Bachelorarbeit

Suchoptimierung mittels maschinellen Lernens

Zeitraum 04.07.2016 - 04.10.2016

Lukas Abegg

Matrikelnummer 798972
Sommersemester 2016
Fachsemester 6
Studiengang Medieninformatik (B.Sc.)
Beuth Hochschule für Technik

Beuth Hochschule für Technik
Luxemburger Str. 10
13353 Berlin

Betreuer **Prof. Dr. habil. Alexander Löser**
Fachbereich VI - Informatik und Medien
Beuth Hochschule für Technik

Gutachter **Prof. Dr. Martin Oellrich**
Fachbereich II - Mathematik - Physik - Chemie
Beuth Hochschule für Technik

Inhaltsverzeichnis

1 Einführung

Springer Nature ist ein weltweit führender Verlag für Forschungs-, Bildungs- und Fachliteratur mit einer breiten Palette an angesehenen und bekannten Medienmarken und zudem der weltweit größte Verlag für Wissenschaftsbücher. Für das Unternehmen Springer Nature ist es darum wichtig, auf seinen Web-Applikationen eine Suche anbieten zu können, die Suchintentionen erkennt und möglichst schnell zum gesuchten Content leitet. Die Suche wird vor allem als Hilfsmittel zur Navigation und zum Finden von Literatur und Dienstleistungen genutzt. Durch die vielen von Springer Nature publizierten Zeitschriften und Querverweise in Artikeln, wird sie aber auch oft zur Suche nach Issues[1] und Artikeln verwendet sowie als Hilfestellung um Diagnosen zu Krankheitsbilder stellen zu können.

Springer Nature sammelt viele User-Tracking-Daten und dadurch viel Wissen über das Verhalten der User[2] bei der Nutzung ihrer Suche, lässt dieses Wissen jedoch bisher noch nicht in ihre Suche einfließen. In dieser Arbeit wollen wir untersuchen, ob mithilfe dieses Wissens, die Suche optimiert werden kann.

1.1 Aufbau der Suche bei Springer Nature

White Label Applikation mit Solr-Suche

Damit die verschiedenen Verlage und Zeitschriften der Verlagsgruppe Springer Nature ihre Produkte und Dienstleistungen online anbieten können nutzt Springer Nature eine eigens entwickelte White Label Applikation[3]. Die White Label Applikation verwendet *Apache Solr* (im Folgenden SSolr"genannt) (siehe [Sol]) als Suchplattform. Die Solr dient hierbei als eine der Schnittstellen zwischen dem Content-Pool von Springer Nature und der White Label Applikation. Bei den vom Content-Pool gelieferten Inhalten, handelt es sich um von Springer Nature Verlag publizierte Zeitschriften, Artikel, Bücher und redaktionelle Inhalte.

User-Tracking mit Webtrekk

Um das Verhalten der User auf ihren Web-Applikationen zu tracken verwendet Springer Nature das Analysetool Webtrekk (siehe [Web]). Die daraus resultierenden Berichte bieten unter anderem die Möglichkeit, *Suchquery-Logs*[4] und *Click-Trough-Rates* (CTR)[5] der User auszuwerten.

[1] Nummer der Zeitschriftenausgabe, in der sich der Artikel befindet.
[2] Als User werden die Nutzer der Springermedizin-Suche bezeichnet
[3] Eine White Label Applikation ist eine wiederverwendbare und agil erweiterbare Applikation
[4] Protokoll über alle ausgeführten Suchanfragen auf der Applikation
[5] Kennzahl um die Anzahl der Klicks auf Links im Verhältnis zu den gesamten Impressionen darzustellen

Architektur

In Abb. 1 ist die Suche nochmals grafisch aufbereitet:

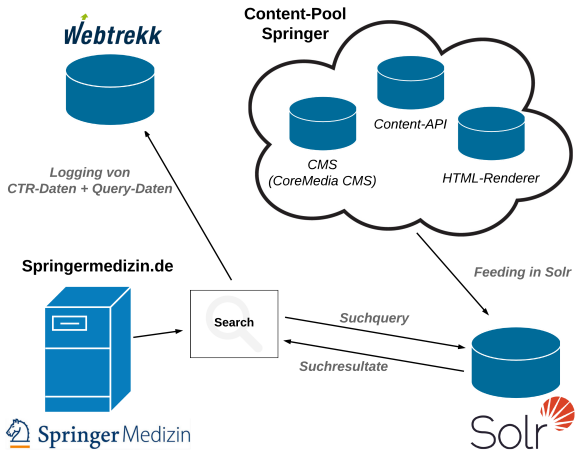

Abb. 1: Aufbau der Suche bei Springer Nature

1.2 Problemstellung: Keine Userrelevanz in der Suche

Userrelevante Dokumente werden nicht gefunden

Die User von Springermedizin suchen oft mit einschlägig, fundierten Fachbegriffen nach den neuesten und relevantesten Zeitschriften, Bücher oder Publikationen. Die zeitlich aktuellsten Suchtreffer zu finden ist für Springermedizin kein Problem. Die für den User *relevantesten* jedoch schon.

