UNIVERSITÄT
KOBLENZ · LANDAU
Fachbereich 4: Informatik

Wie lassen sich die Interessen eines Nutzers einsetzen, um die Relevanz der Ergebnisse einer Websuche zu erhöhen?

Bachelorarbeit

zur Erlangung des Grades eines Bachelor of Science
vorgelegt von

Carina Saal

Erstgutachter:	Prof. Dr. Steffen Staab Institut for Web Science and Technologies
Zweitgutachter:	René Pickhardt Institut for Web Science and Technologies

Koblenz, im September 2013

Erklärung

Hiermit bestätige ich, dass die vorliegende Arbeit von mir selbständig verfasst wurde und ich keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet–Quellen – benutzt habe und die Arbeit von mir vorher nicht in einem anderen Prüfungsverfahren eingereicht wurde. Die eingereichte schriftliche Fassung entspricht der auf dem elektronischen Speichermedium (CD-Rom).

	Ja	Nein
Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.	☐	☐
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	☐	☐

.....

(Ort, Datum) (Unterschrift)

Zusammenfassung

Das World Wide Web bietet jeder Einzelperson die Möglichkeit Webseiten zu erstellen, ohne ihn dabei in der Anzahl der Dokumente und Links einzuschränken. Dies hat zur Folge, dass das World Wide Web einem unregelmäßigen Wachstum unterliegt. Nach dem Stand vom September 2013 besteht das World Wide Web aus mindestens 4,21 Milliarden Seiten. Mit der wachsenden Anzahl an Seiten und Nutzern wird es zunehmend schwieriger Dokumente aufzufinden. Information Retrieval Systeme bieten den Nutzern die Möglichkeit zur Stichwort Eingabe, um diese Menge der Dokumente zu durchsuchen. Doch bieten Suchmaschinen in der Regal einen einzigen Suchalgorithmus für jeden ihrer Nutzer an. Dabei ist relativ unwahrscheinlich, dass sie mit dieser Methode die Bedürfnisse aller Suchenden befriedigen können. Eine Verbesserung für diesen Sachverhalt soll die personalisierte Suche leisten. Diese sieht es vor, dass in die Suche Informationen zum Nutzer einbezogen werden. So sollen dessen Bedürfnisse besser erfasst und mit einer größeren Wahrscheinlichkeit befriedigt werden.

Diese Arbeit präsentiert Algorithmen zu einer personalisierte Suche mithilfe von Präferenzen. Zu gegebenen Interessen und Query liefern diese Algorithmen die top zehn Vorschläge für eine Autovervollständigung. Das Ziel der Arbeit ist es zu zeigen, ob diese Implementierung einer personalisierten Suche eine Erhöhung der Relevanz der Ergebnisse ermöglichen kann.

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Einleitung

Das World Wide Web bietet jeder Einzelperson die Möglichkeit Webseiten zu erstellen, ohne ihn dabei in der Anzahl der Dokumente und Links einzuschränken. Dies hat zur Folge, dass das World Wide Web einem unregelmäßigen Wachstum unterliegt. Nach dem Stand vom September 2013 besteht das World Wide Web aus mindestens 4,21 Milliarden Seiten[1]. Mit der wachsenden Anzahl an Seiten und Nutzern wird es zunehmend schwieriger Dokumente aufzufinden. Information Retrieval Systeme bieten den Nutzern die Möglichkeit zur Stichwort Eingabe, um diese Menge der Dokumente zu durchsuchen. Doch bieten Suchmaschinen in der Regal einen einzigen Suchalgorithmus für jeden ihrer Nutzer an. Dabei ist relativ unwahrscheinlich, dass sie mit dieser Methode die Bedürfnisse aller Suchenden befriedigen können. Selbst wenn zwei Nutzer die gleiche Query eingeben, können sich deren Bedürfnisse unterscheiden. Ein Beispiel sei das Stichwort „Bank“. Für eine Person kann dies die Suche nach einer Sitzgelegenheit bedeuten. Eine andere Person könnte wiederum nach einem Kreditinstitut suchen. Verwenden diese Personen das Stichwort „Bank“ in einer Suchmaschine, erhalten jedoch beide die gleiche Menge an Ergebnissen. Diese Ergebnisse sollten sich nach der Person richten, die die Suche ausführt. Als eine Lösung für diesen Sachverhalt wird die personalisierte Suche angesehen. Diese soll für jeden Nutzer Ergebnisse liefern, welche die individuellen Bedürfnissen dessen beachten.

In den letzten Jahren wurde große Aufmerksamkeit auf das Adaptieren von Suchalgorithmen gerichtet. Diese Adaptionen sind das Integrieren von Informationen über den Nutzer, sowie das Bereitstellen personalisierte Ergebnisse. Dafür wurden zum einen verschiedene Strategien zur Erfassung von Daten über den Nutzer untersucht.

[1] http://www.worldwidewebsize.com/

Außerdem wurden verschiedenen Methoden entwickelt um diese Daten in eine Suche zu integrieren. Einige dieser Vorgehensweisen werden in realen Systemen eingesetzt, andere bedürfen noch weiterer Forschung.

Studien wie [CZG+09][MPS07]und [Spe05] befassten sich mit verschiedenen Daten zur personalisierten Suche. [Spe05] untersuchte 2005 eine personalisierte Suche basierend auf der Suchhistorie eines Nutzers. Eine personalisierte Suche basierend auf Interessen des Nutzers wurde 2007 von [MPS07] untersucht. [CZG+09] personalisierten 2009 anhand des sozialen Netzwerks eines Nutzers.

[DSW07] unternahm 2005 eine Evaluation und Analyse verschiedener Strategien zur personalisierten Suche. Dabei ließ sich feststellen, dass die Strategien unter verschiedenen Voraussetzung unterschiedlich effektiv sind. Ein Ergebnis dieser Untersuchung war, dass die Strategien noch weit davon entfernt sind, optimal zu sein und es noch weiteren Verbesserungen bedurfte.

Dies bedeutet jedoch nicht, dass zu diesem Zeitpunkt noch keine personalisierte Suchen zum Einsatz kamen. 2005 startete Google seine personalisierte Suche[2]. Diese basiert auf der Suchhistorie eines Nutzers. Dafür mussten diese jedoch mit ihrem Google Konto eingeloggt sein und die Möglichkeit zur Suchhistorie in ihrem Konto eingeschaltet haben. Seit 2009 wird auf Google weltweit auch für ausgeloggte Nutzer personalisiert[3].

Studien zur personalisierten Suche sind immer noch ein aktuelles Thema. So ist [FKA13] ein Beispiel einer aktuellen Forschung, die sich vor allem auf die Performance eines solchen Systems konzentriert. [WHC+13] ist ebenfalls eine aktuelle Arbeit. Diese schlägt ein Modell zum Ranking in einem personalisiertem System vor.

Die personalisierte Such ist ein Gebiet, das weiterhin Bedarf zur Forschung hat. Aufgrund dessen beschäftigt sich diese Arbeit mit der Entwicklung und Evaluation einer personalisierten, graphenbasierten Autovervollständigung. Dabei soll untersucht werden, ob die Integration von Interessen eine Erhöhung der Relevanz der Ergebnisse ermöglichen kann. Dafür werden zunächst in den Kapiteln 2 und 3 einige Grundlagen zu den Themen Graphentheorie und Information Retrieval gelegt. Anschließend bietet Kapitel 4 einen Überblick zur personalisierten Suche. Darauf folgt Kapitel 5, welches das entwickelte System im Detail erläutert. An dieses schließt eine Beschreibung der

[2]http://googleblog.blogspot.de/2005/06/search-gets-personal.html (Stand: 29.09.2013, 16:35Uhr)

[3]http://googleblog.blogspot.de/2009/12/personalized-search-for-everyone.html (Stand: 29.09.2013, 16:38Uhr)

Untersuchung und Evaluation dieses Systems an. Abschließend werden in Kapitel 7 die Ergebnisse, sowie in Kapitel 8 Fazit und Ausblick vorgelegt.

Kapitel 2

Graphentheorie

2.1 Grundlagen

Ein Graph dient der Darstellung einer Menge von Objekten und insbesondere ihren Verbindungen. Diese Objekte werden dabei als Knoten bezeichnet und bei der Beschreibung eines Graphen in der Menge V der Knoten zusammengefasst. Die Verbindungen zwischen solchen Knoten sind die sogenannten Links, welche zusammen die Menge E der Kanten ergeben. Dabei gilt: $E \subset V \times V$. Ein Graph ist somit ein Tupel (V,E) mit einer Menge V aus Knoten und einer Menge E aus Links. Graphen werden zum Beispiel zur Modellierung von Stadtplänen, Wasserleitungsplänen und Schienennetzen genutzt. Ein Beispiel für die Darstellung eines Graphen bietet Figur 2.1. In einer solchen Zeichnung eines Graphen werden die Knoten als Kreise und die Links als Striche dargestellt.

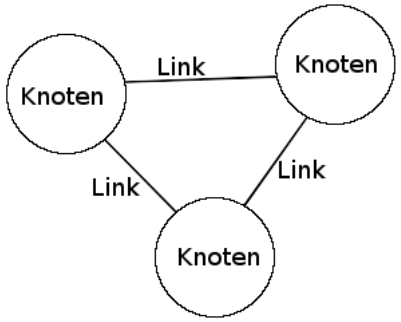

Abbildung 2.1: Die Zeichnung als Darstellung eines Graphen

2.2 Eigenschaften von Graphen

Durch verschiedene Eigenschaften in Bezug auf Knoten und Links, lassen sich verschiedene Problemstellungen durch Graphen darstellen. Im weiteren Verlauf dieser Arbeit kommen die zwei im folgenden beschriebenen Eigenschaften zum Einsatz.

