google might index a page and infer its content just because it for example finds a link to it from another website, without ever needing to parse the page's content. [10] Because of this, Google's documentation [17] explicitly says that `robots.txt` should not be used to keep a web page completely out of Google and one should use other measures to accomplish that, such as requiring authentication to access the page or using the `noindex` directive.

2.3 Manipulation of Search Engines

For as long as there has been search engines, there have been websites that try to trick their content into being shown higher up in the results as it organically would. Google's documentation lists a lot of methods that website owners should not use, and which Google might automatically or manually act upon. These kinds of practices are also known as "black hat SEO" techniques [18]. This section will go over these methods and describe why they are considered harmful.

Automatically generated content on websites is content that has been created without human input or review. Google might take manual action on such content if it deems that the content's sole purpose is to rank the site higher. A common example of auto-generated content that Google may act upon is content that is scraped from other websites without adding anything valuable to it. Other types of auto-generated content are programmatically generated text, automatically translated text, and text that is simply keywords one after another. [19] In addition to scraping, another practice that Google shuns upon is "thin" affiliate programs, which do not add enough value on their own and simply contain affiliate links to products together with copied product information. [20]

“Sneaky redirect“ is the term Google uses for redirects that take users to a different page than what is shown for search engines crawlers. This harms the user since they can get the impression from the search results that the site’s content is valuable for them, but then in reality the site redirects them to some completely different page with often spam-like advertisement infested content. Serving different content for users than what is served for Googlebot is a direct violation of Google’s Webmaster Guidelines. This practice is also known as cloaking. [21]

An attempt to manipulate Google search result ranking with the use of links is called a “link scheme”. Google documentation tells that link schemes include for example buying links that pass PageRank points. Buying or selling links on their own is not forbidden though, one just needs to mark the links with the appropriate `rel="sponsored"` attribute, so that Google doesn’t consider that as a link which would increase the PageRank score of the linked site. Other examples of link schemes are large planned partnerships where sites link to each other based on a mutual agreement, and automatically or programmatically creating links to a site. Other harmful uses of links are uses where the links are not for the users, but just for search engines; an example of this could be an article where every other word is a link to a product on an e-commerce site. [22]

Another trick website owners use to get their site ranked higher — by making it seem like the content is about something else than it actually is — is the use of invisible text or links. Text or links can for example be hidden from users of the website with 0 point fonts, by using text colors that blend into the background or by positioning them off-screen with CSS. [23] The usage of these kinds of methods is a clear indicator that the site owner is trying to manipulate search engines and not creating the site for users first.

The term “keyword stuffing” means the act of adding a lot of keywords, that the site is trying to rank for in search results, to a site [24]. Before the invention of search engines that use link based search engine algorithms, such as RankDex and Google Search and its PageRank, keywords were one of the primary ways search engines evaluated websites. This is not necessarily the case anymore and according to Google, having concentrated keywords blocks or just repeating keywords in a text unnaturally can actually negatively affect a site's search result ranking [24].

Not all content that violates Google's guidelines are placed there consciously by the site owner. Even a good site can have a problem with user generated spam content. Google can warn a site's owner about excessive user generated spam, and if they consider there to be too much of it, they can also take manual action against it. Google tells that the best cause of action for a website owner is to actively moderate against this kind of content, since it can also shadow the more valuable content of the site in Google search results. [25] Some ways to prevent user spam from occurring on a site in the first place is to require users to log in with email verification before they can create comments, placing all comments and other user generated content in a review queue before they are published, or requiring users to solve CAPTCHAs before posting content. Other deterrents for spammers could be to use `rel="nofollow"` attributes on all user posted links so that they do not give PageRank points for the linked sites and using `noindex` meta tags for profile and other pages created by new users. [26]

The most harmful practice from the perspective of web users that Google forbids in their search documentation, are downright maliciously behaving sites. Sites that try to trick their users into downloading malware, sites that inject unwanted ads to other pages, and sites which aim to change the user's browser settings without their consent are all strictly against Google's guidelines. [27] One could also assume that these kinds of intentionally malevolent sites are the most likely to get manual action taken upon them by Google.

Black hat SEO techniques are forbidden by search engines because they actively harm the users, who are just trying to find as relevant information as possible. Websites which implement these practices are trying to get their site ahead of others and doing so makes the actually valuable sites that users would really want to see and from which they would benefit the most, get hidden further up in the results, making finding them harder. Search engines are also all about creating a level playing field for website owners and malicious players really hurt the business of all the well-behaving sites that are competing for the same search terms or keywords. [28]

Google has a lot of measures in place to stop most of the aforementioned disallowed methods. Search Engine Journal tells [29] that a Google employee has explained in a Reddit thread that automatic methods can for example ignore the bad parts of the site completely when constructing search results, and they can also automatically stop the badly acting site from using certain search features at all. They also tell that a site's ranking can simply be lowered as an automatic measure. Manual action is a harsher mean of intervening a website that is breaking the rules Google has put in place. The employee also told that this kind of penalty does not result in a site getting permanently removed from Google's results without possibility for remediation, but completely abolishing a site's content from the results, for as long as the issue stands, is very much a possibility.

2.4 Hiding Content From Search Engines

On the opposite side from getting web pages to be visible in search engines is getting them to not show up there at all. There are many valid reasons for wanting a website or a web page to not be visible in search engines. For example test versions of sites, admin panels, or private files are content that often is wanted to be kept out of search engines.

As was previously stated, Google just finds most websites without any input from the sites' owners. This means that if a webmaster does not want their site or a part of it showing up in Google's search results, they have to quite actively prevent it. Google's documentation lists quite a few different ways to block content from appearing in its results. The first and most obvious way is to remove the content from the Web, this might not prevent historical versions from still appearing in Google for some time though [30]. The second-strongest measure is to password protect the content. By setting up for example "Basic" HTTP authentication scheme on a site, then any users will be required to enter a user ID and a password before they are able to view the content. This fully blocks Googlebot and other web crawlers from accessing it [30]. Password protecting content is the ideal way to hide private information from search engines, since it simultaneously makes sure that no unauthorized viewers can get access to it even if they would know the URL of it. The third way to block content form being indexed by Google, and the only way that still keeps the content publicly accessible for anyone, is to use the `noindex` directive [30]. `noindex` is a directive which can be used to tell Google and other web crawlers to not index a certain page or a site. It can be either specified in a meta tag [31]:

```html
<head>
  ...
  <meta name="robots" content="noindex" />
  ...
</head>
```

or by returning `noindex` value in the `X-Robots-Tag` HTTP header.

3 Google Search

Google Search is perhaps the single most used and most influential software of the twenty-first century. Searching anything from the web is for a reason called just googling nowadays. `google.com` is also the single most visited website in the world according to the analytics site Alexa [32].

3.1 History of the Search Engine and the Company

An old, since removed article [33] from Google's website tells that Google's initial journey started in 1995 when Larry Page and Sergey Brin met for the first time in Stanford University. In 1996, they started to work on their first project; a search engine named after its use of back-links [34] to measure page rankings: BackRub. Soon after, in 1997 BackRub got renamed to Google. The name Google came from the word googol, which is ten raised to the power of a hundred. The name symbolized their mission to make the massive amount of information in the Web easily accessible and available [33]. Around the same time they registered the domain `google.com`. In August 1998, Andy Bechtolsheim invested $100,000 while addressing the check to the yet unfounded *Google Inc.* after seeing a quick demo about the product [34]. On September 4th, Page and Brin registered a company with the name that the check was made out to. They initially worked out of a Garage in Menlo Park in California, and at the start of 1999 moved to a new office space in Palo Alto after the garage got small for the then eight employees. In the summer of 1999, Google

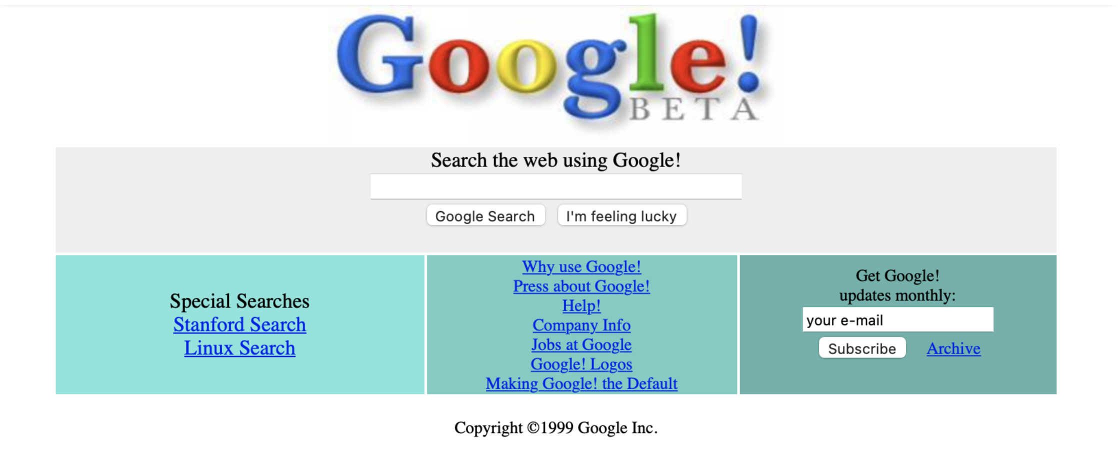


Figure 3.1: Google's frontpage in 1999 [35].

raised $25 million in funding from Sequoia Capital and Kleiner Perkins and soon after they moved offices again, this time to Mountain View.

In May 2000, `google.com` got its first translated versions, and surprisingly Finnish was in the 10 offered languages, along with for example French, German, Spanish, and Swedish. In October of the same year, Google launched AdWords. This was probably one of the biggest moments in the company's history actually, since advertisements are a massive part of modern Google, and accounted for 80% of Google's revenue in 2020 [36]. The next major advancement in Google Search came in July 2001 when Google Images launched. At launch, Google Images offered search access to 250 million indexed images [33].

In spring of 2002, Google released search APIs which allowed developers to programmatically search through the Web. The APIs did not exist for long, and Google discontinued them in 2006 [37]. Nowadays, Google explicitly disallows doing any kind of automated Google Search queries without getting special permission from them to do so [38]. The reason is most likely that they do not want external services or any competitors to be able to benefit from their strong search result generation.

In March 2004, Google's California based headquarters moved to their current location in 1600 Amphitheatre Parkway. This place is also known as Googleplex. One month after that they launched Gmail. Gmail is one of Google's most recognized and successful products. In 2018 Gmail's official Twitter account announced [39] that they had 1.5 billion users at that point. We can assume that the number is likely higher today. In late 2007, Google launched Android [33]. Mobile computing is of course massive nowadays and already in 2015 Google reported [40] that in many countries over half of all Google searches were done on mobile devices. In September 2008, Google Chrome browser was released. Now in 2021, Chrome has around 65% of the global browser usage market share [41]. The mobile trends continued in November 2008, when Google added voice search support to their iPhone app. In 2009, five years after its launch, Gmail's beta label was removed, and a stable version was released. Google's next major product announcement happened when in 2012, when they revealed Project Glass. Google Glass is a pair of augmented reality glasses which allows access to notifications and contextual information about the surroundings with a hands-free approach. It was likely aimed to greatly overhaul the experience of getting access to information, even without doing any manual searches.

On August 11, 2015, Google announced [42] that a new holding company Alphabet Inc. was founded and Google Inc. would move under Alphabet. In their announcement, they tell that one of the reasons for the new company is that more operations can be moved from Google to other more specific companies, making Google slimmer and more focused on its core business. An article in Harvard Business Review [43] states that one reason for the new corporate structure could be to increase the transparency of the key figures of the different business sectors that previously were all bundled under Google Inc.'s accounting numbers. But it also states that Alphabet did not yet get over the challenge that it needs to convince investors to buy into all the different ventures that Alphabet is participating in as a whole, since none of the child companies will still be listed separately. It is also important to note that even though Google was under a lot of antitrust scrutiny from the European Union around that time, according to Business Insider [44] forming Alphabet was not an attempt to mitigate those concerns, since it did not split the company up in ways that would lower the power that Google Search has in boosting the conglomerate's other businesses.

3.2 How Users Interact with Google Search

Google offers few clearly distinct ways for users to do searches: textual search from their website and from browser's search bars, reverse image search for searching with an image file, and voice search through their mobile app and Google Assistant. The classic text based search is the simplest one of the three, but it is not simple at all behind the scenes.

