TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

Fakultät Elektrotechnik Feinwerktechnik Informationstechnik

Entwicklung eines Suchalgorithmusprototypen zur Bewertung von Suchergebnissen verschiedener Kategorien

Studienarbeit im Studiengang Software Engineering

vorgelegt von

Marc Jonas Roser

Matrikelnummer 364 7316

Betreuer: Prof. Dr. Hans-Georg Hopf

Kurzdarstellung

Das Ziel der vorliegenden Studienarbeit ist es, eine bestehende Datenbank mit multimedialen Inhalten möglichst effizient nach unterschiedlichen Kriterien zu durchsuchen. Suchergebnisse sollen nach bestimmten Kriterien gewichtet, gefiltert und sortiert werden. Vorschläge für eine weiterführende Navigation auf der Suchergebnisseite sollen angeboten werden, Suchergebnisse sollen dazu nach Kontext und Wahrscheinlichkeiten gewichtet angezeigt werden. Das theoretische Fundament dieser Arbeit stellt die wissenschaftliche Betrachtung der Methoden zur Bewertung der Relevanz von Suchergebnissen dar. Die Arbeit untersucht die Möglichkeit, einen Suchbegriff so zu analysieren, dass ein Nutzer die bestmögliche Ergebnisliste bzw. zielgerichtete weiterführende Navigationsmöglichkeiten erhält. Die bestehende Anwendung "Crossload" wird vorgestellt, um dem Leser einen Kontext zu bieten, in der sich die Entwicklung bewegt.

Abstract

The goal of this student research project is to search an existing database with multimedia content as efficiently as possible according to various criteria. Search results are to be weighted, filtered, and sorted according to certain criteria. Suggestions for further navigation on the search results page are to be offered, and search results are to be displayed weighted according to context and probabilities. The theoretical foundation of this work is the scientific consideration of methods for evaluating the relevance of search results. The work examines the possibility of analyzing a search term in such a way that a user receives the best possible list of results or targeted further navigation options. The existing application "Crossload" is presented to provide the reader with a context in which the development takes place.

Eidesstattliche Erklärung

Hiermit versichere ich, Marc Jonas Roser, ehrenwörtlich, dass ich die vorliegende Studienarbeit mit dem Titel: „Entwicklung eines Prototypen eines Suchalgorithmus zur Bewertung von Suchergebnissen verschiedener Kategorien“ selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Nürnberg, 25.09.2022

Marc Jonas Roser

Contents

Glossary

API Application Programming Interface. I, 1, 4

Crossload Plattform zum Durchsuchen und Anhören einer umfassenden Predigtdatenbank. I, II, 1, 2

SOLR Open Source Suchframework der Apache Foundation. I, 1

Suchmaschine Eine Anwendung, die gezielt Ergebnisse aus dem Internet für den Nutzer aufbereitet und sortiert. Englisch: "Search Engine". I, 2

1 Einleitung

1.1 Relevanz des Themas

Suchalgorithmen und relevante Suchergebnisse sind derzeit so relevant wie noch nie. Dabei wollen die Benutzer einer Suchmaschine in Sekundenbruchteilen Ergebnisse, die am besten zu ihrem Suchbegriff passen, ohne sich dabei viel Gedanken über die Formulierung eines solchen Begriffes zu machen. Ein Beispiel für einen solchen Algorithmus ist Google, welches seit den frühen 2000ern einen kometenhaften Aufstieg in der Welt der Suchmaschinen hinter sich hat, was anhand der erreichten Werbeeinnahmen sichtbar wird.[1] Google ist im Vergleich zu anderen Suchmaschinen so stark verbreitet[2], dass mittlerweile sogar der Duden das Verb „googeln“ als eigenen Begriff für die Recherche im Internet führt.[3] Dabei stellt sich für die Entwicklung eigener Produkte die Frage, wie aus einem Suchbegriff, der meist nur aus wenigen Wörtern bis zu einem ganzen Satz besteht, relevante Suchergebnisse gefunden werden können. Dies würde zur Akzeptanz der Nutzer im Hinblick auf die entwickelte Funktionalität führen, da gewünschte Ergebnisse schneller und ohne großen Aufwand gefunden werden können.

1.2 Ausgangssituation

Derzeit besteht bei Crossload[4], einer Plattform zum Durchsuchen und Anhören einer umfassenden Predigtdatenbank, eine Datenbank mit einer Such API auf Basis von Spring Boot und SOLR. Diese teilt auf der Suchergebnisseite die Ergebnisse nach Kategorien auf und somit können nur schwer übergreifende Suchanfragen getätigt werden. Zwar werden alle Treffer auf der gleichen Seite angezeigt, doch durch die Aufteilung nach Kategorien werden Ergebnisse gewisser Kategorien über anderen gezeigt, auch wenn niedrig positionierte Kategorien relevantere Ergebnisse enthalten.