Graphen können sich darin unterscheiden, welche Art von Links sie verwenden. Es kann sich um einen gerichteten oder einen ungerichteten Graphen handeln. Bei gerichteten Graphen liegen richtungweisende Links vor. Ein gerichteter Link wird durch ein geordnetes Knotenpaar beschrieben. Dieser Link wird dabei zu einem sogenannten ausgehenden Link für den ersten Knoten, und zu einem eingehenden Link für den zweiten Knoten dieses Paares. Die Anzahl der eingehenden Links eines Knotens wird als Innengrad, sowie die Anzahl der ausgehenden Links als Außengrad bezeichnet. Im ungerichteten Graphen besitzen die Links keine Richtung. In diesem Fall beschreibt der Grad eines Knoten die Anzahl aller mit ihm verbundenen Links.

Eine Eigenschaft die im Graphen in Bezug auf seine Knoten zu finden ist, ist der Abstand. Dieser ist ein Maß für die kürzeste Verbindung zwischen zwei Knoten. Dies entspricht der minimalen Anzahl an Links, die die beiden Knoten miteinander verbinden.

2.3 Suche auf Graphen

Bei dem Suchen auf Graphen differenziert man zwischen zwei Varianten, der Breitensuche und der Tiefensuche. Diese beiden Methoden unterscheiden sich darin, wie sie ihren Weg über die jeweiligen Knoten fortsetzen, in welcher Reihenfolge die Knoten bearbeitet werden.

2.3.1 Breitensuche

Die Breitensuche macht sich die Eigenschaft des Abstands zu Nutze. Ausgehend von einem Startknoten wird der Graph in die Breite gehend durchsucht. Figur 2.2 zeigt den Vorgang einer solchen Suche. Zunächst werden nur diejenigen Knoten besucht, welche zum Startknoten den Abstand eins haben. Anschließend wird ein Schritt in die Breite gemacht. Es werden Knoten besucht, die den Abstand zwei zum Startknoten aufweisen. Diese Erhöhung des Abstands wird so lange fortgesetzt, bis kein Knoten mehr existiert, der noch nicht besucht wurde. Die Suche kann dabei auch auf eine

Breite begrenzt werden, um die Suche auf einen Teil der Knoten einzuschränken. Algorithmus 1 beschreibt den Pseudocode einer möglichen Breitensuche, bei der nach einem Knoten innerhalb des Graphen gesucht wird.

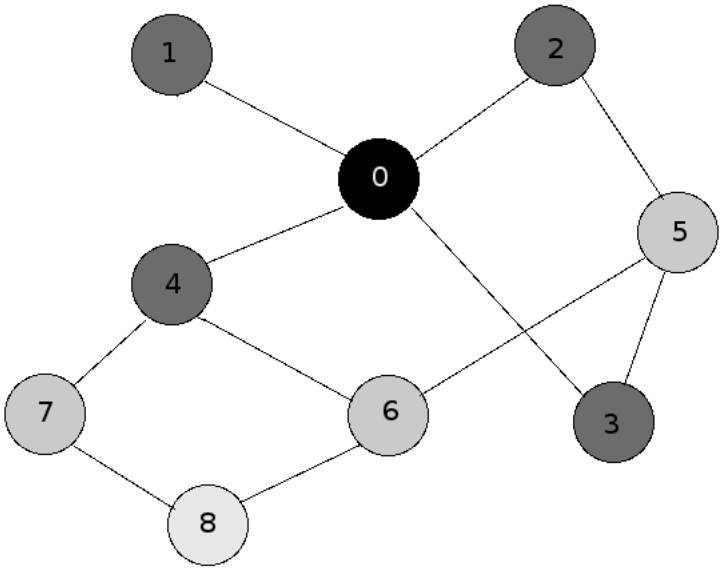

Abbildung 2.2: Die Darstellung des Ablaufs einer Breitensuche

```
   input: Startknoten, Zielknoten
 1 Füge Startknoten zur Queue hinzu;
 2 Markiere Startknoten als besucht;
 3 while Queue nicht leer do
 4 |   nehme Knoten von der Queue;
 5 |   if Knoten == Zielknoten then
 6 |   |   return true;
 7 |   end
 8 |   for jeden Nachbarn des Knotens do
 9 |   |   if Nachbar noch nicht besucht then
10 |   |   |   füge Nachbar der Queue hinzu;
11 |   |   |   markiere Nachbar als besucht;
12 |   |   end
13 |   end
14 end
15 return false;
```

Algorithm 1: Pseudocode einer Breitensuche

2.3.2 Tiefensuche

Im Gegensatz zur Breitensuche wird bei der Tiefensuche der Abstand zum Startknoten nicht kontinuierlich vergrößert. Bei der Tiefensuche kann dieser sich sowohl vergrößern, als auch wieder verkleinern. Figur 2.3 zeigt den möglichen Ablauf einer solchen Tiefensuche. Hierbei wird der Weg über die Knoten als ein durchgehender Pfad ausgeführt. Gelangt man an einen Knoten, dessen Nachbarn alle schon besucht wurden, so wird der Weg zurück verfolgt, und dort nach einem möglichen anderen Knoten gesucht. Algorithmus 2 beschreibt die Suche nach einem Zielknoten im Pseudocode.

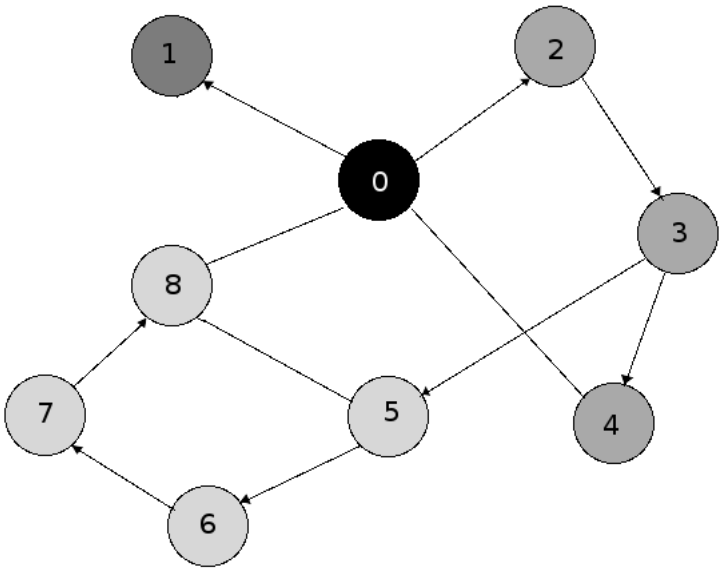

Abbildung 2.3: Die Darstellung des Ablaufs einer Tiefensuche

```
   input: Startknoten, Zielknoten
 1 Lege Startknoten auf den Stack;
 2 Markiere Startknoten als besucht;
 3 while Stack nicht leer do
 4 |   nehme Knoten vom Stack;
 5 |   if Knoten noch nicht besucht then
 6 |   |   if Knoten == Zielknoten then
 7 |   |   |   return true;
 8 |   |   end
 9 |   |   markiere Knoten als besucht;
10 |   |   for jeden Nachbarn des Knotens do
11 |   |   |   if Nachbar noch nicht besucht then
12 |   |   |   |   lege Nachbar auf Stack;
13 |   |   |   end
14 |   |   end
15 |   end
16 end
17 return false;
```

Algorithm 2: Pseudocode einer Tiefensuche

2.4 Graphdatenbank

Daten können auf unterschiedliche Weise dargestellt und gespeichert werden. Ein bekanntes Beispiel ist die Form der relationalen Datenbanken. Diese enthalten Daten in Tabellen, welche in Beziehung zueinander stehen können. Eine weitere Art der Darstellung und Speicherung von Daten ist die sogenannte Graphdatenbank. Bei dieser wird ein Graph zur Darstellung und Sicherung der Daten verwendet. Für diese Art der Repräsentation werden die Daten in Form von Knoten, Links und deren Eigenschaften dargestellt und gespeichert.

Kapitel 3

Information Retrieval

3.1 Was ist Information Retrieval

Eine Person, die auf der Suche nach Informationen ist, verfolgt immer ein oder mehrere Ziele. Um diese zu erreichen kann sie ein sogenanntes Information Retrieval System nutzen, das sie dabei unterstützt. Dieses Verhalten kann daher auch als „Information seeking“ bezeichnet werden. Die Ziele können unterschiedlich sein und folglich unterscheiden sich auch die Prozesse der Informationsbeschaffung. [MRS08] definiert das Verfahren des Information Retrieval, dem Wiederauffinden von Informationen, wie folgt:

> „Information Retrieval ist das Finden von Material (üblicherweise Dokumente) unstrukturierter Natur, aus großen Sammlungen (in der Regal auf Computern gespeichert), das ein Informationsbedürfnis befriedigt.“

Unter einem solchen Informationsbedürfnis versteht man zum Beispiel ein Thema, zu dem der Nutzer mehr erfahren möchte. Im Allgemeinen steht dabei eine Frage im Raum, die der Nutzer mit Hilfe des Information Retrieval Systems beantwortet haben möchte. Diese Frage wird vom Nutzer oft in Form einer sogenannten Query verfasst.

Ein einfaches, altbekanntes Beispiel für einen solchen Prozess ist der herkömmliche Betrieb einer Bibliothek. Hier sucht ein Besucher nach gewissen Informationen, meist in Form eines Buches. Diese kann er durch das von der Bibliothek bereit gestellte System finden. Das System kann zum Beispiel in Form eines Kataloges bereit gestellt werden. Durch die Eingabe des Begriffs „Information Retrieval“ werden dann eine Reihe von Büchern gelistet, die diesem Thema und dem Bedürfnis entsprechen.