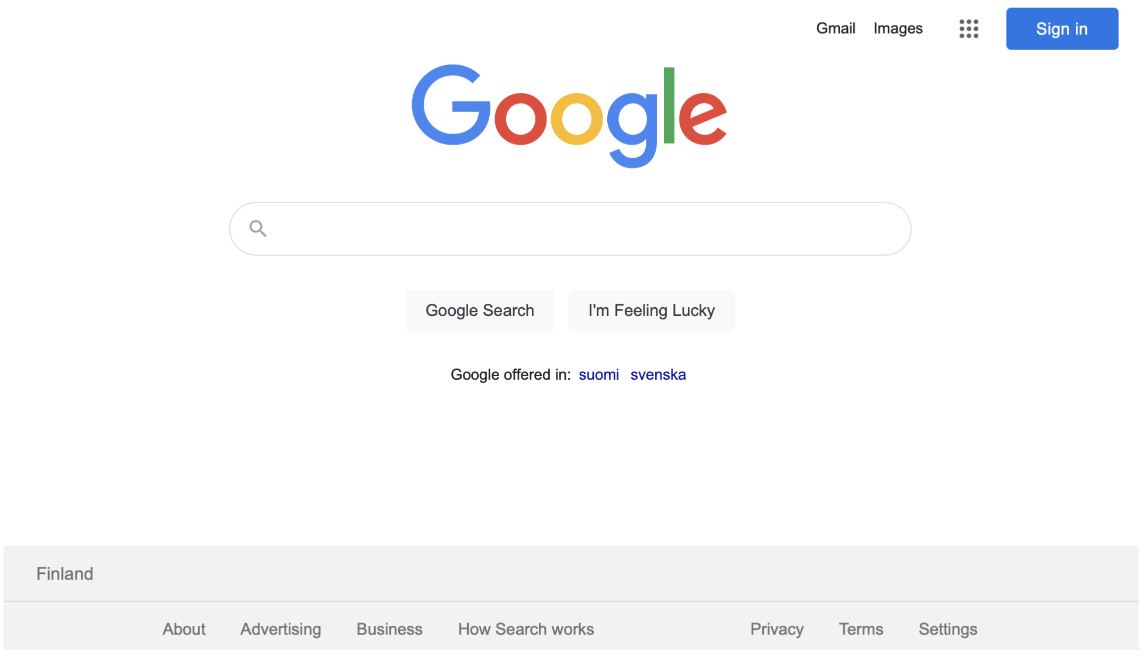


Figure 3.2: Google's frontpage in 2021 [45].

When a user goes to the world's most popular website and types in their search query to the familiar looking white input field, a lot of things will happen on the background before the search results are returned to the user. The process as a whole is of course a heavily guarded trade secret, but Google does still shed quite a bit of light on it with their *How Search algorithms work* [46] guide. Everything starts with trying to understand the query that the user has made. Probably the most straight forward part is fixing obvious typos and spelling errors. This makes understanding the query easier. There also are multiple ways to search for the same thing, so Google has built an extensive synonym system which maps similar sentences and words to the same underlying meaning. In addition to just synonyms and the meaning of words and sentences, Google also tries to understand the category of the information that the user is looking for, and whether the query is trying to search for something very specific or something more general or abstract.

After the query itself has been processed the second step of the search flow is to find relevant results for that query. Google's guide explains that one of the most basic things in that step is looking at which websites contain keywords that appear in the query. The guide also tells that purely the amount of the keywords is not everything, but the context of them and whether they appear naturally in the text is crucial. This also echoes well with what was explained about keyword stuffing earlier. Besides just keywords, Google also uses the data of how other searchers have previously interacted with the websites after searching for similar kind of information. The guide does not explain this much further, but a simple version of how it would work is for example to look for websites that users have clicked as their last click when viewing search results. The site that a user has ended their search to is likely a good search result for other users as well. This simple logic of course does not work for those workflows were a user immediately opens for example the topmost ten search results and starts skimming through them.

Probably Google has thought about that too, and perhaps they for example take the timing of the result clicks into account as well.

Google's guide explains that the relevance of potential search results is also affected by the authoritativeness of the sites. Authoritativeness of a website in a certain topic can be measured with for example the PageRank algorithm, that has been one of the core parts of Google Search all the way from the start. The guide also tells that since 2018, Google has taken the loading speed of websites into account as well, so faster websites get a preferential treatment in search results. This makes a lot of sense, since a website with really valuable information is really not that valuable if it takes a minute to load on a slow mobile internet connection. The final thing that is told to affect the returned search results is the context of the search; this context includes for example the country or the city where the search is conducted from and the user's possible Google account settings or previous searches.

Based on the above descriptions, getting search results for a simple query such as "wat is the whether like today" *could* work for example like this:

1. First the wrongly spelled word *wat* gets changed to the obvious correct spelling *what*.

2. Language processing tries to understand the sentence, and notices that the sentence would be more meaningful and natural flowing if *whether* would be *weather* instead so that gets changed.

3. The today part of the search is replaced with a token for that day's date, e.g. 2021-04-21.

4. Weather information is very location specific, so a rough location estimate is done based on the searcher's IP-address and that is taken to be a part of the search.

5. The whole preprocessed search query is transferred to some raw search tokens:
   `Query: <Weather report>; Location: <Turku, Finland>;`
   `Date: <2021-04-21>;`

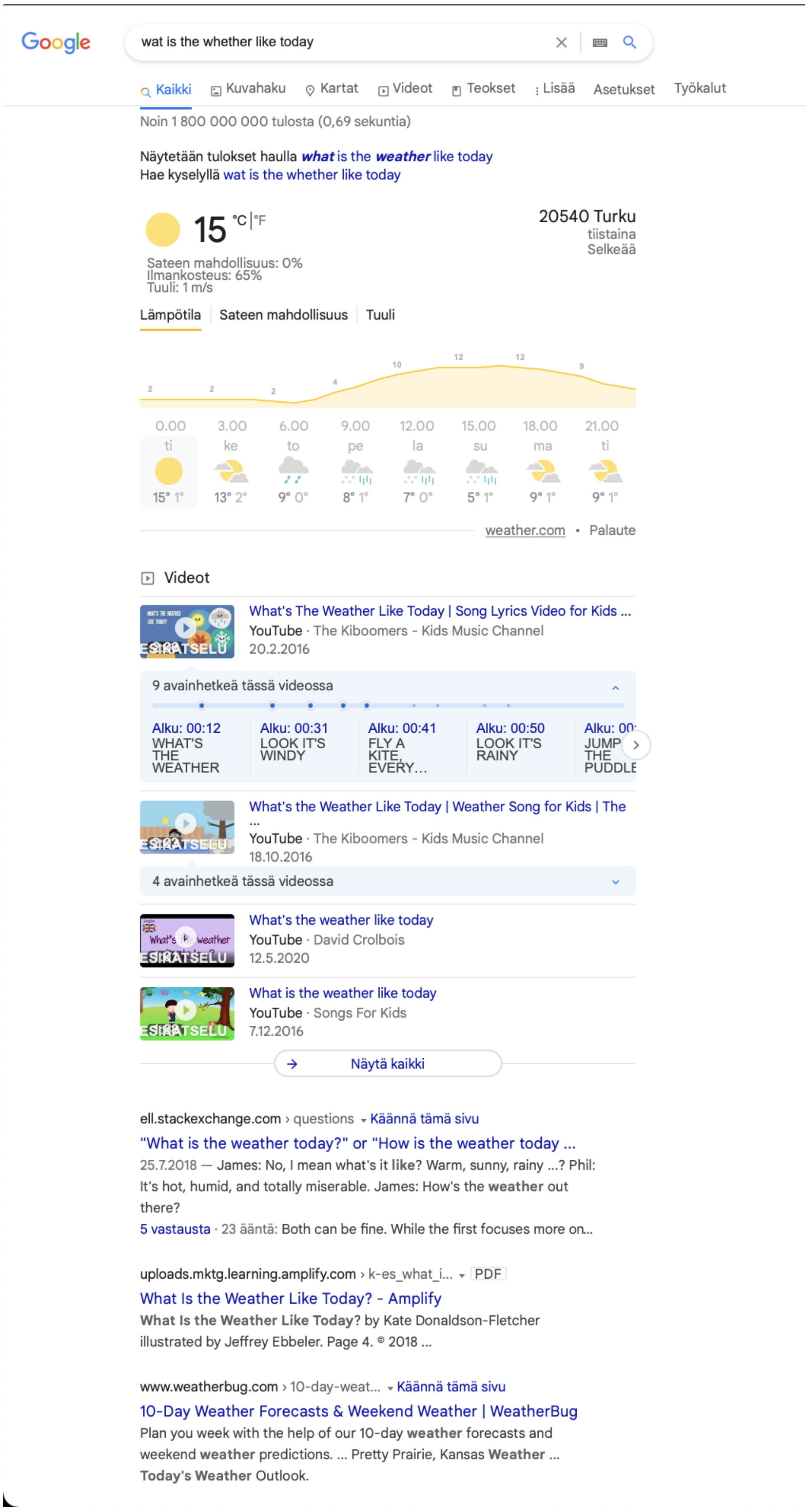
6. The query is in English, so English language results are probably most relevant.

7. The query had today's date in it so recently updated information is preferred.

8. The search tokens are matched against the search index, trying to find results which match the search tokens well.

9. The retrieved results are presented to the user in the search results page (figure 3.3).

The fact is though that Google's inner workings are much more complex than this naïve attempt. This can be clearly seen by trying to simulate the above logic by just searching for "Weather report Turku", and limiting the results to the past 24 hours. Even the topmost results in figure 3.3 are very different compared to the original query in figure 3.4.



Google

wat is the whether like today

Kaikki Kuvahaku Kartat Videot Teokset Lisää Asetukset Työkalut

Noin 1 800 000 000 tulosta (0,69 sekuntia)

Näytetään tulokset haulla ***what*** is the ***weather*** like today
Hae kyselyllä wat is the whether like today

15 °C | °F

20540 Turku
tiistaina
Selkeää

Sateen mahdollisuus: 0%
Ilmankosteus: 65%
Tuuli: 1 m/s

Lämpötila | Sateen mahdollisuus | Tuuli

0.00	3.00	6.00	9.00	12.00	15.00	18.00	21.00
2	2	2	4	10	12	12	9

ti	ke	to	pe	la	su	ma	ti
15° 1°	13° 2°	9° 0°	8° 1°	7° 0°	5° 1°	9° 1°	9° 1°

weather.com • Palaute

Videot

What's The Weather Like Today | Song Lyrics Video for Kids ...
YouTube · The Kiboomers - Kids Music Channel
20.2.2016

9 avainhetkeä tässä videossa

Alku: 00:12 WHAT'S THE WEATHER
Alku: 00:31 LOOK IT'S WINDY
Alku: 00:41 FLY A KITE, EVERY...
Alku: 00:50 LOOK IT'S RAINY
Alku: 00: JUMP THE PUDDLE

What's the Weather Like Today | Weather Song for Kids | The ...
YouTube · The Kiboomers - Kids Music Channel
18.10.2016