1.3 Zielsetzung

Das Ziel der vorliegenden Studienarbeit ist es durch eine theoretische Betrachtung der Bewertung der Relevanz von Suchergebnissen und der anschließenden Entwicklung eines Prototypen, ein bestehendes Produkt zu erweitern. Diese Erweiterung umfasst, die nach Kontext und Wahrscheinlichkeiten gewichtete und gefilterte Suche über eine Datenbank mit Datentypen verschiedener Kategorien bei der zusätzlich Vorschläge zur weiteren Navigation auf der Suchergebnisseite gegeben werden sollen.

[1] Siehe 1
[2] Siehe 2
[3] vgl. Duden [1]
[4] Siehe Crossload.org [2]

2 Grundlagen

2.1 Relevanz

Relevanz ist allgemein beschrieben eine Beziehung zwischen einem Individuum, dem zeitlichen Rahmen, in welchem dieses eine Information benötigt und einer beliebigen Information.[1] Das bedeutet, dass Relevanz von Person zu Person unterschiedlich ist, da zum einen diverse Informationen nur zu einer bestimmten Zeit notwendig bzw. wichtig sind und der Kontext der benötigten Information sich ständig ändert.

2.2 Relevanz von Suchergebnissen und Methoden zur Bewertung

Eine Suchmaschine gibt nach Anfrage Websiten sortiert nach der Relevanz der Ergebnisse abhängig zum gegebenen Suchbegriff des Nutzers. Die Schwierigkeit dabei, ist die Bestimmung der Relevanz für eine beliebige Website. Moderne Suchmaschinen nutzen dutzende oder gar hunderte verschiedener Methoden um Features um die Relevanz der verfügbaren Suchergebnisse zu bewerten. Die spezifischen Funktionen und Methoden werden von den Unternehmen geheim gehalten, um einen Missbrauch ihrer Suchmaschine zu verhindern. Dennoch sind die am häufigsten genutzten Merkmale bekannt und in einigen wissenschaftlichen Arbeiten untersucht worden.[2]

Zur Einfachheit wird von der Webapplikation Crossload abstrahiert und stattdessen Beispiele aus der Internetsuche verwendet, welche zum Beispiel mit Google, Bing, Ecosia oder anderen Suchmaschinen üblich ist.

2.2.1 Textuelle Relevanz

Das einfachste Merkmal für die Bewertung der Relevanz ist den kompletten Inhalt nach der textuellen Relevanz zu bewerten. Da natürliche Sprache, die meist für Suchergebnisse genutzt wird, generell ungenau ist, wird mit sogenannten "Matching Functions" versucht auch ungefähre Übereinstimmungen in einem Fliesstext zu finden. Einige der verwendeten Funktionen um die textuelle Relevanz zu bewerten sind dabei[3]

- Die Anzahl der Treffer für den Suchterm oder Abwandlungen
- Position des Suchterms (früheres Vorkommen)
- Seiten Struktur (für Websiten: Ist der Term eine Überschrift o.ä.)

[1] vgl. Bookstein S. 1 [3]
[2] vgl. Zaragoza, Najork, S. 1 [4]
[3] vgl. Zaragoza, Najork, S. 1 [4]

- Grafisches Layout (für Websiten: Ist der Term z.B. farblich markiert)

2.2.2 Relevanz durch Attribute

2.2.3 Hyperlink Relevanz

2.2.4 Relevanz durch Nutzerverhalten

2.2.5 Performance

2.2.6 Personalisierung

2.2.7 Kombination

3 Anforderungen und Problemanalyse

Im folgenden Kapitel sollen alle wesentlichen Funktionen der Erweiterung dargestellt werden. Da es sich um ein kleines Projekt mit einem abgestecktem Rahmen handelt und es kein Team gibt, welches die Entwicklung durchführt, wird das Wasserfallmodell für diese Entwicklung verwendet.

3.1 Vorgehensweise

Zur Erfassung der Anforderungen bzw. Requirements Engineering werden User Stories benutzt. Diese sind bekannt aus agilen Softwareentwicklungsmodellen, wie z.B. Scrum, entstanden aber durch praktische Erfahrungen in der Softwareentwicklung. Konzeptioniert wurden sie von Dr. Ivar Jacobsen[1] und Ron Jeffries[2]. Mithilfe einfacher Sprache wird aus der Sicht des Stakeholders das Ziel einer Story in einem kurzen Satz zusammengefasst. Anschließend wird dieses Ziel begründet, um die Wichtigkeit und Existenzberechtigung der Story zu begründen. User Stories sind dabei auch Anforderungen nach dem SMART Prinzip[3], da diese nur einen sehr kleinen abgesteckten Teilbereich einer Funktionalität enthalten. Dadurch sind sie einfacher schätzbar, umsetzbar und testbar. Anhand der ermittelten User Stories werden nach der Entwicklung Akzeptanztests durchgeführt, um den Erfolg des Endproduktes objektiv zu bewerten.