Information Retrieval ist ein Prozess, der schon immer Bestandteil unseres Alltags ist. Sei es das Suchen eines Rezeptes in einem Rezept-Buch, das Auffinden eines Dokuments innerhalb eines Aktenordners oder das Suchen eines Buches in einer Bibliothek. Doch diese Aufgabe wird immer mehr zur Herausforderung, je größer eine Ansammlung aus Dokumenten ist. So wächst zum Beispiel das Inventar von Bibliotheken stetig weiter und so wurden schon 1950 erste Information Retrieval Systeme eingeführt, die eine Sammlung großer Mengen von Dokumenten durchsuchbar machten [Sin01]. 1960 gab es automatische Systeme, die zum Beispiel eine Sammlung von Zeitungsartikeln mit Hilfe von Stichworten durchsuchen ließen. Dabei handelte es sich beim Umgang mit solchen ersten Information Retrieval Systemen oft um eine Aktivität, die nur von wenigen Menschen ausgeübt wurde. Erste Systeme, wie die von Bibliotheken, verlangten speziell definierte Eingaben und konnten nur von ausgebildeten Bibliothekaren genutzt werden. Menschen, deren Job darin bestand, solche Suchaufträge auszuführen, wurden speziell darin trainiert, effektive Suchanfragen zu stellen.

Mit dem Einzug der Computer wurde es möglich große Mengen von Informationen zu speichern. Damit einhergehend stieg der Bedarf an Information Retrieval Systemen. Diese mussten es ermöglichen, die großen Mengen von Informationen zu durchsuchen. Computer liefern einerseits das Bedürfnis nach weiteren Information Retrieval Systemen, können aber andererseits auch das Information Retrieval unterstützen. Heute, in einer Zeit, in der Computer ein fester Bestandteil unserer täglichen Aktivitäten sind, gehört die Nutzung eines Informationen Retrieval Systems zum Alltag. Hunderte Millionen Menschen tätigen diese Aktivität täglich, wenn sie das World Wide Web nutzen. Information Retrieval wird zur vorherrschenden Form des Informationszugriffs [MRS08].

3.2 Anforderungen an ein IR System

3.2.1 Bedürfnisse erfassen

Die Hauptaufgabe eines Information Retrieval Systems besteht darin, die Bedürfnisse des Nutzers zu erfassen. In vielen Fällen geschieht die Nutzung eines solchen Systems durch eine Texteingabe, der sogenannten Stichwortsuche. Hierbei stehen Information Retrieval Systeme vor der Herausforderung, das Informationsbedürfnis anhand der Stichworte zu erkennen. Ein Problem mit dem die Systeme sowohl damals, als auch

heute noch zu kämpfen haben, ist die mangelnde Aussagekraft solcher Queries, wie sie [DJ10] beschreibt.

Die Bedürfnisse der Nutzer solcher Systeme sind oft sehr komplex. Diese in Stichworte zu fassen stellt eine sehr begrenzte Möglichkeit des Ausdrucks dar. Die Bedürfnisse werden durch Stichworte vereinfacht. Dadurch kann es dazu kommen, dass zwei Nutzer ihre unterschiedlichen Bedürfnisse durch die gleiche Query ausdrücken.

Hinzu kommt das Problem der Semantik. Damit einher gehen zwei Probleme, die Polysemie und die Synonymie. Unter Polysemie versteht man den Sachverhalt, dass ein Wort mehrere Bedeutungen haben kann. So kann der Begriff „Schloss“ sowohl eine Türschloss als auch ein Gebäude beschreiben. Ein weiteres, ähnliches Problem ist die Synonymie. Dies beschreibt den Fall, dass es mehrere Begriffe gibt, die die gleiche Bedeutung haben. So beschreiben „Kamera“ und „Fotoapparat“ den gleichen Gegenstand. Der Nutzer steht dabei vor der Wahl, welchen diese Begriffe er verwendet.

3.2.2 Kontext erschließen

Um die in Abschnitt 3.2.1 genannten Schwierigkeiten zu mindern, hilft es, den Kontext zu erschließen. Unter Kontext versteht man jegliche Information, die die Situation des Beteiligten bei der Interaktion mit dem System charakterisiert. Darunter fallen Standort, Identität, Zeit oder auch Präferenzen. Insbesondere auch die Suchhistorie kann Informationen zum Kontext liefern. Mithilfe des Kontext kann das Bedürfnis des Suchenden präzisiert werden.

3.2.3 Relevante Ergebnisse liefern

Nachdem sowohl die Bedürfnisse erfasst, als auch durch den Kontext präzisiert wurden, müssen folglich relevante Ergebnisse geliefert werden. Darunter versteht man solche, die das Informationsbedürfnis befriedigen. Die Relevanz beschreibt den Zusammenhang zwischen aufgefundenem Dokument und dem Bedürfnis des Suchenden. Ein Dokument ist relevant, wenn es eine signifikante Bedeutung für das Bedürfnis hat, wenn das Bedürfnis erfüllt wird. Außerdem wichtig ist das Relevanz-Maß, welches eine Einstufung der relevanten Dokumente bieten kann.

Der Relevanz-Begriff spielt bei der Suche eine entscheidende Rolle. Gute Suchalgorithmen, die auf einer Definition der Relevanz aufbauen, die falsch oder wenig befriedigend ist, sind unzureichend hilfreich.

3.3 Maßstäbe

[MRS08] unterteilt Information Retrieval Systeme unter anderem nach Maßstäben, welche sich vorwiegend nach der Größe und der Umgebung der Dokumentensammlung richten. Unterschiedliche Maßstäbe können verschiedene Herausforderungen für die Suche bedeuten. Dabei werden drei führende Maßstäbe aufgeführt. Ersterer ist das Unternehmens-, Instituts-, oder Domain-spezifische Suchen. Diese Suche erfolgt innerhalb einer internen Sammlung. Als zweiter Maßstab wird „personal search“ genannt, eine Suche, die zum Beispiel E-Mail Programme in Form von Spam Filtern anbieten. Als dritter und letzter Maßstab wird die Websuche gelistet. Hierbei handelt es sich um eine Suche auf Billionen von Dokumenten, die wiederum auf Millionen von Computern verteilt sind.

3.4 Die Websuche als Spezialfall

Die Entstehung des World Wide Web wird als die Revolution im Informationsbereich angesehen. Das World Wide Web ist heute die größte, existierende Sammlung von Dokumenten [LM06]. Billionen von Seiten werden jedes Jahr dem Web hinzugefügt. Es besteht eine immer weiter wachsende Menge von Informationen. Hier liegt der entscheidende Unterschied zum traditionellen Information Retrieval. Es gibt keine kleine, überschaubare Sammlung von Dokumenten. Die Sammlung des World Wide Web ist außerdem dynamisch, selbstorganisiert und enthält Hyperlinks. Ständige Veränderungen und extreme Unterschiede in den Metadaten stellen eine große Herausforderung sowohl für den Nutzer, als auch für die Entwickler von Suchmaschinen dar.

Zu Beginn stand dieser Informationsmasse ein großes Problem gegenüber: Die Suche im Web. Viele Informationen blieben den Nutzern des World Wide Web weiterhin verborgen und waren unzugänglich. Die ersten Information Retrieval Systeme die eine Suche im Web ermöglichen sollten, konnten die Bedürfnisse der Benutzer nicht befriedigen. [LM06] beschreibt die ersten Ansätze von Suchen im Web:

> „Die ersten Möglichkeiten von Informationsbeschaffungen bestanden darin, sich auf Yahoo durch eine Hierarchie von Themen zu suchen oder die vielen (oft tausenden) Webseiten zu durchsuchen, die eine damalige Suchmaschine als Ergebnisse lieferte. Dabei mussten die jeweiligen Seiten angeklickt und für sich selbst festgestellt werden, ob eine solche relevant

ist. Viele griffen daher auf Mundpropaganda zurück und fanden die für sie relevanten Seiten durch Empfehlungen anderer.“

Solche Systeme machten mit den Jahren eine große Entwicklung durch, die im folgenden Kapitel weiter beschrieben werden soll.

3.5 Modelle und Methoden

3.5.1 Boolean Retrieval

Die ersten Modelle für Information Retrieval Systeme bauten auf der Idee der sogenannten Aussagenlogik auf. Bei dieser geht es darum, Aussagen aufzustellen und mit logischen Operatoren zu verknüpfen. Diesen Aussagen wird dann ein Wahrheitswert zugeordnet. Suchmaschinen die auf Aussagenlogik beruhen sind die Ersten und Einfachsten ihrer Art. Die grundlegende Idee zum Relevanz-Begriff dieses Modells besteht darin, dass unter den Ergebnissen nur diejenigen Dokumente gelistet werden, die eine exakte Übereinstimmung mit der Suchanfrage des Nutzers haben. Die boolesche Algebra wird dabei genutzt, in dem Stichworte in der Query durch die booleschen Operatoren OR, AND und NOR verknüpft werden. Erfolgt so zum Beispiel eine Suche nach „Cat AND dog“, sollen Dokumente geliefert werden, die sowohl mit dem Stichwort Hund, als auch mit dem Stichwort Katze übereinstimmen. Durch eine Verbindung mit dem Operator OR dagegen sucht man nach Dokumenten, die nur den einen oder den anderen Begriff enthalten müssen. Bei dem Boolean Retrieval Modell ist das Auffinden von passenden Dokumente sehr einfach konzipiert: Ist das Schlüsselwort im Dokument enthalten, wird das Dokument als relevant eingestuft. Findet sich dieser Begriff nicht im Dokument wieder, gilt es als irrelevant für die Suche.

Dieser Ansatz eines Information Retrieval Systems scheint auf den ersten Blick eine brauchbare Lösung zu sein. Er entspricht der Standard Auffassung von Information Retrieval, dem Vergleich von Query und Text. Jedoch wird schnell klar, dass dieses System auch Probleme mit sich bringt. Besonders entscheidend ist das fehlende Maß für die Relevanz. Bei diesem Modell gibt es nur relevant und nicht relevant, so dass die Vorgehensweise einem Filter gleich kommt. Es erfolgt keine Einstufung der Ergebnisse. Stattdessen werden die Dokumente zum Beispiel nur chronologisch sortiert. Dies kann vor allem bei großen Dokumentensammlungen ein Problem darstellen. Zusätzlich kann das System den beiden zuvor erwähnten Problemen des Information Retrieval, der Synonymie und der Polysemie zum Opfer fallen.