4 avainhetkeä tässä videossa

What's the weather like today
YouTube · David Crolbois
12.5.2020

What is the weather like today
YouTube · Songs For Kids
7.12.2016

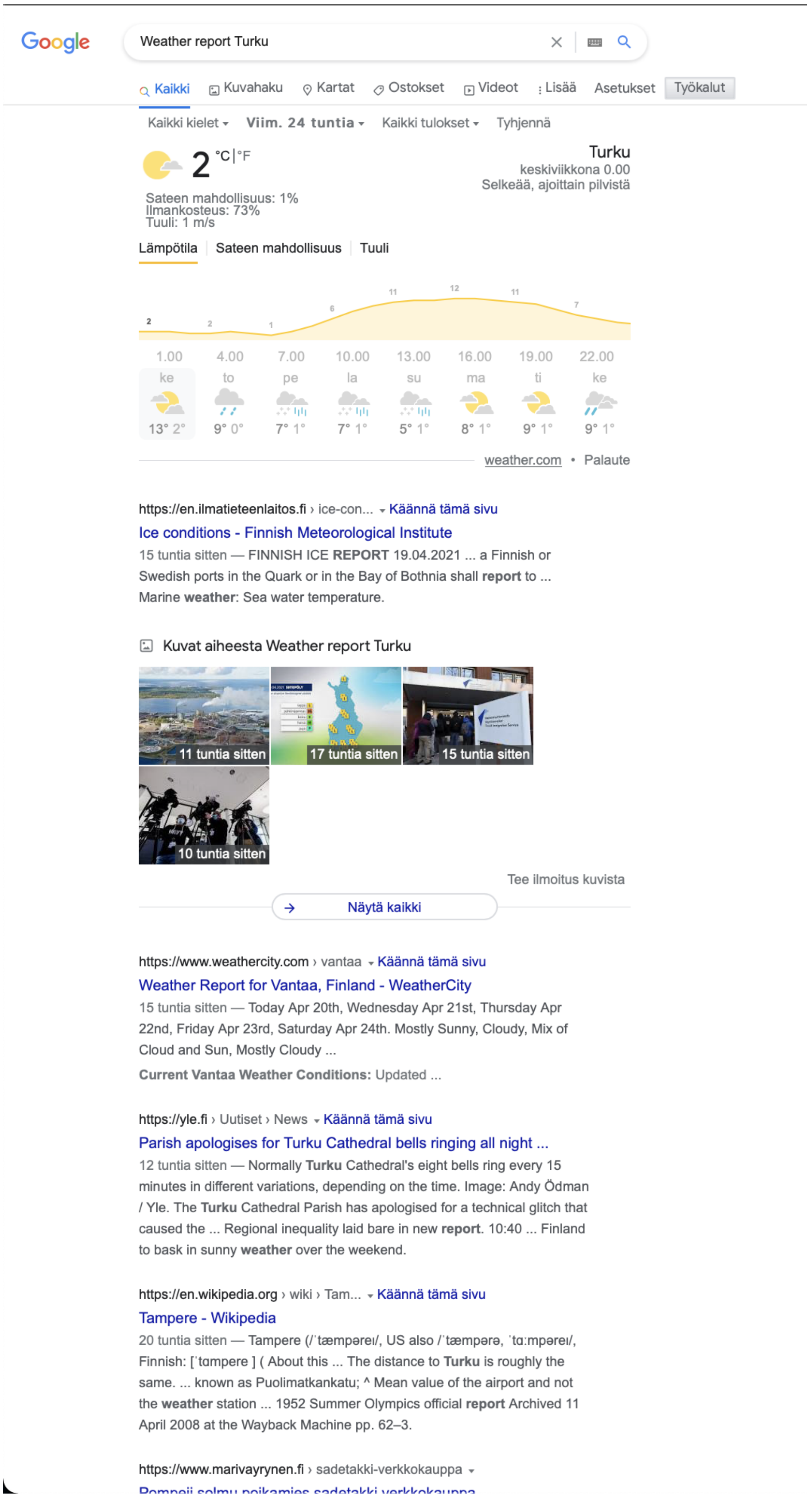
Näytä kaikki

ell.stackexchange.com › questions ▾ Käännä tämä sivu
"What is the weather today?" or "How is the weather today ...
25.7.2018 — James: No, I mean what's it **like**? Warm, sunny, rainy ...? Phil: It's hot, humid, and totally miserable. James: How's the **weather** out there?
5 vastausta · 23 ääntä: Both can be fine. While the first focuses more on...

uploads.mktg.learning.amplify.com › k-es_what_i... ▾ PDF
What Is the Weather Like Today? - Amplify
What Is the Weather Like Today? by Kate Donaldson-Fletcher illustrated by Jeffrey Ebbeler. Page 4. © 2018 ...

www.weatherbug.com › 10-day-weat... ▾ Käännä tämä sivu
10-Day Weather Forecasts & Weekend Weather | WeatherBug
Plan you week with the help of our 10-day **weather** forecasts and weekend **weather** predictions. ... Pretty Prairie, Kansas **Weather** ...
Today's Weather Outlook.

Figure 3.3: Google search results for: “wat is the whether like today” [35].



Weather report Turku

Kaikki Kuvahaku Kartat Ostokset Videot Lisää Asetukset Työkalut

Kaikki kielet Viim. 24 tuntia Kaikki tulokset Tyhjennä

2 °C|°F

Turku
keskiviikkona 0.00
Selkeää, ajoittain pilvistä

Sateen mahdollisuus: 1%
Ilmankosteus: 73%
Tuuli: 1 m/s

Lämpötila | Sateen mahdollisuus | Tuuli

2 2 1 6 11 12 11 7

1.00	4.00	7.00	10.00	13.00	16.00	19.00	22.00
ke	to	pe	la	su	ma	ti	ke
13° 2°	9° 0°	7° 1°	7° 1°	5° 1°	8° 1°	9° 1°	9° 1°

weather.com • Palaute

https://en.ilmatieteenlaitos.fi › ice-con... ▾ Käännä tämä sivu
Ice conditions - Finnish Meteorological Institute
15 tuntia sitten — FINNISH ICE **REPORT** 19.04.2021 ... a Finnish or Swedish ports in the Quark or in the Bay of Bothnia shall **report** to ... Marine **weather**: Sea water temperature.

Kuvat aiheesta Weather report Turku

11 tuntia sitten · 17 tuntia sitten · 15 tuntia sitten · 10 tuntia sitten

Tee ilmoitus kuvista

Näytä kaikki

https://www.weathercity.com › vantaa ▾ Käännä tämä sivu
Weather Report for Vantaa, Finland - WeatherCity
15 tuntia sitten — Today Apr 20th, Wednesday Apr 21st, Thursday Apr 22nd, Friday Apr 23rd, Saturday Apr 24th. Mostly Sunny, Cloudy, Mix of Cloud and Sun, Mostly Cloudy ...
Current Vantaa Weather Conditions: Updated ...

https://yle.fi › Uutiset › News ▾ Käännä tämä sivu
Parish apologises for Turku Cathedral bells ringing all night ...
12 tuntia sitten — Normally **Turku** Cathedral's eight bells ring every 15 minutes in different variations, depending on the time. Image: Andy Ödman / Yle. The **Turku** Cathedral Parish has apologised for a technical glitch that caused the ... Regional inequality laid bare in new **report**. 10:40 ... Finland to bask in sunny **weather** over the weekend.

https://en.wikipedia.org › wiki › Tam... ▾ Käännä tämä sivu
Tampere - Wikipedia
20 tuntia sitten — Tampere (/ˈtæmpəreɪ/, US also /ˈtæmpərə, ˈtɑːmpəreɪ/, Finnish: [ˈtɑmpere] (About this ... The distance to **Turku** is roughly the same. ... known as Puolimatkankatu; ^ Mean value of the airport and not the **weather** station ... 1952 Summer Olympics official **report** Archived 11 April 2008 at the Wayback Machine pp. 62–3.

https://www.marivayrynen.fi › sadetakki-verkkokauppa ▾

Figure 3.4: Google search Results for: "Weather report Turku", filtering results to the past 24 hours [35].

3.3 PageRank

PageRank is the core algorithm in Google Search. It was patented in 1998 and has been used in the search engine from the start. The patent [5], which is titled *Method for node ranking in a linked database*, tells that it was developed as a means to rank web pages more accurately based on their quality and usefulness. Previous search engine algorithms focused mostly on the textual contents of pages which resulted in often low quality search results and allowed spam websites to creep into the results.

The patent explains that this is not the first time that links have been used to determine relevancy of websites in search engines since the Hyperlink Search Engine used them as well. Hyperlink Search Engine (also knows as RankDex) used the description texts of links that point to a site to determine what that site's content is about. These type of inbound links are also called as backlinks. The idea of this was that if a site tries to rank for a certain keyword or topic, but all other sites that link to it use completely different words to describe the link, the site should probably not rank that high for the topic it tries to. And, on the other hand if a website tries to rank for a keyword and that same keyword appears in a lot of other websites' link descriptions on links to that page, the site probably is a good search result for that keyword.

PageRank takes a different approach to using links for ranking pages. Instead of using the textual descriptions of links, it uses the count of links to measure relevancy and authoritativeness. The patent first introduces a naïve approach for doing this: The rank r of a page A could just be the number of backlinks n that the page has [5]:

$$r(A) = n \tag{3.1}$$

This naïve method is referred as "citation rank" in the patent. Citation rank has an obvious flaw though since not all links are equally important. If a certain page has hundreds of links from the page's owner's other webpages, which themselves have no backlinks at all, and no links from anywhere else, it probably should not be looked at to be more relevant than another page that is linked to only from a few pages, where each of the linking pages have a lot of backlinks themselves. This is the basic idea behind the PageRank patent: The rank of a webpage should be the function of the ranks of all the pages that link to it. More formally the formula for a page's rank is defined to be [5]:

$$r(A) = \frac{\alpha}{N} + (1-\alpha)\left(\frac{r(B_1)}{|B_1|} + \ldots + \frac{r(B_n)}{|B_n|}\right) \tag{3.2}$$

The recursiveness of the function can be seen immediately, since r appears in the definition of r. In the above equation N is the amount of pages in the network, B_1 to B_n are all the backlinks of page A and $|B_1|$ to $|B_n|$ are the number of forward links on those pages. This clearly shows that backlinks increase the rank of a page and forward links in a sense pass on rank to other pages. The final component in the formula is α which acts as a damping factor. The patent explains that the damping factor is meant to model the probability of a user navigating away from the original page to any random page without clicking a link. It also states that the value for α is usually selected to be in the realm of 0.1 to 0.15. All of this results in that the rank of any given page can be though to be the probability that a user ends up on that page after navigating through many links. This probabilistic model also results in the conclusion that the total rank of all pages on the network must sum up to 1.

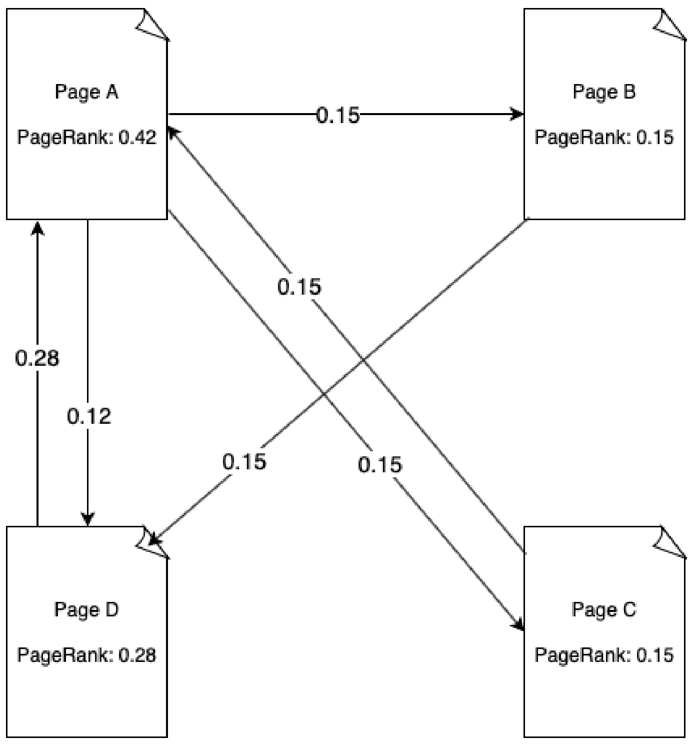


Figure 3.5: Visualizing PageRank in a four-page network.[1]

It is quite simple to use the above definition to calculate the PageRank of all pages in a small network. In the example shown in figure 3.5 we have four pages, and we can use the Python[2] program show in appendix A to calculate the ranks of all pages to be as following (using $\alpha = 0.1$):

- *Page A*: $\frac{0.1}{4} + (1 - 0.1) \cdot \frac{r(C)}{1} \approx 0.42$
- *Page B*: $\frac{0.1}{4} + (1 - 0.1) \cdot \frac{r(A)}{3} \approx 0.15$
- *Page C*: $\frac{0.1}{4} + (1 - 0.1) \cdot \frac{r(A)}{3} \approx 0.15$
- *Page D*: $\frac{0.1}{4} + (1 - 0.1) \cdot \left(\frac{r(A)}{3} + \frac{r(B)}{1}\right) \approx 0.28$

[1]The values passed between pages in figure 3.5 do not always seem to exactly add up to the pages' ranks because all values have been rounded to two decimal places.

[2]The program was run with Python 3.9.

Additionally, the patent also states that the algorithm can be fine-tuned to give more weight on certain kinds of links. For example links that appear at the start of pages or links that are shown with a bigger font can be more relevant from a human's perspective. Also, the age of the linking pages could be taken into account, since more recently updated content is less likely to be irrelevant. Google has also a second PageRank related patent, which has not yet expired like the original one, the new patent [47] was granted in 2015, and it is titled *Producing a ranking for pages using distances in a web-link graph.* All kinds of fine-tuning has probably been done at Google to the methods introduced in the patents, and it is hard to say how much of them is in use today. Nevertheless, one could assume that the core principles are likely still there in some form.

4 Case Skole

The practical part of this thesis will focus on a case study where the theories and guidelines of search engine optimization will be put into use in Skole. Skole is a forum where students can discuss their studies, get help from each other, and improve each other's learning as a community. Skole can be used through either a web or a mobile application, but this thesis will mostly focus on the web application since search engines are not interested in the contents of native mobile applications. It is important to note though that the native application behaves almost identically as the web version does with a mobile browser. Skole's initial version was launched in December 2020, and a new greatly overhauled version “Skole 2.0” in April 2021. Any implementations that need to be done in practical part should be easy to get into the application, since Skole is run by a small and agile, three-person team which has no hierarchy or management overhead that might slow down changes in larger organizations.