Der Springer Nature Stakeholder: Springermedizin setzt auf Webtrekk

Zu den Stakeholder[6] der in Kapitel 1.1 angesprochenen White Label Applikation gehört *Springermedizin* (siehe [Sme]). Springermedizin ist ein Fortbildungs- und Informationsportal für Ärzte. Mithilfe von Webanalysten und Webtrekk versucht Springermedizin das Marketing seines Webauftrittes zu verbessern und ist sehr interessiert an neuen Ansätzen, um die gesammelten Tracking-Daten besser einzusetzen. In dieser Arbeit wird darum der Fokus auf die Verwendung von Tracking-Daten in der Suche von Springermedizin gesetzt.

Der fast gläserne User

Springermedizin sammelt Tracking-Daten über jegliche Aktivitäten auf deren Applikationen und investiert Zeit und Geld in die Individualisierung[7] der Analysedaten auf Webtrekk. Mittlerweile sind knapp 30 Custom-Parameter[8] auf Webtrekk angelegt um genau die Daten zu tracken, die zur Analyse des Verhaltens der User auf ihrer Applikationen relevant sind. Dadurch entsteht ein fast „gläsernen User“. Dieses Wissen könnte zum Vorteil des Users eingesetzt werden, indem es in der Suche verwendet wird.

[6] Bezeichnet Springer Nature interne Kunden, die ein Interesse am Ergebnis der White Label Applikation haben
[7] Mit Individualisierung wird die Speicherung eigener Parameter bezeichnet
[8] Individuell erzeugte Parameter für Berichte und Analysen

1.3 Ziel der Arbeit

1.3.1 Suchoptimierung durch Click-Trough-Daten

In dieser Arbeit werden wir untersuchen, ob mithilfe der von Springermedizin gesammelten Click-Trough-Daten dessen Suche verbessert werden kann. Im Idealfall widerspiegeln die gesammelten Click-Trough-Daten der Suchresultate die Userrelevanz der einzelnen Dokumente[9].

Annahmen

Wir gehen dabei von folgenden Annahmen aus. Relevante Dokumente sind wichtiger als nicht relevante Dokumente. Eine Suchergebnis ist dann gut, wenn die relevanten Ergebnisse in der verwendeten Hierarchie vor den nicht relevanten Ergebnisse auftauchen.

1.3.2 Abbildung auf das Springermedizin-Umfeld

Potential von Userrelevanzen in der Suchoptimierung analysieren

Die Analyse von User-Tracking-Daten bietet viel Potential bezogen auf Userrelevanzen. Sind anhand des hier umgesetzten Lösungsansatzes Verbesserungen in der Qualität der Suche zu verzeichnen, möchte Springermedizin in Zukunft vermehrt User-Tracking-Daten in die Suche einfließen lassen. Diese Arbeit könnte dann als Fundament für weitere Lösungsansätze dienen.

Bekanntes und wirkungsvolles Information Retrieval Verfahren

Suchoptimierung mittels Userrelevanz ist ein bekanntes und nicht triviales, aber relativ wirkungsvolles Information Retrieval Verfahren (siehe [EA06]). Seit Mitte der 2000er Jahre wird mithilfe dieses Verfahrens versucht, Suchmaschinen zu verbessern. Aus dieser Zeit stammen auch die ersten Ansätze um mithilfe von Click-Trough-Daten die Userrelevanz der Suchergebnisse zu berechnen (siehe [TJ05]).

Lösungsansatz basierend auf Click-Trough-Daten aus Webtrekk

Springermedizin führt ein eigenes Tracking der User durch und verwendet auf Webtrekk selbst definierte Tracking-Parameter. Dadurch hängt die Wahl des in dieser Arbeit zu untersuchenden Lösungsansatzes und dessen Umsetzung stark von den durch Webtrekk gegeben Analyse-Daten ab.

Anwendung auf Springermedizin-Umfeld

Bei der Verwendung von Userrelevanzen in der Suche handelt es sich um ein bekanntes und gut erforschtes Problem. Wir werden in dieser Arbeit versuchen, einen bestehenden Lösungsansatz (Position-based Modell) auf das Springermedizin-Umfeld abzubilden. Die Herausforderung wird hierbei die Adaptierung des Lösungsansatzes auf das Springermedizin-Umfeld sein.

[9] Als Dokumente werden die einzelnen Suchresultate bezeichnet

Was wird in dieser Arbeit nicht behandelt?

Durch den vorgegebenen Zeitraum für die Erstellung dieser Bachelorarbeit bedingt, werden wir den Lösungsansatz so wählen, dass er mit den Gegebenheiten bei Springermedizin sinnvoll und in diesem Zeitrahmen realistisch implementiert werden kann. Wir werden daher in dieser Arbeit keine Gegenüberstellung mit anderen Lösungsansätzen machen.

Bei der Umsetzung des Lösungsansatzes konzentrieren wir uns auf die Implementation des Algorithmus zur Berechnung der Userrelevanz. Die semantische Aufschlüsselung von Suchtermen ist nicht Kern dieser Arbeit. Die semantische Aufschlüsselung des Suchterms[10] zur Analyse der Webtrekk-Daten enthält darum keine Gewichtungen der Relationen zwischen Webtrekk-Daten und Suchterm. Alle Relationen werden gleich gewichtet.