Diese signifikanten Probleme sollen jedoch nicht bedeuten, dass dieses Modell keinen Gebrauch mehr findet. Durch seine simple Idee ist es für Programmierer einfach umzusetzen. Außerdem arbeitet diese Art von Algorithmus vergleichsweise schnell. Dies sind Gründe dafür, dass das Modell immer noch als Grundlage für Suchmaschinen verwendet wird.

3.5.2 Ranking

Um gegen das zuvor genannte Problem einer fehlenden Einstufung der Ergebnisse vorzugehen, besteht die Idee des sogenannten Rankings. Mit dem Ranking möchte man erreichen, dass die Ergebnisse eine Sortierung erhalten, die von der Relevanz abhängig ist. Damit sollen dem Suchenden unter den ersten Ergebnissen diejenigen gelieferten werden, die für ihn die höchste Relevanz haben.

Die zuerst ersichtliche und einfache Methode des Rankings ist demnach, die Ergebnisse dahingehend zur ordnen, wie hoch die Anzahl ihrer Übereinstimmungen zwischen Dokument und Stichwörtern der Query ist. Doch diese Idee bringt auch zwei Nachteile mit sich. Zum einen beinhalten längere Texte mehr Wörter und können folglich das gesuchte Stichwort öfter enthalten. Längere Texte haben bessere Chancen ein hohes Ranking zu erhalten. Zum anderen bietet dieser Ansatz die Möglichkeit zur Manipulation. Dokumente können besonders oft mit gewissen Stichwörtern versehen werden, um im Ranking höher eingestuft zu werden. Deshalb bildet dies nur eine grundlegende Idee für einen Algorithmus, der ein statisches Ranking vornimmt. Weitere unterscheiden sich darin, welches Kriterium sie für das Ranking verwenden.

3.5.3 Link Analysis

Einen entscheidenden Durchbruch und eine damit einhergehende deutliche Verbesserung der Suche im World Wide Web gab es 1998. Sowohl die Ausarbeitung von Kleinberg (1998), als auch die von Brin und Page (1998) führten die Idee des „Link Analysis Ranking“ ein. Der entscheidende Punkt dieser neuen Methode ist die Nutzung der Hyperlink Struktur des World Wide Web [LM06].

Zwar bringt die Suche im Web im Vergleich zur traditionellen Suche einige Nachteile mit sich, sie hat aber auch einen klaren Vorteil: Die Dokumente im Web sind untereinander verlinkt und bilden einen Graphen. Hyperlinks bieten uns die Möglichkeit für ein effektives und fokussiertes Suchen. Information Retrieval Systeme nutzen diese Besonderheit der Dokumentensammlung im Web, in dem sie die Links analysieren.

Das daraus erlangte Wissen wird nun in das Suchverfahren integriert. Es wird nicht mehr nur Wert auf Übereinstimmungen mit der Suchanfrage gelegt.

Bei dem Analysieren von Links geht es darum, ein Maß festzulegen, welches eine Aussage über die Qualität beziehungsweise Relevanz einer Menge von Links macht. Untersucht wird eine Menge von Links die auf eine gegebene Seite verweisen. Damit soll folglich wiederum eine Bewertung dieser Seite möglich sein. Suchmaschinen nutzen diese Bewertung um ein Ranking der Suchergebnisse zu ermöglichen.

3.5.4 PageRank

PageRank ist ein Link-Analyse Algorithmus, der nach dem Google-Mitbegründer Larry Page benannt ist und von der Suchmaschine Google verwendet wird [MPS07]. Dieser Algorithmus bietet die Berechnung einer Rangordnung, die auf dem Graphen des World Wide Web basiert. Sein Ziel ist es, jedes Element aus einer Menge von Dokumenten oder Seiten, mit einer numerischen Gewichtung zu versehen. Diese soll dessen relative Wichtigkeit innerhalb der Menge bemessen. Dies geschieht nur in Abhängigkeit der Hyperlinks[1]. Der eigentliche Inhalt einer Seite spielt dafür keine Rolle. Dadurch besteht auch keine Abhängigkeit zu einer Suchanfrage. Diese Werte werden ausschließlich für das anschließende Sortieren, dem Ranking der Ergebnisse, verwendet.

Als grundlegende Idee kommt beim Algorithmus des PageRanks das Zählen von eingehenden Links, den sogenannten Backlinks, zum Einsatz. Doch man musste sich genauere Gedanken darüber machen, welche Bedeutung solche Links haben und wie diese als Maß für die Relevanz einer Seite genutzt werden können. [PBMW99] zeigt in einem einfachen Beispiel auf, dass das bloße Zählen dieser Links nicht ausreichend ist um eine stimmige Wahl über den Stellenwert einer Seite zu treffen:

> „Eine bedeutende Webseite wie http://yahoo.com/ wird zehntausende solcher Backlinks haben. Damit erschließt sich, dass es sich um eine Seite mit hoher Relevanz handelt, deren Einordnung in einer Ergebnisliste weit oben stattfinden sollte. Man habe nun eine andere Webseite mit nur einem einzigen Backlink. Diese Seite scheint demnach einen sehr geringen Stellenwert zu haben. Doch ist dieser Link von einer bedeutenden Seite wie Yahoo, muss sie von weit größerer Relevanz sein.“

[1] Ein Hyperlink ist eine Verknüpfung beziehungsweise ein Querverweis zu einem anderen elektronischem Dokument oder einer anderen Stelle innerhalb des gleichen Dokuments.

Man sieht hier, dass ein weiterer Denkansatz nötig ist, um die Wichtigkeit einer Seite einzustufen. Googles PageRank sieht daher vor, nicht nur die Anzahl dieser Backlinks zu zählen, sondern damit auch Wertigkeiten zu verbinden. Google sieht dafür Links als Stimmen an. Mancher dieser Stimmen sind wichtiger als andere. Googles PageRank ist ein Zähler für diese Stimmen, welche dann das Ranking der Suchergebnisse beeinflussen. Wichtig für diese Idee ist, dass es dabei nicht nur auf die Links ankommt, sondern auch auf die Knoten, welche hier ebenfalls analysiert werden.

Der Algorithmus geht dabei wie in Abbildung 3 beschrieben vor: Das Initialisieren des PageRanks besteht darin, jeder Seite den gleichen Wert zu geben. In früheren Umsetzungen erhielten alle Seiten den Wert 1, so dass die Summe aller PageRank Werte der Anzahl der Seiten entspricht. Heute nutzt man dafür die Wahrscheinlichkeitsverteilung. Jede Seite erhält zu Beginn einen PageRank Wert von $\frac{1}{n}$, wobei n der Anzahl der Seiten entspricht. So ergibt sich eine Summe aller Werte von 1. Nun kommt es zur Betrachtung der Backlinks. Ein jeder Knoten im Graphen zählt seine ausgehenden Links. Nun verteilt er seinen eigenen PageRank-Wert auf die von ihm verlinkten Seiten auf. Algorithmus 3.1 beschreibt die Formel zur Berechnung des aktualisierten PageRank-Wert eines Knotens. Dafür werden die zuvor errechneten Bruchteile $\frac{PR_j}{c_j}$ summiert und mit einem Dämpfungsfaktor versehen. Damit soll verhindert werden, dass die PageRank-Werte nicht zu Seiten fließen, die keine ausgehenden Links besitzen. Weiterhin wird der Bruchteil $\frac{1-d}{n}$ aufaddiert, um das Random Surfer Modell zu realisieren. Dieses besagt, dass der Wechsel einer Seite nicht immer anhand von Aufrufen von Links geschieht. Ein Nutzer kann auch eine zufällige andere Seite besuchen.

$$PR_i = \frac{1-d}{n} + d \sum_{\forall j \in \{(j,i)\}} \frac{PR_j}{c_j}$$

Abbildung 3.1: Formel zur Berechnung des PageRanks

```
input: Graph, numberOfIterations
1  double d = 0.85 ;                          /* Dämpfungsfaktor setzen */
2  for jeden Knoten im Graphen do
3  |  Setze PageRank Wert auf 1/n;      /* Initialisiere PageRank Werte */
4  end
5  for int i = 0; i < numberOfIterations; i++ do
6  |  for jeden Knoten des Graphens do
7  |  |  zähle ausgehende Links des Knotens;
8  |  |  nehme PageRank Wert des Knotens;
9  |  |  Double delta = alpha * pageRank / AnzahlAusgehendeLinks;
   |  |  /* Teile PageRank Wert auf alle ausgehenden Links auf */
10 |  end
11 |  Aktualisiere PageRank Werte;
12 end
```

Algorithm 3: Pseudocode einer PageRank Berechnung

3.6 Query Autovervollständigung

Query Autovervollständigung wird heute von allen führenden Suchmaschinen angeboten [BYK11]. Dabei handelt es sich um eine Funktion, die den Nutzer während der Eingabe der Query mit einer Vervollständigung dieser versorgt. Die Autovervollständigung sagt anhand der bisher eingegebenen Query das Nutzerbedürfnis, die vollständige Query voraus. Dabei entspricht die bisherige Eingabe dem Präfix der Query. Es wird nach Dokumenten gesucht, die mit diesem Präfix übereinstimmen. [BYK11] gibt als das grundlegende Prinzip für die Autovervollständigung die „Intelligenz der Masse“ an. Nutzern werden die Dokumente vorgeschlagen, die in der Vergangenheit am beliebtesten waren.

Die Vorschläge durch die Autovervollständigung können dem Nutzer folglich während der Eingabe helfen, sein Bedürfnis zu formulieren. Außerdem führt dies dazu, dass der Nutzer Tastenanschläge einsparen und sich der Suchprozess beschleunigen kann.

Die Anforderung an ein solches System besteht darin, mit möglichst kurzen Präfixen zuverlässige Vorschläge zu machen. Im idealen Fall kann eine Autovervollständigung bei der Eingabe eines einzigen Buchstabens zuverlässige Vervollständigungen liefern.