4.1 Research Method

The case study will aim to answer all three of the research questions. The research method will primarily be to implement and go over technical examples with guidance from the supporting literature. The results will mainly be measured by documenting how the application's pages show up in Google before and after changes have been made. Answering *RQ3*, which is about the importance of search engine legibility in an early stage web service, will be covered with data from Google's Search Console and data from Simple Analytics, the chosen on-site analytics provider for the service.

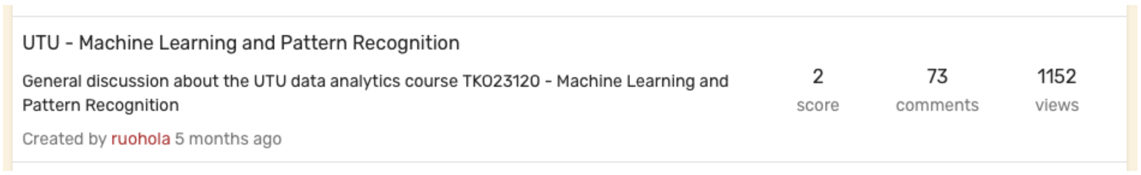
4.2 Introduction of the Application

Skole is an application that offers a platform for higher education students to discuss any topics related to for example their studies or course work. Skole is usable with any modern desktop or mobile browsers at `www.skoleapp.com`, and can be downloaded as an app from the Apple App Store and Google Play Store.

Skole's use is currently[1] limited to only Finnish universities, and when registering an account one must provide a valid Finnish university email address. This is done for mainly two reasons: Firstly a valid student email is required so that people interacting on the platform can be sure that other users are also legitimately students, secondly this makes moderating simple when unwanted users can be banned very effectively from the platform since getting a new university email address is not that easy. The allowed email domains are limited to just the 14 Finnish universities [48], instead of all higher education facilities, just so that the initial target demographic stays focused enough. In addition to requiring a university email, the registration also currently requires an invitation code that can be gotten from an existing user of Skole. The code is required to limit the initial user population which aims to make testing and experimenting with new features easier.

The latest version of Skole has two core means for user interaction. These are creating threads and posting comments. A thread is essentially a discussion opener. When creating a thread the creator can choose a title, add some text, and optionally add an image to it.

[1]As of August 2021

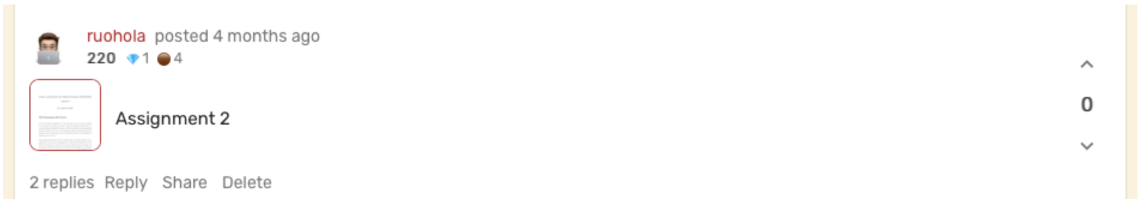

UTU - Machine Learning and Pattern Recognition

General discussion about the UTU data analytics course TKO23120 - Machine Learning and Pattern Recognition

Created by ruohola 5 months ago

Figure 4.1: Preview of a thread in Skole [48].

When a thread has been created other users and the original creator themselves can then add comments to the thread and reply to other posted comments. Comments can contain text and in addition an image or a file can be added to them as an attachment.


ruohola posted 4 months ago
220 1 4

Assignment 2

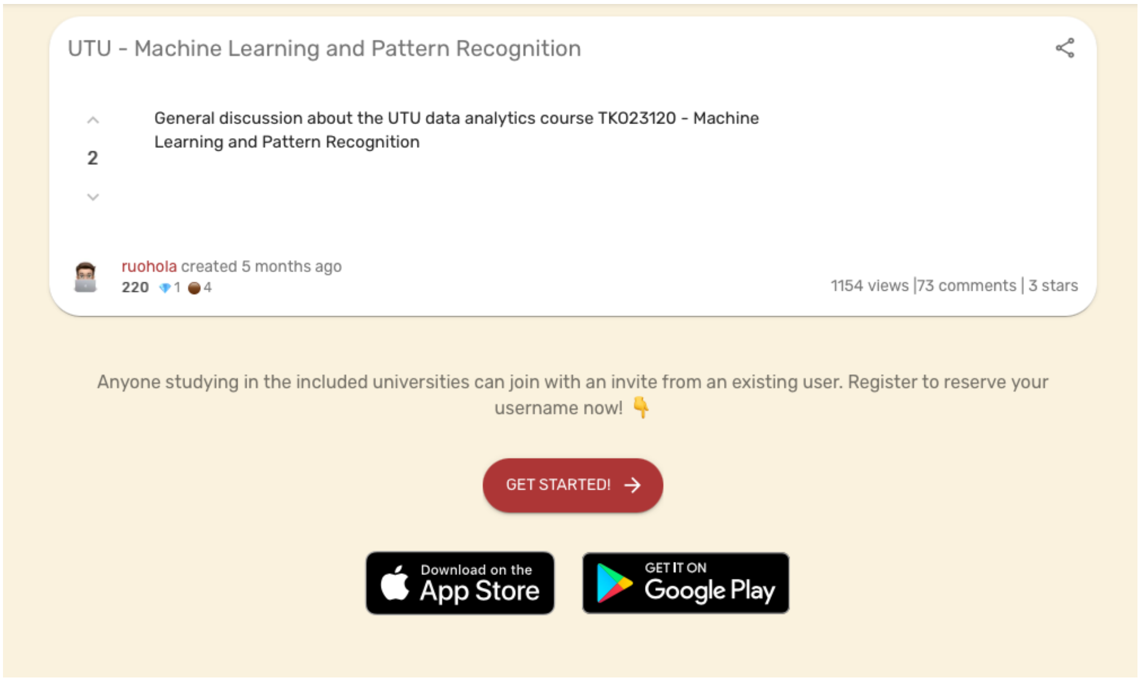
Figure 4.2: A comment with an attached file in Skole [48].

The relevancy of content is determined on mostly how new it is and how much score it has received from users. Score in Skole is means to measure the value of threads and comments. When for example a comment is created, other users can vote it or down. Each vote increases or decreases the score of that comment by one. The exact algorithm behind determining the relevancy of the content uses more factors than just the score, but score does play a big part in it.

4.3 User Experience

Like most modern web services, Skole also tries to be as user-friendly as possible in every way. Providing a good user experience is key part in getting users to like interacting with the application and thus making them come back to it. This is especially important in an application that is dependent on an active community of users who post content for other users to consume.

The user experience starts with getting in to the application. Most of the actual content in Skole is only visible to logged-in users which immediately raises the barrier of entry and can thus be though to hinder the user experience of potential users. Unauthenticated users can access all the static pages which include basic information such as the terms and the privacy policy, and a guide about the scoring system. In addition to those, unauthenticated users can access the profile pages of other users and pages of individual threads, but they cannot see the thread list. They also cannot see the threads or comments that a user has created, or the comments in a thread page. Even though this might make the experience of unauthenticated users worse, this is done to enhance the privacy of the content posted by authenticated users and seems justified for it.



UTU - Machine Learning and Pattern Recognition

2

General discussion about the UTU data analytics course TKO23120 - Machine Learning and Pattern Recognition

ruohola created 5 months ago
220 1 4

1154 views |73 comments | 3 stars

Anyone studying in the included universities can join with an invite from an existing user. Register to reserve your username now! 👇

Figure 4.3: Thread comments are not shown for unauthenticated users in Skole [48].

Anonymity has been taken into account in Skole. Users do not need to appear by their real names, and the names of people are not even asked during the registration.

User's can appear either with their chosen username or completely anonymously. This aims to keep the participation threshold low and to allow users to more freely express their thoughts. The application also collects very little data of its users, and the only actual identifiable information that is collected is the user's email address.

The user interface (UI) of Skole aims to be simple and intuitive to use. The UI is created to be responsive, which means that it scales well for devices with different screen sizes. Responsiveness of the UI aims to make using Skole on mobile devices as easy and pleasant as on a desktop.

4.4 Technical Specs

Skole is a web and mobile application. The backend of the application has been built using Python and the popular web framework Django. The backend uses the relational database management system (RDBMS) PostgreSQL as its data storage. Django was chosen as the backend framework because of the founders' familiarity with it and with Python in general. The only real other option for the RDBMS would have been MySQL, but PostgreSQL was chosen because of its more extensive feature set.

Django offers a templating system which allows creating websites without a separate frontend application. Building the whole application with Django can have the benefit of not needing to develop and manage a whole other code base for the frontend. Nevertheless, there are a lot of benefits in separating the frontend: It allows scaling the frontend and backend individually, if one is experiencing a more heavy load than the other, and writing the frontend with JavaScript arguably makes development faster while allowing to more easily add dynamic content to pages. This is why the user-facing part of the application is built with Next.js. Next.js is web framework which is build on top of the React JavaScript library. Its distinctive features include static site generation (SSG) and server-side rendering (SSR) of web pages [49]. SSR and SSG are both used in Skole. The UI is implemented using React components from the Material-UI library as the basic building blocks. Material-UI was chosen since it offers clean, uniform, and modern looking components which are relatively easy to use to make good-looking responsive layouts.

SSG and SSR are useful features to have when considering the search engine legibility of a site. Even though Googlebot is able to render and understand JavaScript [50], rendering all or some of the content already on the server reduces load times and can make sure that the crawlers sees a more complete and correct looking version of the site [51]. Google's documentation [50] also states that not all bots are able to execute JavaScript content at all.

All the aforementioned technologies are free and open-source, as is the standard nowadays. Utilizing open-source tools reduces costs and vendor-locking, both of which can be very harmful in a project of any size. A big part in choosing the technologies was also their ability to provide a fast development time. Being able to make changes and add new features quickly is crucial when wanting to provide a good user experience while constantly listening on user feedback and iterating based on it.

The application itself is hosted in Amazon Web Services's (AWS) Amazon Elastic Container Service (ECS). AWS was chosen as the cloud providers as it is the industry leader [52], and thus it was likely that its features and services would always be more than enough for application, even if the user base would grow a lot. In addition to ECS, Skole utilizes many other AWS services as well, including for example Amazon Relational Database Service (RDS) for hosting the PostgreSQL database, Amazon Simple Storage Service (S3) for storing static and media files, and Amazon Simple Email Service (SES) for sending transactional emails that certain events in the application trigger.

4.5 Search Engine Legibility

The perspective of search engines has not really been considered in Skole's early development processes. The design and development was done iteratively, and even though it took over a year from the first line of code to the initial launch, it still mainly focused on getting the right functionality there and having the site be attractive looking for users. This approach worked well in practice, since before the site was live and indexable by search engines, improving the search engine legibility would have been quite difficult. One issue also was that none of members of the founding team had previous experience about search engine optimization or actually about even the basic working principles of them. Nevertheless, some things which aid search engines in understanding the site have still been done — mostly coincidentally.

As was stated earlier, Skole's frontend is using SSG and SSR, both of which help in delivering more complete and easily processable HTML to search engines. Skole is built as a progressive web application (PWA), and it is developed to function equally well on a mobile browser as on a desktop browser. According to Google's documentation [53] having a good experience with mobile browsers is very beneficial in getting pages featured well in search results. Skole's pages have also used some SEO helping HTML features, such as using descriptive `<title>` elements, meta descriptions, and link texts. Quite a bit of SEO enhancements were still to be made though.

5 Improving Skole's Structure for Search Engines

This chapter will cover the practical part of this thesis and aims to answer *RQ1* and *RQ2*. One of the biggest challenges in the thesis was that as a very new application, Skole changed a lot in the time period this was written in, since it was and still is constantly developed and improved. The changes here were implemented in the time period between December 2020 to August 2021.

Skole's first version was launched on December 16, 2020. In that version, practically all content of Skole was visible to unauthenticated users. This was done because it was thought that one of the key value propositions of the platform was openness; everyone could see everything and even commenting could be done without registering an account. This also had the benefit of being able to let search engines get access to every page.

The current and greatly overhauled Skole 2.0 version was launched on April 20, 2021. 2.0 version drastically changed the user facing parts of the application, but the improvements made to the search engine legibility stayed largely the same. The biggest challenge in discoverability of the content in the Skole 2.0 is the fact that most of the content is only visible to logged-in users. This obviously hinders the discoverability through search engines since they have access to the exact same limited parts of Skole that unauthenticated users have.