1.4 Methodik

1.4.1 Einführung

Wie in Kapitel 1.3.1 angesprochen, wollen wir das Klick-Verhalten der User in der Suche analysieren um mithilfe der daraus berechenbaren Userrelevanzen die Suchergebnisse zu verbessern. Dieses Klick-Verhalten können wir aus den Click-Trough-Daten lesen. Um mit Click-Trough-Daten arbeiten zu können, müssen wir zuerst verstehen, was Click-Trough-Daten sind und wie sie entstehen.

1.4.2 Click-Trough-Daten verstehen

Was sind Click-Trough-Daten und wie entstehen diese?

Click-Trough-Daten sind Tracking-Daten. Tracking-Daten entstehen durch die Interaktion zwischen dem User der Applikation und der Applikation selbst. Sie verfolgen das Verhalten der User auf der Applikation und speichern diese in einer Datenbank, in unserem Fall in Webtrekk ab. Die für uns interessanten Tracking-Daten entstehen, wenn der User auf der Suche von Springermedizin ein Anfrage stellt und darauf folgend, ein Element aus dem Suchresultat anklickt.

Wie werden die Click-Trough-Daten in Webtrekk gespeichert?

Die Speicherung der Daten auf Webtrekk übernimmt die Springermedizin-Applikation. Führt ein User eine Suche durch und klickt dabei ein Resultat an, sendet die Springermedizin-Applikation die Tracking-Informationen an Webtrekk. Die Tracking-Daten für diese Aktion, setzen sich zusammen aus der Suchanfrage, dem Zeitpunkt der Suche, den Userdaten, der angeklickten Position im Suchresultat und den Dokumentinformationen zum angeklickten Dokument. Aus diesen Daten werden die Click-Trough-Daten erstellt, mithilfe denen wir die Userrelevanz berechnen werden.

[10] Als Suchterm wird eine Suchanfragen bezeichnet

Wie können wir Click-Trough-Daten aus Webtrekk lesen?

Webtrekk ist ein Analysetool. Das heißt für uns, wir können nicht direkt auf die Datenbank mit den Tracking-Daten zugreifen. Um die Tracking-Daten lesen zu können, müssen wir eine Analyse auf Webtrekk ausführen. Mithilfe dieser Analyse können wir uns die Click-Trough-Daten so zusammenstellen lassen, wie wir sie für die Berechnung der Userrelevanz benötigen.

Die Click-Trough-Daten bestehen aus einzelnen Click-Trough-Rates. Eine Click-Trough-Rate zeigt die Anzahl der Klicks, die zu einer bestimmten Suchanfrage auf ein bestimmtes Dokument gemacht wurden und auf welcher Position im Suchresultat sich dieses Dokument dabei befunden hat. Die Webtrekk-Analysen geben uns eine Sammlung von Click-Trough-Rates zurück. Wir können bei diesen Analysen die Click-Trough-Rates nach Suchbegriffen oder auch Suchtermen filtern und den Zeitraum mitgeben, in welchen die Suchanfragen durchgeführt wurden. Des weiteren gibt es die Möglichkeit weitere Filter wie die Anzahl zurückzugebender Click-Trough-Rates oder auch den „Login-Status[11] des Users“ zu setzen.

Wie sehen die Click-Trough-Daten aus?

Eine Beispiel für eine Click-Trough-Rate wie sie von einer Webtrekk-Analyse ausgespielt wird, sieht wie folgt aus:

Click-Trough-Rate	Anzahl Klicks
searchresult-1.Course.chronische Dyspnoe bei Erwachsenen.10621768.chronische Dyspnoe	1

Hier die Aufschlüsselung der Click-Trough-Rate:

Position	Dokumenttyp	Titel	ID	Suchterm
searchresult-1	Course	chronische Dyspnoe bei Erwachsenen	10621768	chronische Dyspnoe

Die Click-Trough-Rate lässt sich wie folgt lesen. In diesem Beispiel haben die User mit der Suchanfrage „chronische Dyspnoe[12]“ gesucht. Dabei haben sie das Dokument mit der ID 10621768 angeklickt. Dieses hat sich dabei auf der Position 1 der Suchresultate gefunden. Es wurde insgesamt einmal angeklickt in der gesuchten Periode.

1.4.3 Reranking mittels Click-Trough-Rate

Im vorherigen Abschnitt haben wir gelernt wie Click-Trough-Daten entstehen und wie sie zu lesen sind. Nun können wir mit diesem Wissen die Userrelevanzen der Dokumente im Suchresultat zur Suchanfrage berechnen. Mithilfe der berechneten Userrelevanzen werden wir dann ein *Reranking*[13] der Suchresultate durchführen. So wollen wir die Userrelevanz in die Suche einbinden. Die Vorgehensweise dazu sieht wie folgt aus.