Kapitel 4

Grundlagen zur Personalisierten Suche

4.1 Motivation

In Kapitel 3 wurde auf einige Ziele und damit verbundene Lösungsansätze für Information Retrieval Systeme, insbesondere die Suche im Web, eingegangen. Doch heutige Suchmaschinen sind noch immer weit davon entfernt, perfekt zu sein und den Bedürfnissen eines jeden Nutzers voll und ganz zu befriedigen. Dies hat verschiedene Ursachen. Neben den in Kapitel 3 erläuterten Problemen kommt hinzu, dass ein solches Information Retrieval System jeweils eine einzige Methode für alle ihre Nutzer anbietet. Durch die Integration von zusätzlichen Informationen über den Nutzer sollen dessen Bedürfnisse besser erfasst und mit einer größeren Wahrscheinlichkeit befriedigt werden können. Personalisierte Systeme sollen dem Nutzer Ergebnisse liefern, die besser auf das Bedürfnis dessen abgestimmt und damit einhergehend für ihn relevanter sind.

Wie bereits zuvor erwähnt, stellt die Stichwortsuche sowohl die Nutzer als auch die Programmierer vor einige Herausforderungen. Bindet das System nun aber Informationen des Nutzers ein, so können diese bewältigt werden. Durch das Einbinden wird die Suchanfrage verfeinert, wodurch Schwierigkeiten mit der Semantik umgangen werden können. Wird zusätzliches Wissen über den Suchenden in die Suche integriert, stellt auch die unzureichende Aussagekraft von Stichwörtern ein nicht mehr so starkes Problem dar. Im Folgenden soll daher näher darauf eingegangen werden, wie

solche personalisierten Such-Systeme im Detail aussehen können. Dafür wird in Kapitel 4.2 erläutert, welche Informationen für eine solche Suche genutzt werden können. Außerdem beschäftigt sich Kapitel 4.3 damit, wie solche Daten erfasst werden können. Kapitel 4.4 befasst sich mit dem Einbinden dieser Daten in den Ablauf der Suche.

4.2 Nutzerdaten

Bei der Umsetzung einer personalisierten Suche liegt der erste Schritt in der Wahl der Daten. Es muss entschieden werden, welche Daten des Nutzers für die Suche genutzt werden sollen. Zu gegebenen Bedürfnissen sollen die Ergebnisse an die Vorlieben, Hintergründe und das Wissen des Suchenden angepasst werden. Dafür unterscheidet [MGSG07] zwischen Anwenderdaten und Anwendungsdaten.

Die Anwenderdaten umfassen solche, die den Nutzer als Person beschreiben. Darunter fallen zum Beispiel das Alter oder Geschlecht des Nutzers. Auch geographische Daten können zum Einsatz kommen. Solche sind zum Beispiel in der Online Suchmaschine von Google in Gebrauch. Durch Beachtung des aktuellen Standorts des Suchenden, enthalten die Suchergebnisse länderspezifische Seiten. Auch bei der Suche nach Software erhält man entsprechende Ergebnisse, die sich an der verwendeten Software oder dem Betriebssystem orientieren können.

Neben den Anwenderdaten gibt es die Anwendungsdaten. Diese beschreiben das Verhalten des Benutzers während der Interaktion mit dem System. Darunter fallen Browser-Historien und Suchmaschinen Protokolle. [Spe05] führt eine weitere Unterscheidung für Nutzerdaten auf, die Präferenzen und Interessen. Erstes bezieht sich nicht auf den Inhalt der gewünschten Informationen, die gesucht werden. Stattdessen kann zum Beispiel das bevorzugte Format des gesuchten Dokumentes eine Rolle spielen. Auch hier bietet sich das Suchsystem von Google als Beispiel an: Der Nutzer kann den Typ des gewünschten Dokumentes schnell und einfach auswählen, wie etwa „Bild“, oder „Newsbeitrag“. Interessen dagegen beziehen sich auf den Inhalt der gesuchten Dokumente. Diese wiederum können in kurzzeitige und langzeitige Interessen unterteilt werden. Systeme, die mit Interessen des Nutzers arbeiten, sind anspruchsvoller, da sie versuchen Themen aus den Dokumenten zu extrahieren, die zum Bedürfnis des Nutzers passen [Spe05].

4.3 Datenerfassung

Die Erfassung der zuvor erläuterten Daten ist eine der wichtigsten Problemstellungen der personalisierten Suche [MGSG07]. Für den Erwerb der Nutzerdaten stehen zwei Möglichkeiten zur Verfügung, welche sich in der Integration des Nutzers unterscheiden. Es gibt sowohl eine direkte, explizite Methode, als auch eine indirekte, implizite Methode.

4.3.1 Die explizite Datenerfassung

Bei der expliziten Datenerfassung wird eine direkte Beteiligung des Suchenden vorausgesetzt. Dieser wird explizit nach zusätzlichen Information gefragt. Ein Beispiel dafür ist die Suche nach einem Buch auf amazon[1]. Neben der Eingabe eines Titels können hier weitere Präferenzen, wie etwa ein Genre, angegeben werden. Auch das Tool „Google Alerts“ behilft sich dieser Methode. Das Tool bietet an, den Nutzer zu benachrichtigen, wenn es neue Ergebnisse zu seiner Suchanfrage gibt. Dafür kann der Nutzer explizite Angaben darüber machen, in welche Art von Informationen er interessiert ist.

Auf diesem Weg Daten zu erfassen hat zur Folge, dass ein zusätzlicher Arbeitsaufwand für den Nutzer entsteht. Der Nutzer könnte aber nicht gewillt sein, diesen zu leisten. Weiterhin steht dieser auch hier vor dem Problem, sein Bedürfnis formulieren zu müssen.

4.3.2 Die implizite Datenerfassung

Als Alternative zur expliziten Methode steht die implizite Datenerfassung zur Verfügung. Hierbei verfolgt und untersucht man das Verhalten des Nutzers ohne dessen direkte Beteiligung. Ein Beispiel dafür ist die personalisierte Suche von Google. Google macht Gebrauch von vorherigen Suchanfragen. Dafür werden alten Anfragen und die dabei ausgewählten Seiten aus den Ergebnissen erfasst. Diese werden dann für zukünftige Suchanfragen verwendet[2]. Nach [JGP+05] bietet diese Methode drei Vorteile gegenüber der expliziten Datenerfassung. Sie kann mit weniger Aufwand betrieben und Daten in größeren Mengen gesammelt werden. Weiterhin fällt kein zusätzlicher

[1] http://www.amazon.de/

[2] http://googleblog.blogspot.de/2009/12/personalized-search-for-everyone.html (Stand: 29.09.2013, 18:54Uhr

Arbeitsaufwand für den Nutzer an. Sie ist jedoch gegenüber der expliziten Methode weniger exakt [LXZY10].

4.4 Integration der Nutzerdaten

Um die erfassten Informationen zu nutzen und in den Prozess des Suchens einzubinden, werden Nutzerprofile erstellt. Information Retrieval Systeme für personalisierte Suchen müssen dafür eine Komponente enthalten, die das sogenannte Benutzerprofil modelliert. [MGSG07] beschreibt drei verschiedene Phasen, in denen die Einbindung des Benutzerprofils stattfinden kann.

Die Methode der personalisierten Suchmaschine bindet das Benutzerprofil in den Suchprozess direkt ein. Das Auffinden und Bewerten von Dokumenten geschieht mit der Unterstützung dessen. Sowohl bei der Frage nach relevant oder nicht relevant, als auch beim Ranking werden die Daten des Profils zur Hand genommen. Ein Beispiel für diese Methode ist das Suchen nach Personen auf Facebook. Dort wird bei einer Personensuche das eigene Profil mit in den Suchprozess einbezogen. Es werden Personen vorgeschlagen die zum Beispiel durch den Wohnort oder andere Freunde eine Verbindung zum Profil aufweisen.

Eine weitere Möglichkeit zur Verwendung des Benutzerprofils bietet ein erneutes personalisiertes Ranking. Diese sieht eine Neuanordnung der Ergebnisse mithilfe des Profils vor. Der eigentliche Suchvorgang enthält dabei keine personalisierte Komponente. Als Beispiel für diese Methode kann die zuvor erwähnte Suche nach einem Buch auf amazon heran genommen werden. Nach der Ausführung einer Suche zu einem Buchtitel kann der Suchende eine Sortierung, wie etwa „nach Erscheinung“ auswählen. Folglich werden die Ergebnisse seinen Präferenzen entsprechend neu sortiert.

Eine dritte Methode bildet die Modifikation der Query. Hier kommt die Komponente des Benutzerprofils schon vor der eigentliche Suche zum Einsatz. Aus der eingegebenen Query wird mithilfe des Profils eine Neue erschlossen. Dies geschieht zum Beispiel durch das Hinzufügen weiterer Schlüsselbegriffe zur eigentlichen Query.

Diese Methoden sind nicht exklusiv und können miteinander kombiniert werden.

Kapitel 5

Personalisierte Suche mithilfe von Interessen

In Kapitel 4 wurden verschiedene Ansätze zur Umsetzung einer personalisierten Suche erläutert. Mit dieser Arbeit soll nun die Frage beantwortet werden, wie sich eine Websuche, insbesondere eine Autovervollständigung, mit Hilfe der Interessen eines Nutzers, in Bezug auf Relevanz verbessern lässt. Um diese Frage zu beantworten, wurde eine graphenbasierte Autovervollständigung entwickelt[1]. Diese setzt ein Benutzerprofil voraus, welches die Interessen des Nutzers repräsentiert. Ermöglicht wird damit eine sogenannte soziale Suche, dessen Kontext die Interessen des Nutzers sind. Auf der Grundlage einer bestehenden Graphdatenbank wurden verschiedene Algorithmen entworfen, um eine Autovervollständigung auf dem Graphen zu personalisieren.