5.1 Allowing Google to Find Skole's Pages

To let Google know about a website they advise [54] to add it as a website property in Google Search Console. As advised, this was done in Skole a few days before launching it. In practice this happened by adding a specific DNS record to the domain. This record is shown in Google Search Console, and was added with Terraform to AWS Route 53, where Skole's domain is managed.

Right after the launch of the first version of Skole, Google did not find almost any of the site's pages. One thought was that it could have been caused by the fact that search engines work best with fully server-side rendered content where all the HTML is returned to the client pre-rendered and its already viewable as is. Skole does use server-side rendering in a few places, but mostly it utilizes Next.js's static site generation. In SSG most of the page is rendered during build time and then the rest is rendered on the client after all the necessary dynamic data has been fetched from the backend server. The reason SSG was initially selected was because of its benefits in reducing the load experienced by the frontend's web server, so it was not really an option to move away from it. It is hard to say if full SSR would have actually helped at all, but since another method was anyways needed to let Google more easily find pages of the site, a sitemap was added.

A sitemap can simply be an XML file which lists all the pages that the site has. In Skole the sitemap is generated dynamically when requested, so that in addition to containing all the static pages of the site, it also fetches data from the backend server to get all the automatically generated slugs for all dynamically created pages. Currently, these dynamic pages include just the `/threads/<slug>` and `/users/<slug>` URLs.

To accompany the created sitemap, a `robots.txt` file was also added to the root of the site. The file is very simple, and it just points to the created sitemap:

```
User-agent: *
Sitemap: https://www.skoleapp.com/sitemap.xml
```

The `robots.txt` file was simply implemented as a fully server-side rendered page in Next.js, where the contents of the file was written straight into the response's body:

```
import { GetServerSideProps } from 'next';

export const getServerSideProps: GetServerSideProps = async ({ res }) => {
  const robots =
    'User-agent: *\nSitemap: https://www.skoleapp.com/sitemap.xml';

  res.setHeader('Content-Type', 'text/plain');
  res.write(robots);
  res.end();

  return {
    props: {},
  };
};

const RobotsTxt = (): void => {};

export default RobotsTxt;
```

The sitemap page uses a very similar technique to return the XML contents but as stated, it does require some additional logic to be able to render entries for the dynamic pages. Here is a fragment taken from the beginning of Skole's current sitemap [55]:

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset
  xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:xhtml="http://www.w3.org/1999/xhtml"
  >
  <url>
    <loc>https://www.skoleapp.com</loc>
    <lastmod>2021-07-01</lastmod>
    <xhtml:link
```

```
      rel="alternate"
      hreflang="en"
      href="https://www.skoleapp.com"
      />
    <xhtml:link
      rel="alternate"
      hreflang="fi"
      href="https://www.skoleapp.com/fi"
      />
    <xhtml:link
      rel="alternate"
      hreflang="sv"
      href="https://www.skoleapp.com/sv"
      />
    <xhtml:link
      rel="alternate"
      hreflang="x-default"
      href="https://www.skoleapp.com"
      />
  </url>
```

Here the `rel` and `hreflang` attributes are used in the tag to indicate that the default language of the page is English and that it also has alternative Finnish and Swedish versions available [56].

5.2 Allowing Google to More Easily Process Skole's Pages

The user interface of Skole has a lot of elements that can be clicked. Initially they were mostly made of Material-UI's `Button` elements, which themselves are wrappers for HTML `<button>` tags. This was a reasonably logical choice and worked fine for most of the elements. A problem appeared in elements which navigated away from the current page and changed the URL. With those kinds of buttons one was not able to open the linked page in a new tab by for example clicking the button with the middle mouse button. A bigger issue in links made from buttons was that Google does not understand them [57] and cannot follow them to find the pages that the buttons link to. These problems were addressed by replacing all the button elements with Material-UI's `MaterialLink`s, which are anchor (`<a>`) tags and have the proper `href` attributes that Google needs [57].

As was stated previously, when Skole launched its pages already had good titles and meta descriptions. Google still did not use those to form the search results snippets, but instead it used the landing page's title and description values for all pages. The issue was that the meta descriptions and titles were not available in the pre-rendered HTML of the loading screen. The loading screen was always initially served to any crawler or user that accessed the site, and only after some necessary backend data-populating requests had completed, the content of the actual page would be rendered on the client. This issue was fixed by passing the page title and meta description as props to the component for the loading screen and then making sure that it renders them to the HTML's `<head>`. It is hard to say why the values of the meta tags in the initially served pre-rendered page were the ones shown in Google results, since Googlebot should be able to execute JavaScript just fine [50]. Perhaps the issue was that the tags were already there in the initially served HTML and Google just took those values into account, instead of waiting for the page to fully render and only then reading the tags' contents.

Initially Skole's frontend translations were designed so that the language of the current user was stored only in the user's browser's local storage. This worked well and was convenient for the user; no matter where a user would find and click open a link to Skole, the page would automatically open in their chosen language. This had to be changed though while fixing the title and description issue. If the language choice would only live in the browser, there would be no way to have the correct language available on the pre-rendering phase. Luckily the translation library that was in use (next-translate) supports activating the language by having it at the start of the URL. So previously when the URL to for example the privacy policy would have been `skoleapp.com/privacy` it would now be `skoleapp/privacy` when accessing the English language version and `skoleapp.com/fi/privacy` when accessing the Finnish language version of the page. This change also allows Google to index all the different translated versions of the pages separately, which may aid in showing more localized content for users who see Skole's pages in Google results.

Other useful tags that have been added to Skole's non-user-facing HTML are Open Graph meta tags. Open Graph is a protocol that allows a website to define specific kind of metadata which other applications can consume when interacting with the site. Open Graph was originally developed by Facebook, and it enables developers to make web pages that when linked in Facebook posts feel like they are move heavily integrated into the social platform. [58] In addition to improving the look of Facebook posts that link the site, specifying these tags also improves link previews on various other services, such as WhatsApp and Twitter [59]. Skole's rendered `<head>` element contained the following Open Graph tags:

```html
<meta property="og:title" content="Skole" />
<meta
  property="og:description"
  content="Skole is a next-gen study forum ..."
/>
<meta property="og:type" content="website" />
<meta property="og:site_name" content="Skole" />
```

These were added a long time ago, and no greater though had been put into them. After a closer examination it could be recognized that they were incomplete. The Open Graph documentation [58] states that `og:image` and `og:url` tags should be defined as well, but they were not defined in Skole. The missing tags were of course quite simple to just add in:

```html
<meta property="og:title" content="Skole" />
<meta
  property="og:description"
  content="Skole is a next-gen study forum ..."
```

```
/>
<meta property="og:type" content="website" />
<meta property="og:site_name" content="Skole" />
<meta property="og:url" content="https://www.skoleapp.com" />
<meta
  property="og:image"
  content="https://www.skoleapp.com/images/icons/skole-icon-og.png"
/>
```

A more challenging thing that was noticed to be problematic with Skole's usage of Open Graph meta tags, was the fact that they were not working at all on the dynamically created pages. On dynamically created thread and user pages the `og:title`, `og:description`, and `og:image` meta tags should contain details about that specific object that the page is for, and the tags should be unique to each dynamic page. One thing that also added complexity here was that unlike Googlebot, Facebook's Open Graph crawler does not execute any client-side JavaScript [60]. This meant that unlike the static pages (like the landing page and privacy policy page) which used `getStaticProps` to utilize Next.js's static site generation, the dynamically created pages had to now be rendered on the server for every request [61]. This was accomplished by fetching the object data, corresponding to the requested slug, from the backend server in the `getServerSideProps` function:

```
export const getServerSideProps: GetServerSideProps = async ({
  query,
  locale,
}) => {
  const { slug } = query;
  const apolloClient = initApolloClient();

  const { data, error } = await apolloClient.query({
    query: ThreadDocument,
    variables: { slug },
  });

  const notFound = !data?.thread && !error;