[11] Mit Login-Status wird zwischen einem zum Zeitpunkt der Suche auf der Springermedizin-Applikation angemeldeten und nicht angemeldeten User unterschieden
[12] Als Dyspnoe wird eine unangenehm erschwerte Atemtätigkeit bezeichnet
[13] Mit Reranking bezeichnen wie die Umsortierung einer Liste von Suchresultaten

Suchterm semantisch aufschlüsseln

Um die Userrelevanz berechnen zu können müssen wir zunächst die relevanten Click-Trough-Daten filtern. Click-Trough-Daten müssen nicht immer mit dem vollständigen Suchterm in Relation stehen. Sie können auch nur mit einem Wort des Suchterms oder einem Synonym des Wortes in Relation stehen. Wir müssen darum den Suchterm semantisch aufschlüsseln um alle relevanten Click-Trough-Daten filtern zu können.

Aufbereitung Click-Trough-Daten

Können wir alle relevanten Click-Trough-Daten zu einer Suchanfrage filtern, müssen lernen wie wir diese richtig aufbereiten, um die Userelevanzen berechnen zu können.

Userrelevanz in Suchprozess einbinden

Result-Reranking mittels PBM Algorithmus

Vergessen der alten Daten

1.5 Gliederung und Aufbau

Der Lösungsansatz und deren Grundlagen

Im ersten Kapitel wurde der zu untersuchenden Lösungsansatz vorgestellt. Dabei sind wir auf die Hintergründe dieser Arbeit und die Vorgehensweise eingegangen. Im zweiten Kapitel (Grundlagen) folgt die Theorie des beschriebenen Lösungsansatzes. Hier werden wir uns auf die fachlichen Grundlagen konzentrieren.

Umsetzung des Lösungsansatzes

In Kapitel 3 (Reranking mittels Click-Trough-Rate Ergebnis) werden wir die in Kapitel 1.4 angesprochene Methodik verfeinern und detailliert die Vorgehensweise bei der Umsetzung diskutieren. Die Umsetzung selbst folgt dann in Kapitel 4 (Implementierung).

Erkenntnisse verarbeiten

Um zu prüfen ob der umgesetzte Lösungsansatz die erhofften Verbesserungen erzielt, werden wir diesen in Kapitel 5 (Evaluation und Auswertung) in einer Evaluation mit der bisherigen Springermedizin-Suche vergleichen. Aufgrund der resultierenden Erkenntnisse werden wir in Kapitel 6 ein Fazit ziehen können und einen Ausblick auf mögliche zukünftige Arbeiten geben.

2 Grundlagen

2.1 Grundbegriffe

2.1.1 Semantik von User-Interaktionen

Nehmen wir als Beispiel die im vorherigen Abschnitt besprochene Suchanfrage „chronische Dyspnoe“. Würde eine andere Person stattdessen nach dem gleichbedeutenden Suchterm „konstanter Atemnot“ suchen, können die Ergebnisse beider Suchresultate und somit auch deren Click-Trough-Daten gleichermaßen relevant sein.

2.1.2 Userrelevanz mittels Click-Trough-Rate (CTR)

2.1.3 Result-Reranking mittels PBM Algorithmus

2.2 Zusammenfassung

3 Reranking mittels Click-Trough-Rate Ergebnis

3.1 Probleme und Lösungsansätze

3.2 Methodik

3.2.1 Einführung

3.2.2 Suchterm Segmentierung

Suchterm semantisch aufschlüsseln mittels Segmentierung

Ein Suchterm kann aus mehr als einem Wort bestehen. Wir müssen darum davon ausgehen, dass möglicherweise jedes Wort des Suchterms in unterschiedlichen Suchanfragen verwendet wird. Außerdem kann auch nach Synonymen eines Wortes gesucht werden. Der Suchterm muss darum semantisch aufgeschlüsselt werden, um alle relevanten Click-Trough-Daten berechnen zu können. Die Click-Trough-Daten sind dann relevant, wenn mindestens ein Wort des aufgeschlüsselten Suchterms in Relation zu diesen Daten steht. Von einer Gewichtung der Relation wird abgesehen.

Die Auftrennung des Suchterms in die einzelne Worte können wir mithilfe einer Segmentierung[1] durchführen. Hier könnten wir uns überlegen, zusätzlich mit Stoppwörtern[2] nicht relevante Wörter aus dem Suchterm zu entfernen. Dieses Verfahren macht aber im Springermedizin-Kontext keinen Sinn. Wie in Kapitel 1.2 angesprochen, suchen die User der Springermedizin-Applikation oft mit einschlägig, fundierten Fachbegriffen. Wir gehen darum davon aus, dass alle Wörter des verwendeten Suchterms für das Suchergebnis relevant sind. Diese Erkenntnis basiert auf Aussagen der Redakteure von Springermedizin und Webtrekk-Analysen der meist gesuchtesten Suchtermen der letzten Monate. Auch sind Stoppwörter veraltet und werden in modernen Information Retrieval Verfahren nicht mehr eingesetzt. Wir verzichten darum auf den Einstatz von Stoppwörtern.