3.2 User Stories

Die Anforderungen umfassen alle Aktionen, welche der Nutzer in der Anwendung durchführen will. Ziel aller Anforderungen ist die übergreifende Suche über mehrere Kategorien nach effizient nach mehreren Kriterien zu durchsuchen und zu bewerten. Anhand dieses Ziels werden User Stories entwickelt, in denen ein Nutzer und andere Personen ihre Anforderungen an das zu entwickelnde Produkt stellen. Nichtfunktionale Anforderungen werden bei diese Anforderungserhebung nicht beachtet, da es sich hierbei um die Erweiterung einer bestehenden API handelt und Punkte wie Benutzerfreundlichkeit und User Experience hierbei wenig relevant sind, beziehungsweise das Entwicklungsumfeld durch die bereits bestehende Anwendung vorgegeben ist. Die Priorität der einzelnen User Stories ergibt sich aus der unten gegebenen Reihenfolge. Ich, als Benutzer, will ...

... für ein gegebenes Suchkriterium relevante Suchergebnisse über mehrere Kategorien hinweg erhalten, damit mit einer einzelnen Suche nur eine geringe Teilmenge der Datenbank angezeigt wird.

[1] vgl. Jacobson, Spence, Kerr 2016. [5]
[2] vgl. Ron Jeffries [6]
[3] vgl. Witte 2019a, S. 67 [7]

... die erhaltenen Suchbegriffe nach Kontext und Wahrscheinlichkeit gewichtet erhalten, damit diese im späteren Verlauf sortiert werden können. Der Kontext ergibt sich aus möglichen Schlagworten, die im Suchbegriff verwendet worden, womit z.B. eine Kategorie, ein Attribut eines Ergebnisses oder höher gewertet wird. Beispiele wären:

... Der Titel eines Buches wird „relativ“ genau als Suchbegriff eingegeben, folglich wird dieses Buch stärker gewichtet.

... Der Suchbegriff enthält den Term „Video“, folglich werden alle Videos priorisiert.

... die erhaltenen Suchbegriffe anhand des errechneten Gewichts absteigend sortiert zurückgegeben werden, damit das relevanteste Suchergebnis auf der Suchergebnisseite ganz oben steht.

... ein mit hoher Wahrscheinlichkeit gesuchte Suchergebnis als Vorschlag angezeigt bekommen, damit auf der Suchergebnisseite eine schnelle Navigation zu diesem Ergebnis möglich ist. Dabei soll nur ein Vorschlag angezeigt werden, wenn er der einzige mit einer hohen Wahrscheinlichkeit, in der Suchergebnisliste vorhanden durch das vorher errechnete Gewicht, und dieses Gewicht eine gewisse Schwelle überschreitet. Dadurch sollen verhindert werden, dass von ähnlich relevante Suchergebnisse nur eines vorgeschlagen wird und ein Suchergebnis vorgeschlagen wird, welches für das gegebene Suchkriterium irrelevant ist.