  return {
    props: {
      _ns: await loadNamespaces(['thread', 'thread-tooltips'], locale),
      threadData: data || null,
      threadError: error || null,
    },
    notFound,
  };
};
```

If the `notFound` property in the returned object is `true`, Next.js automatically knows to return the `404.tsx` page [61] for requests that try to access a thread or a user profile with a slug that does not belong to any object in the database. Previously this checking was done manually in the page components' client-side code, which had the problem that if a crawler would for example access a deleted page, it would still get a 200 status code in the response and might not realize that the page should now be deindexed. After[1] the Open Graph fix was done, any non-existing pages will now properly return a 404 status code.

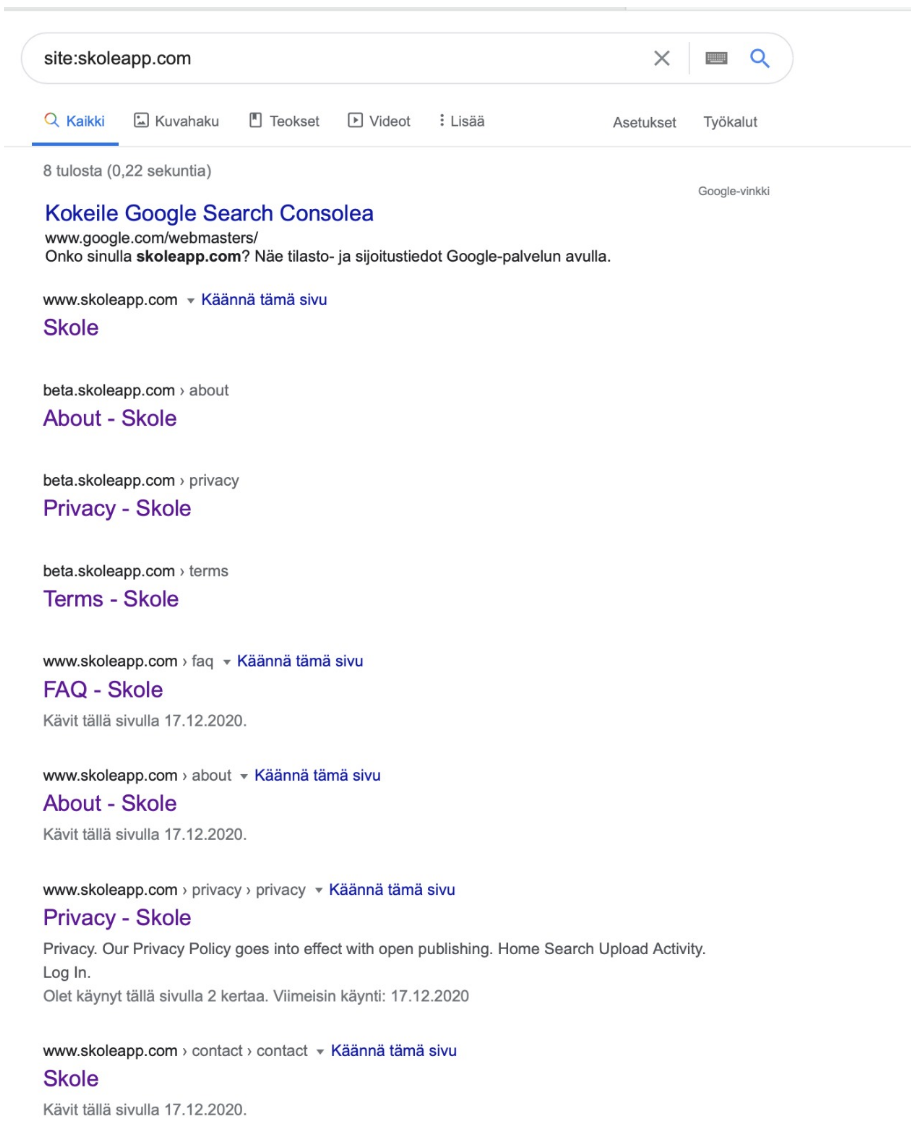
[1]In August 2021

6 Analysis and Results

When the original version of Skole launched in December 2020, its presence in Google was atrociously bad (which can be seen from figure 6.1). Google was not able to find almost any of the site's pages and for those few that it did find, it still was not able to generate meaningful snippets for.

After adding a `sitemap.xml` to the site, Google immediately found quite a lot of new pages. Figure 6.2 shows that the titles and descriptions of these new pages shown in the search results were all duplicates of each other. This was caused by an issue where the correct content for the title and meta description tags was only available on the client, and the incorrect contents of the tags in the server-side rendered page were consumed by Googlebot. This resulted in Google seeing the title and meta description from the index page on all pages.

After the duplicate snippet issue was fixed, the results started to look better. In figure 6.3 Google has found a lot more pages. It is important to note though that it has been stated in Google's support documentation [62] that the search count that is shown in the top of the results is an estimate and should not be relied on too much.


site:skoleapp.com

8 tulosta (0,22 sekuntia)

Google-vinkki

Kokeile Google Search Consolea
www.google.com/webmasters/
Onko sinulla **skoleapp.com**? Näe tilasto- ja sijoitustiedot Google-palvelun avulla.

www.skoleapp.com ▾ Käännä tämä sivu
Skole

beta.skoleapp.com › about
About - Skole

beta.skoleapp.com › privacy
Privacy - Skole

beta.skoleapp.com › terms
Terms - Skole

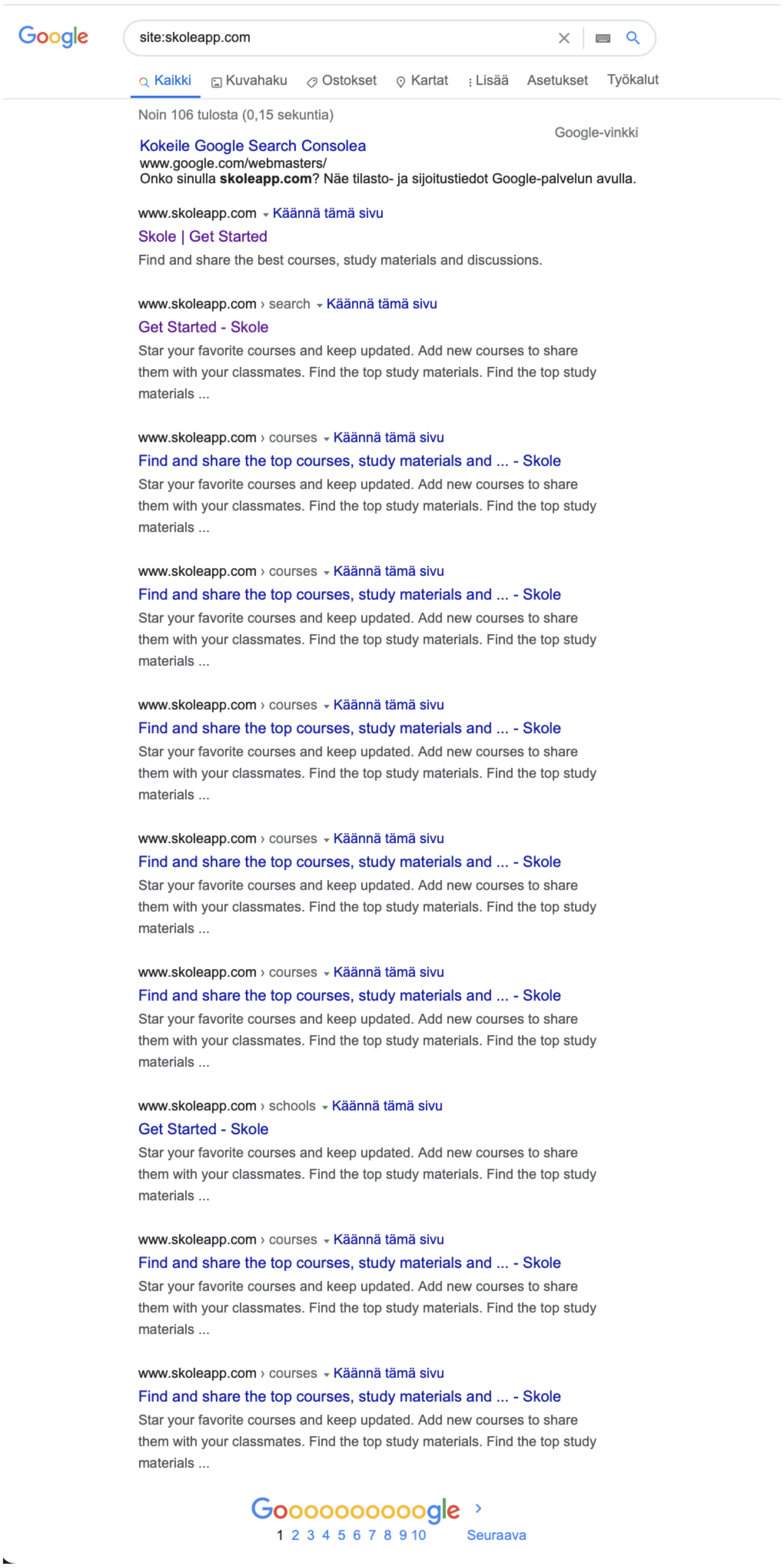
www.skoleapp.com › faq ▾ Käännä tämä sivu
FAQ - Skole
Kävit tällä sivulla 17.12.2020.

www.skoleapp.com › about ▾ Käännä tämä sivu
About - Skole
Kävit tällä sivulla 17.12.2020.

www.skoleapp.com › privacy › privacy ▾ Käännä tämä sivu
Privacy - Skole
Privacy. Our Privacy Policy goes into effect with open publishing. Home Search Upload Activity. Log In.
Olet käynyt tällä sivulla 2 kertaa. Viimeisin käynti: 17.12.2020

www.skoleapp.com › contact › contact ▾ Käännä tämä sivu
Skole
Kävit tällä sivulla 17.12.2020.

Figure 6.1: Google search results for `skoleapp.com` in 2020-12-18.



Noin 106 tulosta (0,15 sekuntia)

Google-vinkki

Kokeile Google Search Consolea
www.google.com/webmasters/
Onko sinulla **skoleapp.com**? Näe tilasto- ja sijoitustiedot Google-palvelun avulla.

www.skoleapp.com ▾ Käännä tämä sivu
Skole | Get Started
Find and share the best courses, study materials and discussions.

www.skoleapp.com › search ▾ Käännä tämä sivu
Get Started - Skole
Star your favorite courses and keep updated. Add new courses to share them with your classmates. Find the top study materials. Find the top study materials ...

www.skoleapp.com › courses ▾ Käännä tämä sivu
Find and share the top courses, study materials and ... - Skole
Star your favorite courses and keep updated. Add new courses to share them with your classmates. Find the top study materials. Find the top study materials ...

www.skoleapp.com › courses ▾ Käännä tämä sivu
Find and share the top courses, study materials and ... - Skole
Star your favorite courses and keep updated. Add new courses to share them with your classmates. Find the top study materials. Find the top study materials ...

www.skoleapp.com › courses ▾ Käännä tämä sivu
Find and share the top courses, study materials and ... - Skole
Star your favorite courses and keep updated. Add new courses to share them with your classmates. Find the top study materials. Find the top study materials ...

www.skoleapp.com › courses ▾ Käännä tämä sivu
Find and share the top courses, study materials and ... - Skole
Star your favorite courses and keep updated. Add new courses to share them with your classmates. Find the top study materials. Find the top study materials ...

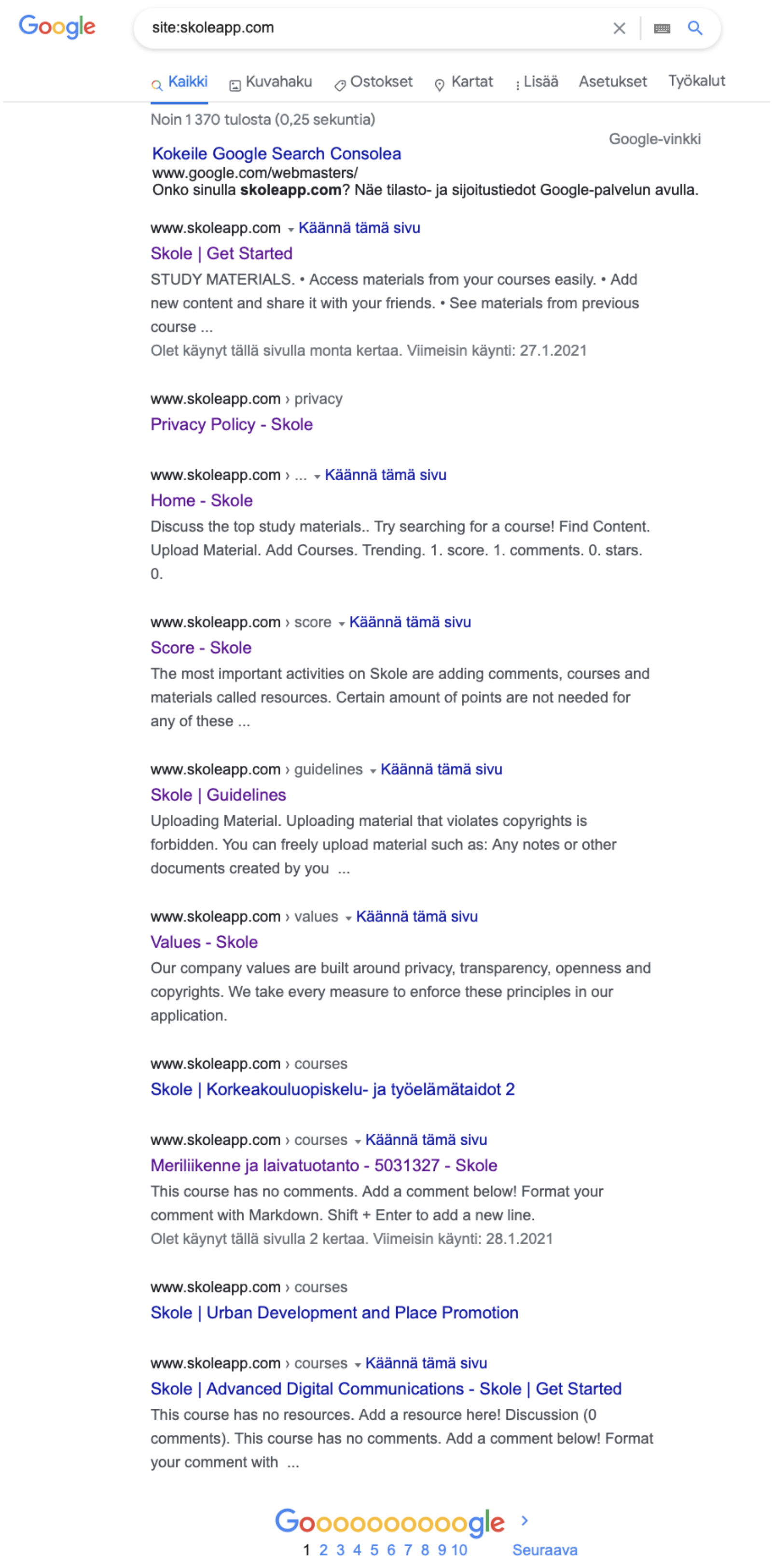
www.skoleapp.com › courses ▾ Käännä tämä sivu
Find and share the top courses, study materials and ... - Skole
Star your favorite courses and keep updated. Add new courses to share them with your classmates. Find the top study materials. Find the top study materials ...

www.skoleapp.com › schools ▾ Käännä tämä sivu
Get Started - Skole
Star your favorite courses and keep updated. Add new courses to share them with your classmates. Find the top study materials. Find the top study materials ...

www.skoleapp.com › courses ▾ Käännä tämä sivu
Find and share the top courses, study materials and ... - Skole
Star your favorite courses and keep updated. Add new courses to share them with your classmates. Find the top study materials. Find the top study materials ...

www.skoleapp.com › courses ▾ Käännä tämä sivu
Find and share the top courses, study materials and ... - Skole
Star your favorite courses and keep updated. Add new courses to share them with your classmates. Find the top study materials. Find the top study materials ...

Figure 6.2: Search results in 2021-01-15, after adding a sitemap.



Noin 1 370 tulosta (0,25 sekuntia)

Google-vinkki

Kokeile Google Search Consolea
www.google.com/webmasters/
Onko sinulla **skoleapp.com**? Näe tilasto- ja sijoitustiedot Google-palvelun avulla.

www.skoleapp.com ▾ Käännä tämä sivu
Skole | Get Started
STUDY MATERIALS. • Access materials from your courses easily. • Add new content and share it with your friends. • See materials from previous course ...
Olet käynyt tällä sivulla monta kertaa. Viimeisin käynti: 27.1.2021

www.skoleapp.com › privacy
Privacy Policy - Skole

www.skoleapp.com › ... ▾ Käännä tämä sivu
Home - Skole
Discuss the top study materials.. Try searching for a course! Find Content. Upload Material. Add Courses. Trending. 1. score. 1. comments. 0. stars. 0.

www.skoleapp.com › score ▾ Käännä tämä sivu
Score - Skole
The most important activities on Skole are adding comments, courses and materials called resources. Certain amount of points are not needed for any of these ...

www.skoleapp.com › guidelines ▾ Käännä tämä sivu
Skole | Guidelines
Uploading Material. Uploading material that violates copyrights is forbidden. You can freely upload material such as: Any notes or other documents created by you ...

www.skoleapp.com › values ▾ Käännä tämä sivu
Values - Skole
Our company values are built around privacy, transparency, openness and copyrights. We take every measure to enforce these principles in our application.

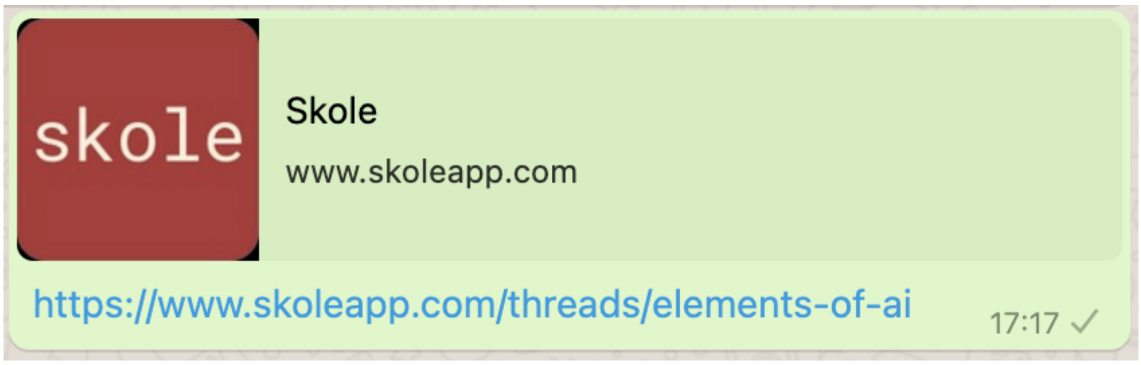
www.skoleapp.com › courses
Skole | Korkeakouluopiskelu- ja työelämätaidot 2

www.skoleapp.com › courses ▾ Käännä tämä sivu
Meriliikenne ja laivatuotanto - 5031327 - Skole
This course has no comments. Add a comment below! Format your comment with Markdown. Shift + Enter to add a new line.
Olet käynyt tällä sivulla 2 kertaa. Viimeisin käynti: 28.1.2021

www.skoleapp.com › courses
Skole | Urban Development and Place Promotion

www.skoleapp.com › courses ▾ Käännä tämä sivu
Skole | Advanced Digital Communications - Skole | Get Started
This course has no resources. Add a resource here! Discussion (0 comments). This course has no comments. Add a comment below! Format your comment with ...

Figure 6.3: Search results in 2021-01-29, after fixing duplicate snippets.

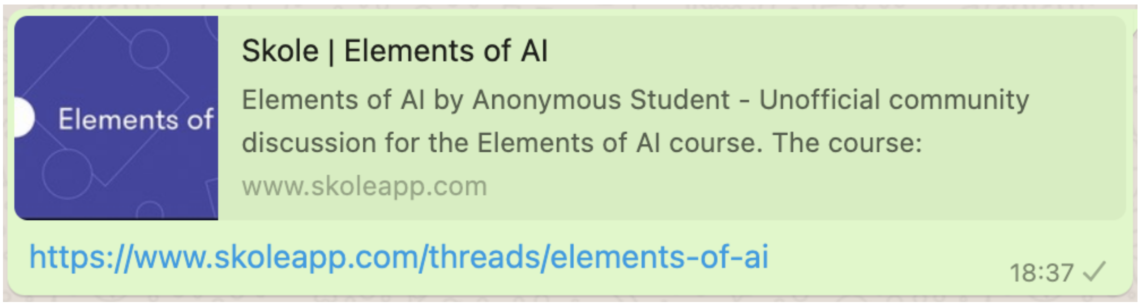

skole

Skole

www.skoleapp.com

https://www.skoleapp.com/threads/elements-of-ai 17:17 ✓

Figure 6.4: Sharing a link from Skole, before Open Graph protocol was correctly implemented.


Elements of

Skole | Elements of AI

Elements of AI by Anonymous Student - Unofficial community discussion for the Elements of AI course. The course:

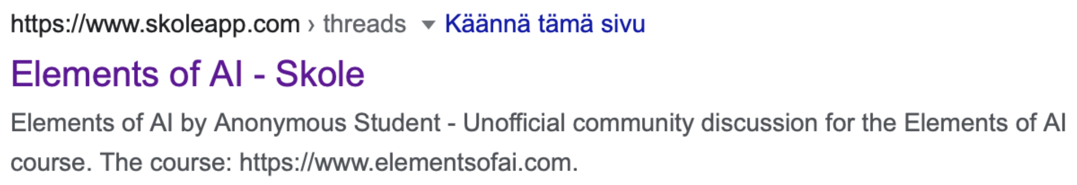
www.skoleapp.com

https://www.skoleapp.com/threads/elements-of-ai 18:37 ✓

Figure 6.5: Sharing a link from Skole, after Open Graph protocol was correctly implemented.

While not purely related to search engines, sharing content outside the platform is also very important to any modern web service. Before the Open Graph improvements were made for Skole, sharing a link in for example WhatsApp would result in a relatively uninteresting preview, like shown in figure 6.4. After all the enhancements to those were made, the sharing preview got a lot more enticing, as seen in figure 6.5.
The Open Graph fixes done for dynamic pages also improved the previews of those pages in Google results. Google search results preview for the same thread that was shown in the sharing figures, is shown in figure 6.6.

What was found out during this time is that search engine legibility of this kind of very early stage web service was actually quite a lot less important in the end than


https://www.skoleapp.com › threads ▾ Käännä tämä sivu

Elements of AI - Skole

Elements of AI by Anonymous Student - Unofficial community discussion for the Elements of AI course. The course: https://www.elementsofai.com.

Figure 6.6: Preview of a Skole thread page in Google search results.

it was initially though out to be. Much more important is the core functionality of the service, and the fact that its usability and value proposition are so good that the users come back to it over and over again. Being highly visible in search engines of course helps new users find the site if they have not heard about it before, but it does not help at all in retaining them by getting them to revisit the site. It was found out that retention can basically only be increased by improving the service in ways that the users actually appreciate. One way this was practiced in Skole was by conducting user interviews, where existing users would be asked about what they would want to change or improve in the service. Improving a web service's attractiveness through user feedback is not in the scope of this thesis though.

Even though technically Skole arguably became a much better service in the several months after its launch that this thesis was conducted in, the technical improvements did not do anything to help get more users on the platform. On the contrary, Skole's usage never really started to increase, and has been stagnating since May 2021, as can be seen in figure 6.7. Very similar progress can also be seen by viewing the performance of the website in Google Search Console, as seen in figure 6.8. We cannot conclude that improving the technical SEO aspects of the site does not help in getting users on a web service, but we can conclude that these kinds of improvements are not sufficient on their own to increase the attractiveness of a service enough to acquire more users consistently.

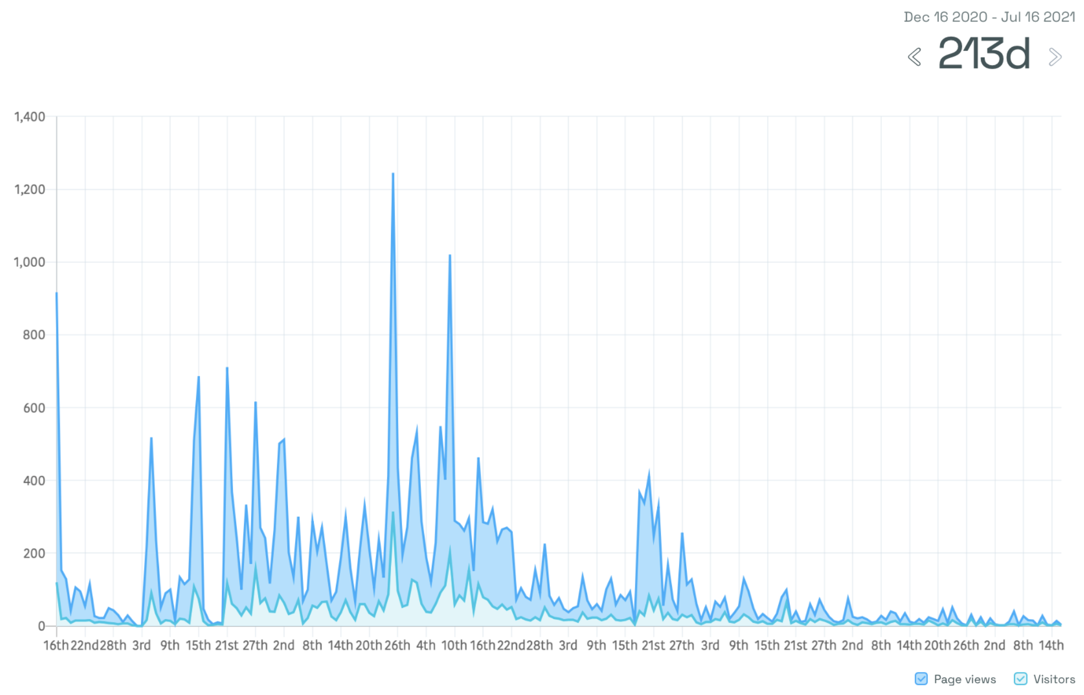


Figure 6.7: Visitors and page views for `skoleapp.com`, collected by Simple Analytics between December 16, 2020, and July 16, 2021.

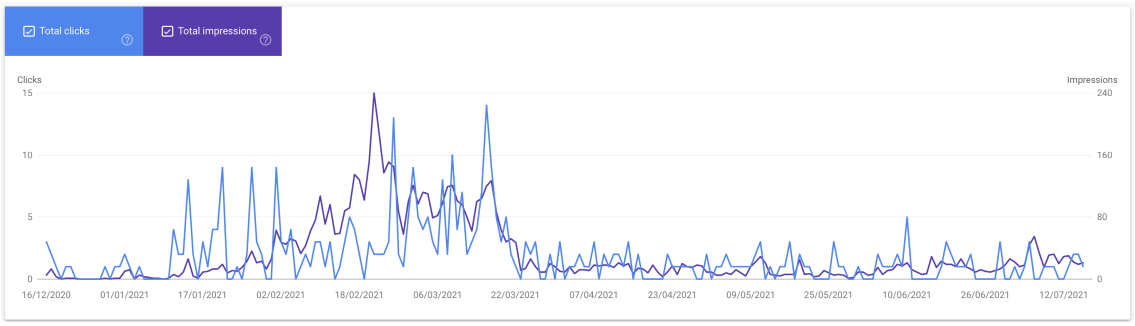


Figure 6.8: Google Search performance of `skoleapp.com` between December 16, 2020, and July 16, 2021.

7 Conclusions

The thesis has covered a cross-section of practical search engine optimization topics that are good to keep in mind when developing any public facing web service. The introduction to the topics was backed up by the overview into the general history of search engines. The thesis focused quite heavily on Google Search as it is the market leader, but most of the introduced concepts are likely applicable true for other general purpose web search engines as well. The thesis's introduction chapter presented the three research questions, and the rest of the thesis was either building up some groundwork for them or dedicated to directly answering them.

A simple concluding answer to the first research question (*How to structure the content of a website so that it is easy for search engines to read, and at the same time intuitive to use for humans?*) is that one needs to take in mind the public guidelines that for example Google Search has provided, when developing any web service which should be indexed by it. One also must remember to make all user experience affecting decisions with the end user as the number one priority and not to design anything only for search engine crawlers.