Suchterm semantisch erweitern mittels Thesaurus

Für die semantische Erweiterung eines Suchwortes wird ein Thesaurus[3] benötigt. Die Erweiterung umfasst zum Suchterm gleichbedeutende Begriffe (Synonyme), sehr ähnliche Begriffe (Narrow Terms), ähnliche Begriffe im weiteren Sinne (Broader Terms) und verwandte Begriffe (Related Terms).

[1] Bezeichnet die Aufteilung in Abschnitte, in diesem Fall in einzelne Worte
[2] Stoppwörter sind Wörter, die sehr häufig auftreten und für gewöhnlich keine Relevanz für den Dokumentinhalt besitzen
[3] Als Thesaurus wird ein strukturiertes Verzeichnis von Begriffen, welche allesamt in irgendeiner Beziehung stehen bezeichnet

Springer Nature besitzt einen Webservice mit welchem auf den Thesaurus *Unified Medical Language System* (UMLS) (siehe [Uml]) zugegriffen werden kann.

3.2.3 Aufbereitung Click-Trough-Daten

Gewichtung der Click-Trough-Daten

Durch die semantische Aufschlüsselung des Suchtermes haben wir verschieden starke Relationen zwischen Click-Trough-Daten und dem Suchterm. Die Gewichtung der Stärke dieser Relation ist aber nicht Kern dieser Arbeit. Wir gehen darum davon aus, dass unabhängig der stärke der Relation zum Suchterm, alle Click-Trough-Daten eine gleiche Relevanz besitzen.

Komplexe Auswertungen der Click-Trough-Daten nicht möglich

Die Webtrekk-Analysen bieten uns nur beschränkte Informationen zum Klickverhalten der User. Wichtige Informationen wie die Verweildauer auf einem Dokument oder ob nach diesem Dokument ein weiteres Dokument zum gleichen Suchterm angeklickt worden ist, lassen diese Analysen nicht zu. Da diese Informationen für komplexe Auswertungen der Click-Trough-Daten nicht ausreichen, können wir in dieser Arbeit *Feedback-Strategien* für die Click-Trough-Rate Auswertung, wie in [TJ05] beschrieben, nicht verwenden. Stattdessen müssen wir davon ausgehen, dass jeder Klick auf ein Dokument relevant ist.

3.2.4 Userrelevanz in Suchprozess einbinden

Es gibt drei mögliche Eingriffspunkte während des Suchprozesses, um Userrelevanzen in der Springermedizin-Suche zu verwenden. Möglich wäre die Verwendung der Userrelevanzen in der Aufbereitung der Suchanfrage auf der Springermedizin-Applikation. Denkbar wäre auch, die Berechnung der Userrelevanzen in den Suchindex der Solr einzubauen. Die dritte Variante wäre die Verwendung der Userrelevanzen in der Aufbereitung der Suchresultate der Springermedizin-Applikation. Eine davon, wollen wir in dieser Arbeit untersuchen.

Ansatz: Suchindex-Erweiterung in der Solr-Suche

Um die Userrelevanzen direkt in die Solr einzubeziehen gibt es zwei Varianten. Wir können das Schema des Suchindexes über die Schema API (siehe [Sch]) erweitern und alle Einträge neu indexieren, oder wir ergänzen den Index um ein externes Feld (ExternalFileField) (siehe [Ext]).

Beide Lösungsansätze ergeben nur bei der Speicherung einer einfachen Click-Count Popularität[4] Sinn. Diese genügen allerdings den hier gegebenen Anforderungen nicht, da die Click-Trough-Rate abhängig vom Suchterm ist. Der erste Lösungsansatz ist zudem besonders heikel, weil bei jeder Änderung des Click-Count-Wertes, das Dokument in der Solr neu indexiert werden.

[4] Kennzahl für alle Klicks auf ein Dokument unabhängig des Suchterms

Ansatz: Aufbereitung der Suchanfrage

Die Solr-Suche bietet eine Boost-Funktion namens *DisMax Query Parser* (siehe [Dis]). Mit dieser können basierend auf Feldwerten, einzelne Dokumente besser im Suchergebnis positioniert werden. Die Boost-Funktion müssten wir in den Aufbau der Suchanfrage für die Suche auf der Springermedizin-Applikation einbauen. Dieser Ansatz beinhaltet einige Gefahren die wir beachten müssen.

Dazu zählen beispielsweise die Abhängigkeiten von anderen Boost-Faktoren[5]. Alle Boost-Faktoren hängen voneinander ab und müssten bei jeder Ergänzung um neue Faktoren normalisiert werden, um kein „über-Boosting“[6] einzelner Faktoren zu riskieren. Zudem besteht die Gefahr des „blinden Boosting“ von Dokumenten. Die Solr-Relevanzberechnung ist komplex und der Einfluss des „Boosting“ in die Solr-Relevanzberechnung schwer erkennbar. Auch hat Springermedizin bereits sehr schlechte Erfahrungen mit „Boosting“ gemacht und bevorzugt einen Lösungsansatz ohne „Boosting“.