4 Konzeption

4.1 Analyse der Arbeitspakete

4.2 Konzeption

5 Entwicklung des Prototyps

6 Evaluation

7 Diskussion der Ergebnisse

7.1 Abgleich der Ergebnisse mit den gestellten Anforderungen

7.1.1 Funktionale Anforderungen

7.1.2 Nichtfunktionale Anforderungen

7.2 Identifizierte Probleme

7.3 Bewertung

8 Ausblick

Anhang

1 Einleitung

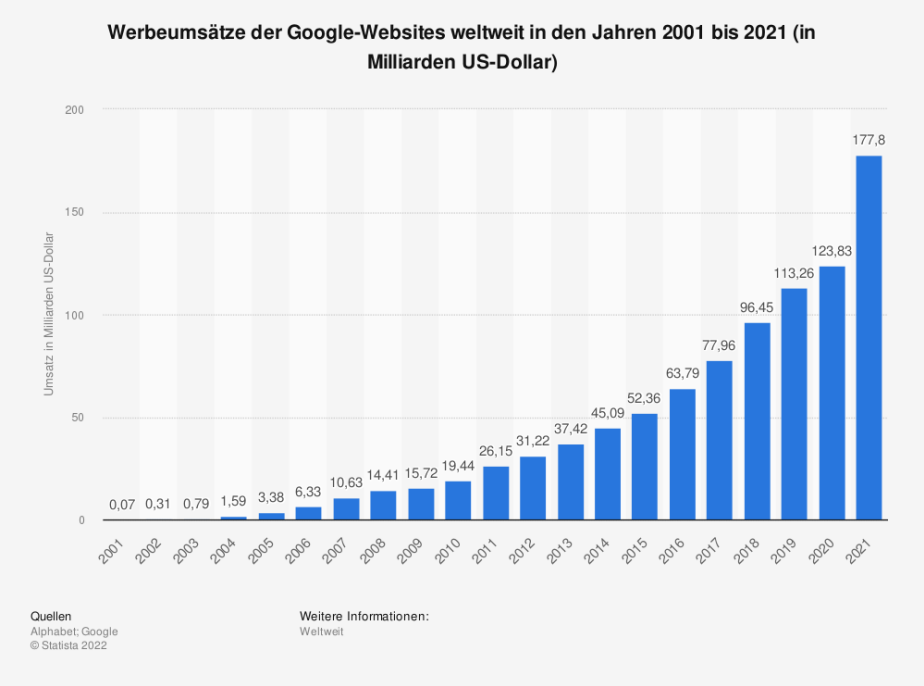

Figure 1: *Werbeumsätze Google Websites [8]*

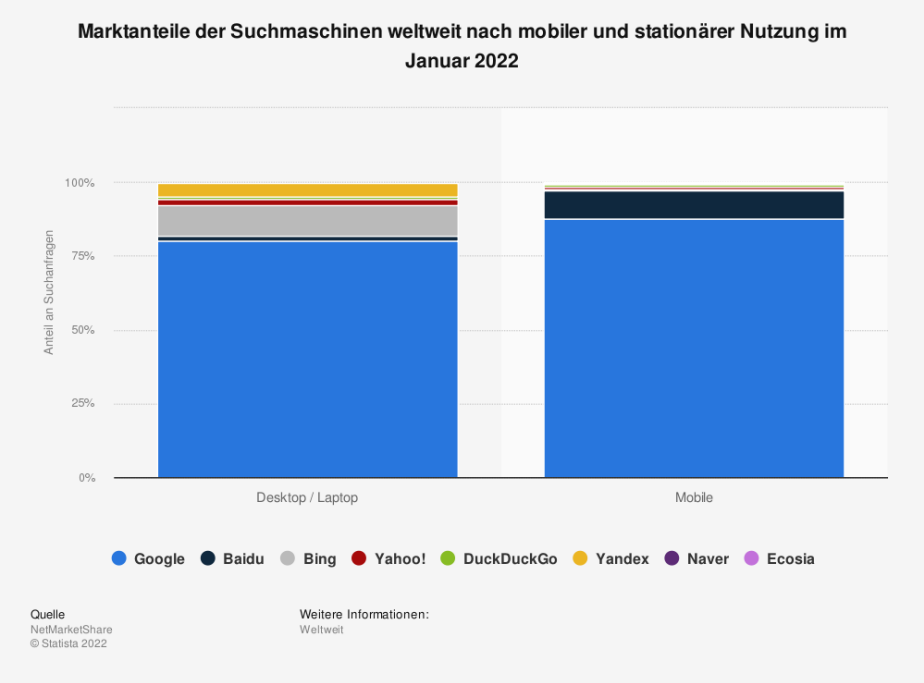

Figure 2: *Marktanteil Google [9]*

List of Figures

List of Tables

Bibliography

[1] Duden, "Duden | Wie schreibt man „googeln"? | Rechtschreibung." https://www.duden.de/rechtschreibung/googeln, 2022.

[2] D. Pfleiderer, "CROSSLOAD." https://crossload.org/info/ueber, Sept. 2022.

[3] A. Bookstein, "Relevance," *Journal of the American Society for Information Science*, vol. 30, pp. 269–273, Sept. 2007.

[4] H. Zaragoza and M. Najork, "Web Search Relevance Ranking," in *Encyclopedia of Database Systems* (L. Liu and M. T. Özsu, eds.), pp. 4650–4655, New York, NY: Springer New York, 2018.

[5] I. Jacobson, I. Spence, and B. Kerr, "Use-Case 2.0: The Hub of Software Development," *Queue*, vol. 14, pp. 94–123, Jan. 2016.

[6] R. Jeffries, "Essential XP: Card, Conversation, Confirmation." https://ronjeffries.com/xprog/articles/expcardconversationconfirmation/, 2022.

[7] F. Witte, *Testmanagement und Softwaretest*. Wiesbaden: Springer Fachmedien, 2016.

[8] Alphabet, "Werbeumsätze der Google-Websites weltweit in den Jahren 2001 bis 2021." https://de.statista.com/statistik/daten/studie/75181/umfrage/werbeumsatz-der-google-websites-seit-2001/, Feb. 2022.

[9] NetMarketShare, "Marktanteile der Suchmaschinen - Mobil und stationär 2022." https://de.statista.com/statistik/daten/studie/222849/umfrage/marktanteile-der-suchmaschinen-weltweit/, Feb. 2022.