5.1 Das Suchverfahren

Durch die Möglichkeit zur graphenbasierten Suche, wurde die Autovervollständigung anhand einer Breitensuche umgesetzt. Diese basiert auf der folgenden Idee zur Relevanz:

> Gegeben sei ein Startknoten, welcher das Interesse eines Nutzers repräsentiert. Ausgehend von diesem Knoten wird definiert, wie relevant alle anderen Knoten des Graphen für diesen Nutzer sind. Je kleiner der Abstand eines Knoten zum Startknoten, desto höher seine Relevanz.

[1] https://github.com/xCarina/PersonalizedAutoComplete

Ausgehend von einer Menge an Knoten des Benutzerprofils, werden mithilfe der Breitensuche diejenigen Knoten gesucht, deren Abstand zu einem Interessen-Knoten zwei oder kleiner ist. Weiterhin basieren die Algorithmen auf dem in Kapitel 3 beschriebenen Boolean Retrieval Modell. Dokumente werden als relevant eingestuft, wenn deren Titel eine Übereinstimmung zur Query aufweist.

Dabei handelt es sich um die in Abschnitt 4.4 vorgestellte Methode zur Einbindung des Benutzerprofils. Dieses wird direkt in die Suche integriert. Die Personalisierung geschieht sowohl im Ablauf des Suchprozesses selbst als auch bei der Einstufung zur Relevanz.

5.2 Das Ranking der Ergebnisse

Besonders bei einer Autovervollständigung spielen die wenigen ersten Ergebnisse eine wichtige Rolle. Dem Nutzer soll eine überschaubare Anzahl von Vorschlägen gemacht werden können. Deshalb sollen die Ergebnisse zusätzlich einem Ranking unterzogen werden. Aufbauend auf der in Abschnitt 5.1 beschriebenen Vorgehensweise zur Personalisierung wurden die in Tabelle 5.1 gelisteten Algorithmen entwickelt. Diese unterscheiden sich darin, wie sie beim Ranking der Ergebnisse vorgehen.

	Algorithmus	**Ranking**
1	*BFS_noR*	kein Ranking, zufällige Sortierung
2	*BFS_PR*	PageRank
3	*BFS_adaptPR*	adaptierter PageRank
4	*BFS_newR*	Anzahl Besuche bei der Breitensuche

Tabelle 5.1: Algorithmen zur personalisierten Suche

Der erste Algorithmus *BFS_noR* nimmt kein Ranking der Ergebnisse vor. Die Ergebnisse unterliegen keiner Sortierung und werden zufällig ausgegeben. Dieser Algorithmus soll für Vergleichszwecke zum Einsatz kommen und aufzeigen, dass ein Ranking zusätzlich zur Verbesserung beitragen kann.

BFS_PR arbeitet mit dem in Kapitel 3.5.4 beschriebenen PageRank. Dieser setzt voraus, dass die PageRank-Werte der Knoten des betreffenden Graphen bekannt sind. Die bei der Breitensuche gefundenen Knoten werden bei diesem Algorithmus ausgehend von ihren PageRank-Werten sortiert.

Ein dritter Algorithmus, *BFS_ adaptPR*, nutzt ebenfalls die PageRank-Werte. Weiterhin sieht dieser jedoch eine weitere Personalisierung anhand des Rankings vor. Die PageRank-Werte der gefunden Dokumente werden im Laufe der Breitensuche adaptiert. Die geschieht auf der Grundlage folgender Idee:

> Wird ein Knoten während der Breitensuche mehrmals besucht, so bedeutet dies, dass er mehrere Verbindungen zu den Startknoten, den Interessen, aufweist. Diese mehrfache Verbindung zu den Interessen kann ein Indiz für eine höhere Relevanz des Knotes sein.

Umgesetzt wird dies durch ein erneutes Aufaddieren des PageRank-Wertes bei jedem weiteren Besuch des Knotens.

Der vierte Algorithmus, *BFS_ newR*, sieht ebenfalls eine Personalisierung beim Ranking der Ergebnisse vor. Dieser nutzt jedoch nicht die Werte des PageRank. Stattdessen zählt dieser die Besuche eines jeden Knotens und sortiert anschließend die Ergebnisse anhand dieser Zahlen. Da hier der PageRank nicht zum Einsatz kommt, werden die Dokumente nicht mehr anhand ihrer Rolle im gesamten Graphen beurteilt. Das Ranking konzentriert sich auf deren Rolle im Teilgraphen, der von der Breitensuche beachtet wird.

Kapitel 6

Das Experiment

Die in Kapitel 5 beschriebene Methode zur personalisierten Autovervollständigung wurde einem Experiment unterzogen. Bei diesem kam ein Datensatz zum Einsatz, welcher in Abschnitt 6.1 erläutert wird. In Abschnitt 6.2 wird auf das Benutzerprofil für die Personalisierung eingegangen. Mit Abschnitt 6.3 folgt eine Beschreibung zum Ablauf des Experimentes. In Kapitel 7 werden folglich die Ergebnisse dieses Experiments im Detail dargelegt.

6.1 Der Datensatz

Bei diesem Experiment kam der Datensatz der deutschen Wikipedia[1] zum Einsatz. Dieser besitzt folgende Eigenschaften und kann die für das Experiment nötige Bedingungen erfüllen.

Eine wichtige Rolle spielen Größe und Struktur des Datensatzes. Bei Wikipedia handelt es sich um das weltgrößte Projekt frei verfügbarer Inhalte. Die deutsche Wikipedia stellt aktuell mit über 1.5 Millionen Artikeln die drittgrößte unter den Wikipedias dar[2]. Zusätzlich weisen dessen Artikel eine Hyperlink-Struktur auf. Diese beiden Aspekte führen dazu, dass eine Suche auf den Daten der Wikipedia einer Suche im Web in geringeren Maßstäben gleich kommt.

Eine Voraussetzung an den Datensatz ist die Möglichkeit zur Erstellung eines Benutzerprofils. Dies wird durch die Bearbeitungshistorie ebenfalls vom Datensatz der Wikipedia bereitgestellt. Dass die bearbeiteten Artikel eines Wikipedia-Nutzers

[1] Wikipedia Dumb vom 30.03.2013: http://dumps.wikimedia.org/dewiki/20130330/
[2] http://de.wikipedia.org/wiki/Wikipedia:Statistik (Stand: 03.09.2013)

als Interessen in Frage kommen, zeigt die Studie von [Nov07]. Hieraus geht hervor, dass einer der Hauptmotivationen, sich an Wikipedia zu beteiligen, der Spaß ist. Dies kann mit dem Interesse am Thema des jeweiligen Artikels einhergehen. Dazu ist das Bearbeiten von Wikipedia-Artikeln auch ein Einsatz von Zeit, Talent und Wissen, ohne eine Entlohnung. Ein weiterer Grund der dafür spricht, dass es sich bei solchen Bearbeitungen um Artikel handelt, die dem Interesse des Nutzers entsprechen. Für die Frage nach den Nutzern, die für das Experiment geeignet sind, ist ein weiterer Blick auf eine Analyse der Beteiligungen nötig. Hierfür bietet [PHT09] einen Blick auf die Qualität der Beiträge und von wem diese geleistet werden. Zum einen wird festgestellt, dass Nutzer mit insgesamt 1.000 oder mehr Bearbeitungen eine hohe Qualität liefern. Nutzer mit 1.000 bis 5.000 Überarbeitungen liefern zu einem Zeitpunkt eine höhere Qualität als Nutzer mit über 5.000 Bearbeitungen. Die gemessene Metrik beträgt im Durchschnitt 0,9. Bei Nutzern mit insgesamt weniger als 100 Bearbeitungen, liegt dieser Wert bei 0,7. Aus diesem Grund soll sich auf Nutzer mit 500 bis zu 5.000 Bearbeitungen konzentriert werden. Die Beitragszahlen von Wikipedia[3] zeigen, dass es von diesen Nutzern ausreichend viele gibt, um ein Experiment zu ermöglichen.

6.2 Das Nutzerprofil

Wie in Kapitel 4 beschrieben gibt es verschiedene Arten von Benutzerprofilen, welche sich sowohl durch ihre Erstellung als auch durch ihren Inhalt und letztlich auch in der Integration unterscheiden. Das Benutzerprofil das bei diesem Experiment zum Einsatz kam, wurde durch das Tool von René Pickhardt[4] erstellt. Die Interessen eines Nutzers lagen hier als Andwendungsdaten vor. Der Nutzer interagiert mit dem System, indem er Artikel auf Wikipedia bearbeitet. Aus der Bearbeitungshistorie[5] kann dann mittels des Tools auf indirektem Weg ein Benutzerprofil erstellt werden. Dieses extrahiert aus der Historie diejenigen Nutzer, die für das Experiment geeignet sind. Dies sind Nutzer, bei denen die Bearbeitungen als Interessen eingestuft werden können. Mit Hilfe des Tools konnten 2.259 Nutzer gefunden werden, die diese Bedingung erfüllen. Mit insgesamt über 3 Millionen Bearbeitungen durch diese Nutzer, decken diese fast ein Drittel aller Bearbeitungen und folglich einen ausreichend großen Teil des Wikipedia Datensatzes ab. Bei der Erstellung des Benutzerprofils wird weiterhin

[3] http://de.wikipedia.org/wiki/Datei:WP_Beitragszahlen.svg (Stand: 25.09.2013)
[4] http://www.rene-pickhardt.de/
[5] http://dumps.wikimedia.org/dewiki/20130330/dewiki-20130330-pages-logging.xml.gz

beachtet, wie oft ein Nutzer einen Artikel bearbeitet hat. Anhand der Anzahl der bearbeiteten Artikel können die Artikel einer zusätzlichen Ordnung unterliegen.