Ansatz: Aufbereitung der Suchresultate anhand eines Klick-Modell basierten Algorithmus

Die dritte Möglichkeit ist bei der Aufbereitung der Suchresultate aus der Solr-Suche einen Klick-Modell (siehe [Chu+15]) basierten Algorithmus zu verwenden. Dieser soll die Suchergebnisliste analysieren, die Userrelevanzen der Dokumente berechnen und die Liste neu sortieren.

Diese Lösung können wir relativ einfach in die Springermedizin-Applikation integrieren, ohne die restliche Suchlogik[7] zu beeinflussen. Hierbei müssen wir jedoch beachten, dass die Solr durch die Pagination-Funktion (siehe [Pag]) nur die Top-N-Ergebnisse (bei Springermedizin sind es 20 Ergebnisse) zurückgibt. Diese Logik liegt in der Springermedizin-Applikation im Aufbau der Suchanfrage. Daher können wir diese selber steuern und uns statt 20 beispielsweise die nächsten 100 Ergebnisse zurückgeben lassen. Am Ende filtern wir die ersten 20 Ergebnisse und stellen diese dar. Außerdem wissen wir bei diesem Lösungsansatz, in welcher Reihenfolge die Ergebnisse aus der Solr zurückgegeben werden. Wir kennen die Dokumente und deren Rang. Dadurch haben wir hilfreiches Zusatzwissen, welches wir in den Klick-Modell basierten Algorithmus einfließen lassen können.

Entscheidung für den Ansatz der Aufbereitung der Suchresultate anhand eines Klick-Modell basierten Algorithmus

Wägen wir die besprochenen Fakten ab, wirkt der Ansatz mit der Aufbereitung der Suchresultate durch einen Klick-Modell basierten Algorithmus am sinnvollsten. Wir wissen bei diesem Ansatz, welche Dokumente für die Userrelevanz-Berechnung überhaupt in Frage kommen. Zudem kennen wir alle Einfluss-Faktoren für den Algorithmus und wir sind unabhängig von der Suchlogik auf der Solr. Dadurch können wir Änderungen in unserer Logik schnell und einfach implementieren.

[5] Die Solr besitzt eine Boosting-Funktion, um bestimmte Wertübereinstimmungen in der Suche höher gewichtet zu können
[6] Bezeichnet die über-priorisierte Bewertung einzelner Faktoren
[7] Dazu gehört die Aufbereitung der Suchanfrage für die Solr und die Suche auf der Solr

3.2.5 Result-Reranking mittels PBM Algorithmus

Klick-Wahrscheinlichkeit mit Position-based Modell berechnen

Mithilfe der Click-Trough-Daten aus Webtrekk, können wir zwei wichtige Informationen zu jedem Suchterm ermitteln. Wir wissen welche Dokumente und welche Positionen im Suchresultat angeklickt worden sind. Zudem kennen wir die Reihenfolge der Dokumente im Suchresultat der Solr.

Ein Ansatz der genau auf diesen Informationen aufbaut, ist das *Position-based Modell* (PBM) (siehe [Chu+15]). Dieses berechnet die Wahrscheinlichkeit dafür, dass ein User ein Dokument wirklich genau analysiert, bevor er es anklickt. Es setzt sich aus zwei Wahrscheinlichkeiten zusammen. Die Wahrscheinlichkeit für einen Klick auf die Position im Suchresultat und die Wahrscheinlichkeit für einen Klick auf das Dokument. Diesen Ansatz werden wir in dieser Arbeit implementieren.

Verhältnis zwischen den Klick-Wahrscheinlichkeiten abhängig der Position im Suchresultat definieren

Aus eigener Erfahrung wissen wir, dass die ersten Dokumente im Suchresultat immer zuerst gesehen werden. Die dahinter gelisteten Dokumente werden fortlaufend analysiert. Wir sollten darauf achten, dass je *schlechter* der Rang des angeklickten Dokumentes im Suchresultat der Solr ist, desto *höher* das Relevanzfeedback zu bewerten ist.

Smoothing Faktor in Position-based Modell

Wir wissen dass eine Wahrscheinlichkeit einen Wert zwischen 1 und 0 besitzt. Dadurch können Nullwerte entstehen. Das PBM multipliziert die Positions- und Dokument-Wahrscheinlichkeit miteinander, um die Klick-Wahrscheinlichkeit zu berechnen. Wir müssen aber davon ausgehen, dass es Dokumente geben kann, deren Rang nie angeklickt worden ist und umgekehrt.

Multiplikationen mit Null ergeben immer einen Nullwert. An dieser Stelle führen wir einen *Smoothing-Faktor* ein. Der Smoothing-Faktor soll zwei Probleme lösen. Zum einen wollen wir einen Wahrscheinlichkeitswert trotz der Multiplikation mit Null beachten. Zum anderen wollen wir die im vorherigen Absatz beschriebene Gewichtung abhängig des Relevanzfeedbacks in den Algorithmus einbeziehen. Wir transformieren dazu das Produkt der beiden Wahrscheinlichkeiten in eine gewichtete Summe, dem sogenannten *Weighted Moving Average*, dessen Gewichte sich zu Eins aufsummieren. Diese Gewichte sind die Smoothing-Faktoren, weshalb das Verfahren zählt zu den Smoothing-Algorithmen zählt.