An answer to the second research question (*How the search engine legibility of an existing application can be improved in practice?*) is that a good start is to search for the site in one or more search engines and simply look at the returned pages. One should make sure that all the pages that should show up there do show, and that no unwanted pages are returned in the results. One should also check that all the search result previews look good and contain information which describes that page well. If some pages do not look representative enough, one can always refer back to the relevant documentation and do the needed improvements. This practice worked quite well in the case study. A big challenge in the case study was that it takes a lot of time for Google to reindex pages after some SEO improvements have been made to them. It often took weeks for the search result previews to change, and since manually requesting reindexing from Google Search Console was not feasible for Skole's large number of dynamically created pages, there was really no way to speed up the process. Another thing that hindered this improvement process in the case study, was the fact that when doing big overhauls in the code base, often things that were already fixed would end up breaking again. As a very early stage service, these kinds of big revamps happened a few times, and they would again set back the quality of the search results until the regression was noticed.

The third research question (*How important is search engine legibility in an early stage web service?*) was answered quite exhaustively at the end chapter six. Search engine legibility and presence is a lot less important than the actual quality and usability of the service. Making improvements to search engine legibility never does harm, but spending time on it — instead of doing user facing improvements — does not seem to be worth it.

This research could be continued by taking a more user facing approach. For example interviews could be conducted that would collect data about what users value to be the most important aspects of this kind of community based web service, and how valuable or useful is a good search engine presence from their perspective. The interviewees for this kind of study could be past, current, and potential users of Skole and other similar web services.

References

[1] M. R. Patterson, “Google and search engine market power”, *SSRN Electronic Journal*, 2012. DOI: https://doi.org/10.2139/ssrn.2047047.

[2] *The History of Search Engines.* [Online]. Available: https://carlhendy.com/history-of-search-engines (accessed 2021-01-30).

[3] T. Seymour, D. Frantsvog, and S. Kumar, “History of search engines”, *International Journal of Management & Information Systems (IJMIS)*, vol. 15, no. 4, p. 47, Sep. 2011. DOI: https://doi.org/10.19030/ijmis.v15i4.5799.

[4] O. Nierstrasz, *W3 Catalog History*, Nov. 1996. [Online]. Available: http://scg.unibe.ch/archive/software/w3catalog/W3CatalogHistory.html (accessed 2021-01-30).

[5] L. Page, “Method for node ranking in a linked database”, US6285999B1, Sep. 2001.

[6] *Most Popular Search Engines 1994 - 2019.* [Online]. Available: https://www.youtube.com/watch?v=yMpj33Y95ZU (accessed 2021-02-13).

[7] *Search engine market share.* [Online]. Available: https://netmarketshare.com/search-engine-market-share.aspx (accessed 2021-02-13).

[8] *Search engine market share worldwide | Statista.* [Online]. Available: https://www.statista.com/statistics/216573/worldwide-market-share-of-search-engines/ (accessed 2021-02-13).

[9] *Search Engine Market Share Worldwide | StatCounter Global Stats.* [Online]. Available: `https://gs.statcounter.com/search-engine-market-share/all/worldwide/2019` (accessed 2021-02-13).

[10] *How Google Search Works | Google Search Central | Google Developers.* [Online]. Available: `https://developers.google.com/search/docs/beginner/how-search-works` (accessed 2021-02-26).

[11] B. Swapna and T. Anuradha, "Achieving higher ranking to webpages through search engine optimization", in *Proceedings of International Conference on Computational Intelligence and Data Engineering*, N. Chaki, A. Cortesi, and N. Devarakonda, Eds., Singapore: Springer Singapore, 2018, pp. 105–112, ISBN: 978-981-10-6319-0.

[12] G. Kumar and R. K. Paul, "Literature review on on-page & off-page seo for ranking purpose", *United International Journal for Research & Technology*, vol. 01, no. 06, 2020.

[13] *Get your website on Google - Search Console Help.* [Online]. Available: `https://support.google.com/webmasters/answer/34397` (accessed 2021-02-24).

[14] *What are Sitemaps?* [Online]. Available: `https://www.sitemaps.org/index.html` (accessed 2021-01-28).

[15] *Learn about sitemaps | Google Search Central | Google Developers.* [Online]. Available: `https://developers.google.com/search/docs/advanced/sitemaps/overview` (accessed 2021-01-26).

[16] *Sitemaps XML format.* [Online]. Available: `https://www.sitemaps.org/protocol.html` (accessed 2021-01-28).

[17] *Robots.txt Introduction & Guide | Google Search Central.* [Online]. Available: `https://developers.google.com/search/docs/advanced/robots/intro` (accessed 2021-02-20).

[18] P. Patil Swati, B. Pawar, and S. Patil Ajay, “Search engine optimization: A study”, *Research Journal of Computer and Information Technology Sciences*, vol. 1, no. 1, pp. 10–13, 2013.

[19] *Automatically generated content | Google Search Central*. [Online]. Available: `https://developers.google.com/search/docs/advanced/guidelines/auto-gen-content` (accessed 2021-03-15).

[20] *Affiliate programs | Google Search Central | Google Developers*. [Online]. Available: `https://developers.google.com/search/docs/advanced/guidelines/affiliate-programs` (accessed 2021-03-17).

[21] *Sneaky Redirects | Google Search Central | Google Developers*. [Online]. Available: `https://developers.google.com/search/docs/advanced/guidelines/sneaky-redirects` (accessed 2021-03-15).

[22] *Link Schemes | Google Search Central | Google Developers*. [Online]. Available: `https://developers.google.com/search/docs/advanced/guidelines/link-schemes` (accessed 2021-03-15).

[23] *SEO Implications: Hidden Text and Links | Google Search Central*. [Online]. Available: `https://developers.google.com/search/docs/advanced/guidelines/hidden-text-links` (accessed 2021-03-16).

[24] *Irrelevant Keywords & Keyword Stuffing | Google Search Central*. [Online]. Available: `https://developers.google.com/search/docs/advanced/guidelines/irrelevant-keywords` (accessed 2021-03-17).

[25] *User-Generated Spam | Google Search Central | Google Developers*. [Online]. Available: `https://developers.google.com/search/docs/advanced/guidelines/user-gen-spam` (accessed 2021-03-17).

[26] *How To Prevent Comment Spam | Google Search Central*. [Online]. Available: https://developers.google.com/search/docs/advanced/guidelines/prevent-comment-spam (accessed 2021-03-17).

[27] *Malicious Websites and Links | Google Search Central*. [Online]. Available: https://developers.google.com/search/docs/advanced/guidelines/malicious-behavior (accessed 2021-03-18).

[28] *Manual Actions report - Search Console Help*. [Online]. Available: https://support.google.com/webmasters/answer/9044175#what-is-manual-action (accessed 2021-03-18).

[29] *Google On The Worst it Can Do Against Black Hat SEO*. [Online]. Available: https://www.searchenginejournal.com/google-on-the-worst-it-can-do-against-black-hat-seo/396146/ (accessed 2021-06-29).

[30] *Control what you share with Google | Google Search Central*. [Online]. Available: https://developers.google.com/search/docs/advanced/crawling/control-what-you-share (accessed 2021-03-01).

[31] *Robots meta tag, data-nosnippet, and X-Robots-Tag specifications*. [Online]. Available: https://developers.google.com/search/reference/robots_meta_tag (accessed 2021-02-21).

[32] *Alexa - Top sites*. [Online]. Available: https://www.alexa.com/topsites (accessed 2021-04-17).

[33] *Our history in depth - Google Company*. [Online]. Available: https://web.archive.org/web/20170129090547/https://www.google.com/about/company/history/ (accessed 2021-04-23).

[34] *Almost Everything You Need to Know About Google's History*. [Online]. Available: https://interestingengineering.com/almost-everything-you-need-to-know-about-googles-history (accessed 2021-04-23).

[35] *Google*. [Online]. Available: https://www.google.com (accessed 2021-04-17).

[36] *Alphabet: revenue by segment 2018 | Statista*. [Online]. Available: https://www.statista.com/statistics/633651/alphabet-annual-global-revenue-by-segment/ (accessed 2021-05-02).

[37] *Google makes it Official: WebPosition Gold™ is Dead. - Aimclear®*. [Online]. Available: https://www.aimclear.com/google-makes-it-official-webposition-gold-is-dead/ (accessed 2021-03-17).

[38] *Automated queries | Search Central | Google Developers*. [Online]. Available: https://developers.google.com/search/docs/advanced/guidelines/automated-queries (accessed 2021-03-17).

[39] *Gmail on Twitter*. [Online]. Available: https://twitter.com/gmail/status/1055806807174725633 (accessed 2021-05-09).

[40] *Inside AdWords: Building for the next moment*. [Online]. Available: https://adwords.googleblog.com/2015/05/building-for-next-moment.html (accessed 2021-05-11).

[41] *StatCounter Global Stats - Browser, OS, Search Engine including Mobile Usage Share*. [Online]. Available: https://gs.statcounter.com (accessed 2021-06-22).

[42] *G is for Google*. [Online]. Available: https://blog.google/alphabet/google-alphabet/ (accessed 2021-06-23).

[43] *Why Google Became Alphabet*. [Online]. Available: https://hbr.org/2015/08/why-google-became-alphabet (accessed 2021-06-23).

[44] *Google Alphabet Split: EU Antitrust Not a Factor*. [Online]. Available: https://www.businessinsider.com/google-alphabet-split-eu-antitrust-not-a-factor-2015-8 (accessed 2021-06-23).

[45] *Google 1999*. [Online]. Available: `https://web.archive.org/web/19990117032727/http://www.google.com/` (accessed 2021-05-02).

[46] *Google's Search Algorithm and Ranking System - Google Search*. [Online]. Available: `https://www.google.com/search/howsearchworks/algorithms` (accessed 2021-04-17).

[47] N. Hajaj, "Producing a ranking for pages using distances in a web-link graph", US9165040B1, Oct. 2015.

[48] *Skole*. [Online]. Available: `https://www.skoleapp.com` (accessed 2021-05-24).

[49] *Create a Next.js App | Learn Next.js*. [Online]. Available: `https://nextjs.org/learn/basics/create-nextjs-app` (accessed 2021-05-25).

[50] *Understand JavaScript SEO Basics | Google Search Central*. [Online]. Available: `https://developers.google.com/search/docs/guides/javascript-seo-basics#how-googlebot-processes-javascript` (accessed 2021-05-27).

[51] *Rendering on the Web | Google Developers*. [Online]. Available: `https://developers.google.com/web/updates/2019/02/rendering-on-the-web#seo` (accessed 2021-05-27).

[52] *Cloud Market Ends 2020 on a High while Microsoft Continues to Gain Ground on Amazon | Synergy Research Group*. [Online]. Available: `https://www.srgresearch.com/articles/cloud-market-ends-2020-high-while-microsoft-continues-gain-ground-amazon` (accessed 2021-05-27).

[53] *Mobile-first indexing best practices | Search Central*. [Online]. Available: `https://developers.google.com/search/mobile-sites/mobile-first-indexing` (accessed 2021-07-01).

[54] *Add a website property - Search Console Help*. [Online]. Available: `https://support.google.com/webmasters/answer/34592?hl=en` (accessed 2021-06-01).

[55] *Skole's Sitemap.* [Online]. Available: https://www.skoleapp.com/sitemap.xml (accessed 2021-06-01).

[56] *Tell Google about localized versions of your page | Search Central.* [Online]. Available: https://developers.google.com/search/docs/advanced/crawling/localized-versions#sitemap (accessed 2021-06-11).

[57] *Create Crawlable Links | Google Search Central | Google Developers.* [Online]. Available: https://developers.google.com/search/docs/advanced/guidelines/links-crawlable?safari_group=9 (accessed 2021-07-01).

[58] *The Open Graph protocol.* [Online]. Available: https://ogp.me (accessed 2021-07-14).

[59] *Advanced Technical SEO: How social image sharing works and how to optimize your og:image tags • Yoast.* [Online]. Available: https://yoast.com/advanced-technical-seo-social-image-ogimage-tags/ (accessed 2021-07-14).

[60] *How does Facebook Sharer select Images and other metadata when sharing my URL? - Stack Overflow.* [Online]. Available: https://stackoverflow.com/a/7623986 (accessed 2021-08-29).

[61] *Basic Features: Data Fetching | Next.js.* [Online]. Available: https://nextjs.org/docs/basic-features/data-fetching (accessed 2021-07-02).

[62] *How does Google calculate the number of results? - Webmasters/Site owners Help.* [Online]. Available: https://web.archive.org/web/20090423004739/http://www.google.com/support/webmasters/bin/answer.py?hl=en%5C&answer=70920 (accessed 2021-05-31).

Appendix A PageRank in Python