Auf diesen, durch das Tool gewonnenen Interessen, wurde dann ein 20/80 Split vollzogen. Dabei wurden die ersten 20 Prozent der Interessen zur Personalisierung herangezogen. Dies sind die Artikel, deren Anzahl der Bearbeitungen durch den Nutzer am größten war. Angenommen wird, dass dies mit dem Interesse an einem Artikel einhergeht.

6.3 Die Evaluation

Mit diesem Experiment sollten zwei Dinge untersucht werden:

1. Erhöht sich die Relevanz der Ergebnisse durch die Personalisierung?
2. Liefern die Algorithmen entscheidende Unterschiede beim Ranking der Ergebnisse?

Um dies zu tun, wurde ein weiterer Algorithmus implementiert. Dieser ist nicht personalisiert und kann so zu Vergleichszwecken herangezogen werden. Bei diesem handelt es sich um eine lineare Suche, welche die Ergebnisse entsprechend ihrer PageRank-Werte sortiert. Weiterhin wurden die Algorithmen jeweils mit den gleichen Nutzer, Query Paaren getestet, um diese untereinander vergleichen zu können. Bei der Nutzung der Algorithmen wurden folgende Parameter fest gewählt:

1. Länge der Query l = 1
2. Anzahl der vorgeschlagenen Autovervollständigungen k = 10

Für dieser Einstellung wurden 26 Queries gewählt. Diese entsprachen dem Alphabet, den Buchstaben A bis Z. Außerdem wurden 300 Nutzer zufällig ausgewählt, für die personalisiert wurde. Für jeden Algorithmus wurden folglich $300 \times 26 = 7800$ Paare aus Query und Nutzer gebildet und zur Autovervollständigung genutzt.

Die Ergebnisse dieser Autovervollständigungen sollten anschließend untersucht werden. Dafür wurden die bisher ungenutzten 80 Prozent der bekannten Interessen eines jeden Nutzers verwendet. Da diese bekannt sind, konnten die folgenden vier Metriken für die Ergebnisse jeder Suche berechnet werden:

1. $Precision = \frac{|\{relevant\ documents\} \bigcap \{retrieved\ documents\}|}{|\{retrieved\ document\}|}$

2. $Recall = \frac{|\{relevant\ documents\} \bigcap \{retrieved\ documents\}|}{|\{relevant\ document\}|}$
3. $F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$
4. $Precision\ at\ 10 = \frac{1}{m} \sum_{k=1}^{m} Precision(R_k)$

Mit Precision und Recall sollen Aussagen über die komplette Ergebnismenge gemacht werden können. Precision sagt aus, wie viele der gefundenen Dokumente relevant sind. Der Recall gibt an, wie viele der relevanten Dokumente gefunden wurden. Da mit einer erhöhten Precision meist ein verringerter Recall einher geht, soll zusätzlich die dritte Metrik berechnet werden. Mit dieser soll ein direkter Vergleich der Algorithmen möglich sein. Da bei einer Autovervollständigung wenige top Ergebnisse eine große Rolle spielen, soll außerdem ein Blick auf das Ranking geworfen werden. Dieses kann mit der Berechnung der Precision at 10 bewertet werden.

Kapitel 7

Ergebnisse

Um die Frage nach einer erhöhten Relevanz der Ergebnisse zu beantworten, wurde zuerst ein Vergleich der Preicison und Recall Werte vollzogen. Dabei wurden die Werte der nicht personalisierten Suche denen der personalisierten Suche gegenüber gestellt. Figur 7.1 zeigt eine Gegenüberstellung der durchschnittlichen Precision Werte beider Suchverfahren.

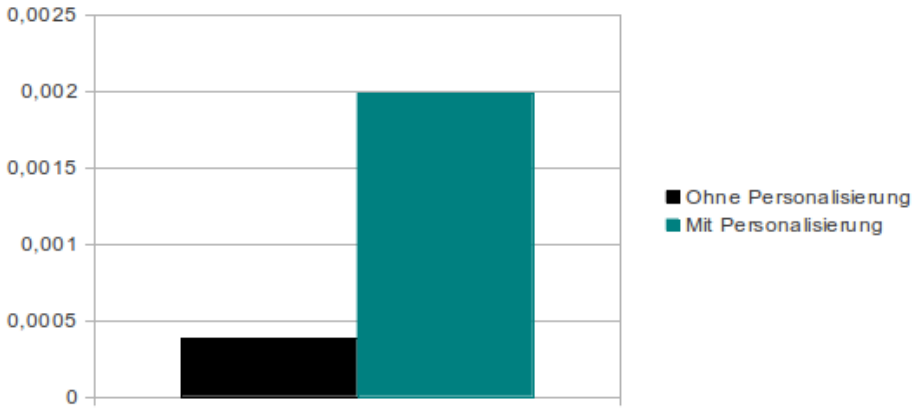

Abbildung 7.1: Durchschnittliche Precision Werte für eine Suche ohne (schwarz) und eine Suche mit Personalisierung (türkis)

Es ist zu erkennen, dass die personalisierte Autovervollständigung eine deutliche Erhöhung der Precision Werte mit sich bringt. Die Precision der personalisierten Suche beträgt 523% des Wertes der nicht personalisierten Methode. Dieser signifikante Unterschied ist ebenfalls in Tabelle 7.2 abzulesen. Das personalisierte Verfahren liefert

einen erkennbar größeren Maximalwert. Dies ist darauf zurück zu führen, dass die personalisierte Breitensuche eine kleinere Menge an Ergebnissen liefert als eine lineare Suche über alle Dokumente.

Eine kleinere Ergebnismenge hat wiederum eine Verminderung des Recall Wertes zur Folge. Je weniger Ergebnisse gefunden werden, desto geringer werden die Chancen, jedes relevant Dokument unter ihnen zu finden. Figur 7.2 stellt diesen Sachverhalt dar. Da die lineare Suche jeden Artikel findet, wurde hier ein hoher Recall Wert erwartet. Da jedoch nicht jeder Nutzer Interessen zu jeder Query hat, war ein Recall kleiner als 1.0 zu erwarten. Sowohl der Durchschnittswert als auch der Maximalwert in Tabelle 7.1 bestätigen diese Vermutung. Zwar liefert auch die personalisierte Autovervollständigung Maximalwerte von 1.0, jedoch nimmt der Recall im Durchschnitt ab. Die personalisierte Autovervollständigung liefert im Durchschnitt 60% der relevanten Dokumente. Damit beträgt der Recall 62% des Wertes der linearen Suche.

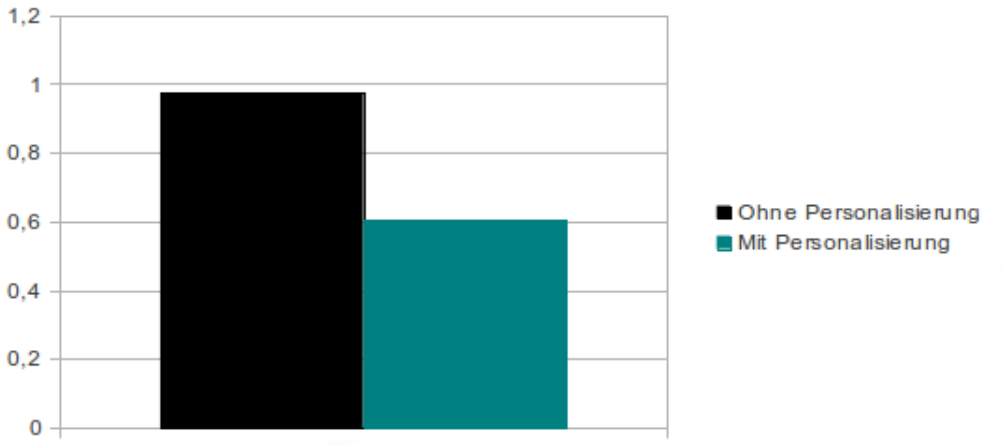

Abbildung 7.2: Durchschnittliche Recall Werte für eine Suche ohne (schwarz) und eine Suche mit Personalisierung (türkis)

Da bei diesen beiden Metriken auf der einen Seite eine Verbesserung, auf der anderen jedoch eine Verschlechterung zu beobachten war, wurde zusätzlich das sogenannte *F-Measure* berechnet. Dabei handelt es sich um ein kombiniertes Maß aus Recall und Precision. Abbildung 7.3 stellt die Werte der nicht personalisierten Suche und der personalisierten Suche gegenüber. Hierbei wird deutlich, dass es insgesamt eine Verbesserung gegeben hat. Der Wert bei der personalisierten Methode beträgt

500% des Wertes der nicht personalisierten Methode.

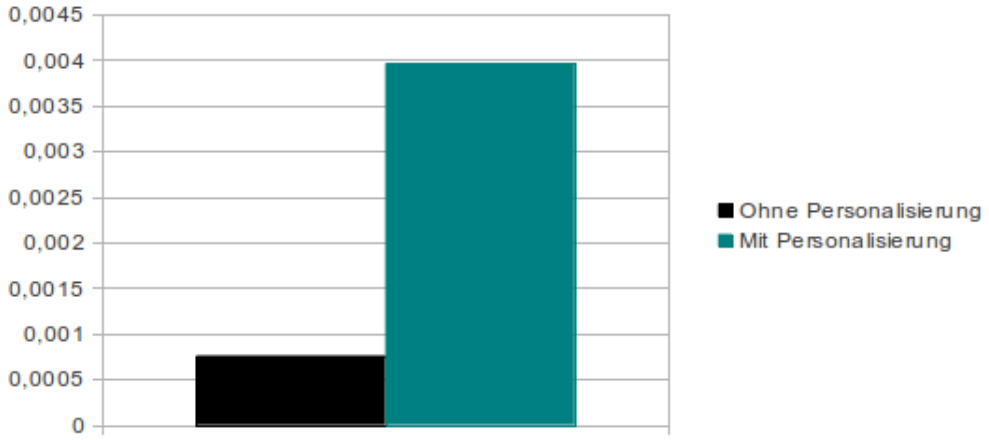

Abbildung 7.3: F-Measure für eine Suche ohne (schwarz) und eine Suche mit Personalisierung (türkis)

Weitere Vergleichswerte liefern die Precision at 10 Werte für die Algorithmen. Diese sind in Abbildung 7.4 gegenüber gestellt. Daraus abzulesen ist, dass der *BFS_noR* Algorithmus ohne ein Ranking signifikant schlechter abschneidet. Ebenfalls auffällig ist der Wert des *BFS_newP* Algorithmus. Auch dessen Ranking schneidet im Vergleich schlechter ab. Diese beiden Werte betragen nur 14% der Werte der drei anderen Algorithmen. Festzustellen ist, dass die nicht personalisierte Autovervollständigung bei den Precision at 10 Werten genauso gut abschneidet, wie eine personalisierte Autovervollständigung, deren Ranking den PageRank verwendet. Hieran wird deutlich, welche entscheidende Rolle die PageRank Werte spielen.