3.2.6 Vergessen der alten Daten

Ein Algorithmus zur Berechnung von Wahrscheinlichkeiten muss sich ein gewisses Grundwissen aneignen. Dies geschieht üblicherweise durch Trainingsdaten. Genauso muss er alte Daten wieder vergessen können, um overfitting[8] zu vermeiden.

[8] Überanpassung des Algorithmus durch zu viele (falsche oder veraltete) Daten

Durch Webtrekk ist kein komplexer Lern-Algorithmus notwendig

Durch Webtrekk haben wir eine Wissensbasis, die sich stetig und zeitnah aktualisiert. So muss der Algorithmus nicht stetig neues Wissen lernen und altes vergessen, sondern er kann direkt diese Wissensbasis zugreifen. Dies geschieht, indem zur Laufzeit[9] Analysen gegen Webtrekk über eine frei definierbare Periode gemacht werden. Dadurch kann *overfitting* vermieden werden. Deshalb verwenden wir keinen komplexen Lern-Algorithmen wie in [EA06] vorgestellt.

Die Klick-Wahrscheinlichkeit ist kein absoluter Wert für Userrelevanz

Nun könnten wir die Klick-Wahrscheinlichkeit als absoluten Wert für die *Userrelevanz* betrachten. Dies wäre jedoch falsch, wir müssen davon ausgehen, dass viele User der Qualität der Suchmaschine vertrauen. Diese betrachten die Top-Suchresultate als die relevanten Suchresultate. Denkbar wäre auch, dass sie unabsichtlich das falsche Dokument anklicken.

Overfitting vermeiden

Um dem entgegenzuwirken und ein *overfitting* zu vermeiden, darf der Algorithmus nicht immer anschlagen. Wir müssen sicherstellen, dass vereinzelt zufällige Dokumente in den „Top-Suchresultaten“ angezeigt werden. So können auch andere Dokumente in den Fokus des Users gerückt werden. Das System fährt sich dadurch nicht auf falschen Annotationen fest.

Zusätzliche Varianz durch Zufallsfaktor

Mithilfe eines Zufallsfaktors kann eine solche Varianz in den Klick-Modell basierten Algorithmus gebracht werden. Wie bereits weiter oben erwähnt, werden viele Suchresultate nie und deren Rang selten bis gar nicht angeklickt. Sie haben darum keine Click-Trough-Daten. Deren Klick-Wahrscheinlichkeit ist entweder Null oder sehr klein. Der Zufallsfaktor soll darum nur leichte Einflüsse in die Klick-Wahrscheinlichkeitsberechnung haben. Auch hier können wir wieder mit dem oben eingeführten *Weighted Moving Average* (siehe [Kar]) arbeiten.

3.3 Der PBM-Algorithmus

3.4 Zusammenfassung

[9] Unter Laufzeit wird in diesem Fall der Zeitpunkt der direkte Abfrage während der Suchanfrage bezeichnet

4 Implementierung

5 Evaluation und Auswertung

5.1 Einführung

Suchvarianten mithilfe eines Evaluationssystems vergleichen

Das große Kernproblem der Überprüfung der Verbesserungen durch den untersuchten Lösungsansatz wird das Messen der Qualität der erzielten Suchergebnisse sein. Mithilfe einer Evaluation wollen wir messen, wie gut die Suchergebnis-Qualität der aktuellen Springermedizin-Suche im Vergleich zur im Zuge dieser Arbeit entwickelten Lösung ist.

Ziel der Evaluation

Die Evaluation soll Informationen darüber liefern, wie viel Verbesserung der neue Lösungsansatz bringt. Aus den Ergebnissen wollen wir erkennen, an welchen „Schrauben“ etwas gedreht werden muss, damit die Suche wirklich gute Ergebnisse aus Sicht der User bringt.

5.2 Aufbau der Analyse

5.2.1 Datengrundlage

Filterung der nutzbaren Daten mittels Cohens Kappa

Um die Zuverlässigkeit der Relevanzbewertungen zu messen, werden wir die gleichen Suchterme jeweils von zwei fachlichen Experten bewerten lassen. Das meist verwendete Maß zur Bewertung der Übereinstimmungsgüte ist der *Cohens Kappa Koeffizient* (siehe [Gro+07]). Diese Zahl misst den Anteil übereinstimmender Bewertungen. Hierbei können aber auch zufällige Übereinstimmungen entstehen. Der Cohens Kappa Koeffizient korrigiert das Maß an Übereinstimmung um diesen Zufallsfaktor. Anhand der Auswertungen werden wir ein Mindestmaß der Übereinstimmungsgüte definieren. Die darunter liegenden Bewertung werden wir in der Auswertung ignorieren.

5.2.2 Metrik

Evaluationsdaten mittels NDCG-Algorithmus auswerten

Um das Qualitätsmaß der beiden Suchen vergleichen zu können werden wir den Bewertungsalgorithmus *NDCG* (siehe [Wan+13]) einsetzen. Dieser geht davon aus, dass besser positionierte Suchergebnisse eine höhere Relevanz als schlechter positionierte haben. Der NDCG vergleicht die Reihenfolge der

Relevanzbewertungen der Suchergebnisse mit der idealen Reihenfolge derselben Relevanzbewertungen. Im Idealfall entspricht die Reihenfolge der Suchergebnisse der Relevanz der Suchergebnisse.