```python
from dataclasses import dataclass, field
from decimal import Decimal

ALPHA = Decimal("0.1")
N = Decimal("4")


@dataclass
class Page:
    rank: Decimal = 1 / N
    backlinks: list["Page"] = field(default_factory=list)
    links: list["Page"] = field(default_factory=list)


def rank(page: Page) -> Decimal:
    return ALPHA / N + (1 - ALPHA) * sum(
        b.rank / len(b.links) for b in page.backlinks
    )


def main() -> None:
    page_a = Page()
    page_b = Page()
    page_c = Page()
    page_d = Page()

    page_a.backlinks = [page_c, page_d]
    page_a.links = [page_b, page_c, page_d]

    page_b.backlinks = [page_a]
    page_b.links = [page_d]

    page_c.backlinks = [page_a]
    page_c.links = [page_a]

    page_d.backlinks = [page_a, page_b]
    page_d.links = [page_a]

    for __ in range(10):
        # Do enough iterations for the ranks to fully converge.
        a_new = rank(page_a)
        b_new = rank(page_b)
        c_new = rank(page_c)
        d_new = rank(page_d)
        page_a.rank = a_new
        page_b.rank = b_new
        page_c.rank = c_new
        page_d.rank = d_new

    print("A:", round(page_a.rank, 2))  # A: 0.42
    print("B:", round(page_b.rank, 2))  # B: 0.15
    print("C:", round(page_c.rank, 2))  # C: 0.15
    print("D:", round(page_d.rank, 2))  # D: 0.28


if __name__ == "__main__":
    main()
```