Die Durchschnittswerte zeigen, dass die personalisierte Autovervollständigung unter den festgelegten Parametern keine Verbesserung darstellt. Durch den eingeschränkten Blick auf das Verhalten der Algorithmen kann hier zusammenfassend nur festgehalten werden, dass diese personalisierte Autovervollständigung für eine Query-Länge von eins die Relevanz der ersten zehn Ergebnisse nicht erhöhen kann.

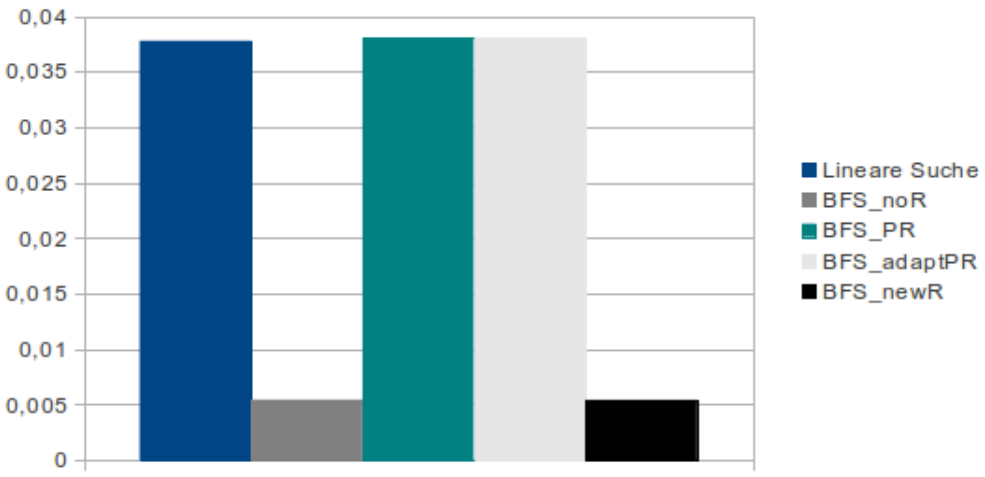

Abbildung 7.4: Durchschnittliche Precision at 10 Werte

Methode	Mittelwert	Std.-Abweichung	Min	Max
Ohne Personalisierung	0,97551	0,15457	0,0	1,0
Mit Personalisierung	0,60276	0,24112	0,0	1,0

Tabelle 7.1: Recall Werte der Algorithmen

Methode	Mittelwert	Std.-Abweichung	Min	Max
Ohne Personalisierung	0,00038	0,00046	0,0	0.01956
Mit Personalisierung	0,00199	0,00277	0,0	0.11409

Tabelle 7.2: Precision Werte der Algorithmen

Methode	Mittelwert	Std.-Abweichung	Min	Max
Lineare Suche	0,03777	0,13433	0,0	1,0
BFS_noR	0,00545	0,05433	0,0	1,0
BFS_PR	0,03802	0,13450	0,0	1,0
BFS_adaptPR	0,03802	0,13450	0,0	1,0
BFS_newR	0,00543	0,05424	0,0	1,0

Tabelle 7.3: Precision at 10 Werte der Algorithmen

Kapitel 8

Fazit und Ausblick

Mit dieser Arbeit wurde eine personalisierte Autovervollständigung entwickelt und untersucht, welche die Interessen eines Nutzers verwendet. Dafür wurden verschiedene Algorithmen mit einer Menge von Queries und Nutzern getestet und miteinander verglichen. Im Vergleich zur nicht personalisierten Suche konnte ein signifikanter Unterschied für die Precision erreicht werden. Dies brachte wie zu erwarten einen verringerten Recall Wert mit sich. Weiterhin zeigte jedoch keiner der Algorithmen eine deutliche Verbesserung im Vergleich zur nicht personalisierten Suche, bei den Precision at 10 Werten. Da das Experiment unter festgelegten Parametern von Query Länge und Anzahl an Autovervollständigungen ausgeführt wurde, bietet sich nur ein eingeschränkter Blick auf das Verhalten der entwickelten Algorithmen. Es kann noch keine vollständige Bewertung der vorgeschlagenen Methode gemacht werden. Dieses Experiment stellt nur ein Erstes dar, welches zukünftig erweitert werden kann.

Eine erste fortführende Studie sieht eine Untersuchung unter variierenden Parametern vor. Zum einen kann analysiert werden, wie sich die Metriken mit der Vergößerung der Query verhalten. Je länger eine Query, desto kleiner die Ergebnismenge. Dies kann signifikante Einflüsse auf die Werte von Precision und Recall haben. Weiterhin kann der Parameter k, der die Anzahl der top Ergebnisse bestimmt, variiert werden. Es können Beobachtungen angestellt werden, ob es signifikante Unterschiede bei den Werten der Precision at k gibt. Damit einher geht die Untersuchung, welcher Wert für k für die personalisierte Autovervollständigung am besten geeignet ist.

Eine andere Möglichkeit zur Erweiterung der Studie sieht die Entwicklung eines weiteren Algorithmus vor. Das Experiment zeigte, wie erfolgreich der PageRank ist. Somit bietet ein Algorithmus, der auf dem Teilgraphen der Breitensuche eine erneute

Berechnung der PageRank-Werte vollzieht, eine zusätzliche Alternative.

Fortführend können auch die integrierten Interessen des Nutzers variiert werden. In diesem Experiment kamen die meist bearbeiteten Artikel für das Benutzerprofil zum Einsatz. Stattdessen können aber die zuletzt bearbeiteten Artikel verwendet werden. Dies würde einen Vergleich zwischen langzeitigen und kurzzeitigen Interessen ermöglichen.

Literaturverzeichnis

[BYK11] Ziv Bar-Yossef and Naama Kraus. Context-sensitive query auto-completion. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 107–116, New York, NY, USA, 2011. ACM.

[CZG+09] David Carmel, Naama Zwerdling, Ido Guy, Shila Ofek-Koifman, Nadav Har'el, Inbal Ronen, Erel Uziel, Sivan Yogev, and Sergey Chernov. Personalized social search based on the user's social network. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1227–1236, New York, NY, USA, 2009. ACM.

[DJ10] Easley David and Kleinberg Jon. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World.* Cambridge University Press, New York, NY, USA, 2010.

[DSW07] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 581–590, New York, NY, USA, 2007. ACM.

[FKA13] Haizhou Fu, HyeongSik Kim, and Kemafor Anyanwu. Scaling concurrency of personalized semantic search over large rdf data, 2013.

[JGP+05] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 154–161, New York, NY, USA, 2005. ACM.

[LM06] Amy N. Langville and Carl D. Meyer. *Google's PageRank and Beyond: The Science of Search Engine Rankings.* Princeton University Press, Princeton, NJ, USA, 2006.

[LXZY10] Nathan N. Liu, Evan W. Xiang, Min Zhao, and Qiang Yang. Unifying explicit and implicit feedback for collaborative filtering. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1445–1448, New York, NY, USA, 2010. ACM.

[MGSG07] Alessandro Micarelli, Fabio Gasparetti, Filippo Sciarrone, and Susan Gauch. The adaptive web. chapter Personalized search on the world wide web, pages 195–230. Springer-Verlag, Berlin, Heidelberg, 2007.

[MPS07] Zhongming Ma, Gautam Pant, and Olivia R. Liu Sheng. Interest-based personalized search. *ACM Trans. Inf. Syst.*, 25(1), February 2007.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* 2008.

[Nov07] Oded Nov. What motivates wikipedians? *Commun. ACM*, 50(11):60–64, November 2007.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web, 1999.

[PHT09] Katherine Panciera, Aaron Halfaker, and Loren Terveen. Wikipedians are born, not made: a study of power editors on wikipedia. In *Proceedings of the ACM 2009 international conference on Supporting group work*, GROUP '09, pages 51–60, New York, NY, USA, 2009. ACM.

[Sin01] Amit Singhal. Modern Information Retrieval: A Brief Overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24(4):35–42, 2001.

[Spe05] Mirco Speretta. Personalized search based on user search histories. In *In Proc. of International Conference of Knowledge Management(CIKM), Washington D.C., 2004*, pages 622–628, 2005.

[WHC+13] Hongning Wang, Xiaodong He, Ming-Wei Chang, Yang Song, Ryen W. White, and Wei Chu. Personalized ranking model adaptation for web search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '13, pages 323–332, New York, NY, USA, 2013. ACM.

Danksagung

Zunächst möchte ich mich an dieser Stelle bei all denjenigen bedanken, die mich während der Anfertigung dieser Bachelor-Arbeit unterstützt und motiviert haben. Insbesondere meine Familie, von der ich mir nicht mehr Unterstützung hätte wünschen können.

Ganz besonders gilt mein Dank René Pickhardt, der meine Arbeit und somit auch mich betreut hat. Neben seinen kritischen Auseinandersetzungen mit meiner Arbeit, die mir immer wieder wertvolle Hinweise einbrachten, waren auch seine moralische Unterstützung und Motivation unschlagbar. Vielen Dank für die Geduld und Mühe.