Qualitätsmaß einer Suchvariante bestimmen

In der Evaluation werden zu jedem Suchterm zwei Bewertungen für die Springermedizin-Suche und zwei Bewertungen für die Suche mit dem hier zu untersuchenden Lösungsansatz abgegeben. Um das Qualitätsmaß einer Suchvariante zu einem Suchterm zu bestimmen, berechnen wir den NDCG der beiden Bewertungen. Nehmen wir den Mittelwert der beiden resultierenden NDCG-Werte, erhalten wir den effektiven NDCG-Wert. Die NDCG-Werte der beiden Suchen können wir dann miteinander vergleichen.

5.2.3 Vorgehen

Evaluationssystem aufbauen

Um eine Evaluation durchführen zu können, müssen wir eine passende Testumgebung aufbauen. Diese besteht aus einem Evaluationssystem, einer Instanz der aktuellen Springermedizin-Applikation und einer Instanz des neu implementierten Lösungsansatzes. Auf dem Evaluationssystem sollen fachliche Experten (Redakteure von Springermedizin) die Relevanz der Suchergebnisse der beiden Suchmaschinen vergleichen. Dazu sollen die jeweils besten 10 Suchergebnisse nach Relevanz zum Suchterm bewertet werden. Der Ergebnisse werden in einer Datenbank gespeichert, um sie später auszuwerten.

Evaluationssystem auswerten

Nach Ablauf der Evaluationsphase werden wir die Evaluations-Daten auswerten. Die Auswertung der Daten findet direkt im Evaluationssystem statt.

Dazu werden die Daten aus der Datenbank gelesen und mit dem Cohens Kappa Koeffizienten die nutzbaren Daten gefiltert.

5.2.4 Durchführung

Verschiedene Varianten des neuen Lösungsansatzes werden evaluiert

Der in dieser Arbeit zu untersuchende Lösungsansatz kann verschieden konfiguriert werden. Wir können den Einfluss des Zufallsfaktors bestimmen. Um verschiedene Konstellationen testen zu können, werden wir mit zwei verschiedenen Werten für den Einfluss des Zufallsfaktor evaluieren. Die Click-Trough-Daten von an der Applikation angemeldeten Benutzern können wir von den Click-Trough-Daten von anonymen Benutzern unterscheiden.

Aus den beiden Einflusswerten des Zufallsfaktors und der Unterscheidung zwischen angemeldeten und anonymen Benutzern, ergeben sich vier Konstellationen, die evaluiert werden können. Jeder Konstellation werden wir jeweils 25 Prozent der Suchterme zuteilen. Mithilfe des Evaluationssystems werden wir die Zuteilung der Suchterme zufällig generieren lassen.

5.3 Auswertung der Suchergebnis-Qualität

5.3.1 Quantitative Auswertung

5.3.2 Diskussion

5.4 Zusammenfassung

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

6.2 Ausblick

Anhang

7

Literatur

[Chu+15] Aleksandr Chuklin, Ilya Markov und Maarten de Rijke. *Click Models for Web Search*. Morgan & Claypool, 2015 (siehe S. 10, 11).

[Dis] *The DisMax Query Parser - Apache Solr Reference Guide* (siehe S. 10).

[EA06] Susan Dumais Eugene Agichtein Eric Brill. *Improving Web Search Ranking by Incorporating User Behavior Information*. Report. Microsoft, 2006 (siehe S. 3, 12).

[Ext] *ExternalFileField (Solr 4.8.0 API) - Apache Lucene* (siehe S. 9).

[Gro+07] Grouven, Bender, Ziegler und Lange. „Der Kappa-Koeffizient“. In: Thieme, 2007, S. 111–128 (siehe S. 14).

[Kar] Marzena Narodzonek Karpowska. *Smoothing methods* (siehe S. 12).

[Pag] *Pagination of Results - Apache Solr Reference Guide* (siehe S. 10).

[Sch] *Schema API - Apache Solr Reference Guide - Apache Software* (siehe S. 9).

[Sme] *Springer Medizin – das Fachportal für Ärzte* (siehe S. 2).

[Sol] *Solr - Apache Lucene* (siehe S. 1).

[TJ05] Bing Pan Thorsten Joachims Laura Granka. *Accurately Interpreting Clickthrough Data as Implicit Feedback*. Report. Random, 2005 (siehe S. 3, 9).

[Uml] *Unified Medical Language System (UMLS)* (siehe S. 8).

[Wan+13] Yining Wang, Liwei Wang, Yuanzhi Li u. a. „A Theoretical Analysis of NDCG Type Ranking Measures“. In: *CoRR* abs/1304.6480 (2013) (siehe S. 14).

[Web] *Webtrekk* (siehe S. 1).

Abbildungs-Verzeichnis

Tabellen-Verzeichnis

Sourcecode-Verzeichnis