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 Marken und zudem der größte Verlag für Wissenschaftsbücher. Für Springer Nature ist es darum wichtig, auf ihren 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 Suche nach 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 auf ihrer Suche, lässt dieses Wissen jedoch 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 inhouse entwickelte White Label Applikation[2]. Die White Label Applikation verwendet *Apache Solr*[3] als Suchplattform. Die Solr dient hierbei als eine der Schnittstellen zwischen dem Content-Pool von Springer Nature und der Core-Applikation. Bei dem vom Content-Pool gelieferten Content, handelt es sich um vom Springer Nature-Verlag publizierte Zeitschriften, Artikel, Bücher, Chapters und redaktionelle Inhalte.

User-Tracking mit Webtrekk

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

[1] Nummer der Zeitschriftenausgabe, in der sich der Artikel befindet.
[2] Eine White Label Applikation ist eine wiederverwendbare und agil erweiterbare Applikation
[3] http://lucene.apache.org/solr
[4] https://www.webtrekk.com
[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:

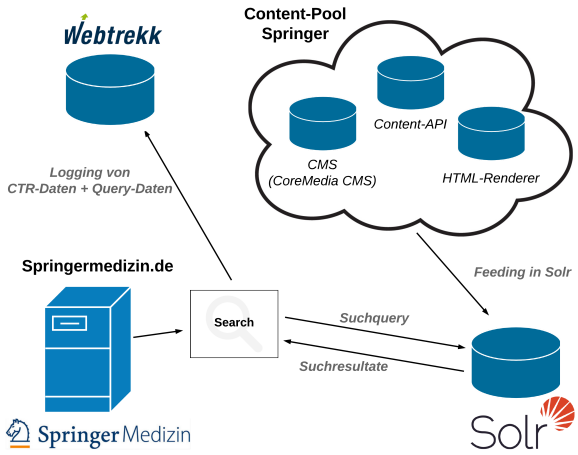

Abb. 1: Aufbau der Suche bei Springer Nature

1.2 Das große Problem

Keine Userrelevanz in der Suche

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*[7]. Springermedizin ist ein Fortbildungs- und Informationsportal für Ärzte. Mithilfe von Webanalysten und Webtrekk versucht Springermedizin das Marketing ihres 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 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 der Analysedaten auf Webtrekk. Mittlerweile sind knapp 30 Custom-Parameter[8] auf Webtrekk angelegt um genau die Daten zu tracken, die sie über das Verhalten der User auf ihrer Applikationen wissen wollen. Dadurch ensteht ein fast „gläsernen User“.

[6] Bezeichnet Springer Nature interne Kunden, die ein Interesse am Ergebnis der White Label Applikation haben
[7] https://www.springermedizin.de/
[8] Individuell erzeugte Parameter für Reports und Analysen

1.3 Ziel der Arbeit

1.3.1 Suchoptimierung durch Click-Trough-Daten

In dieser Arbeit werden wird untersuchen, ob mithilfe der von Springermedizin gesammelten Click-Trough-Daten deren Suche verbessert werden kann. Im Idealfall entsprechend die gesammelten Click-Trough-Daten der Suchresultate, der 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 vor den nicht relevanten Ergebnisse auftauchen.

1.3.2 Abbildung auf 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 Lösungsansatz Verbesserungen in der Qualität der Suche zu verzeichnen, möchte Springer 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 ganz 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 Lösungsansatzes und dessen Umsetzung stark von den durch Webtrekk gegeben Analyse-Daten ab.

Anwendung auf Springermedizin-Umfeld

Bei dieser Problemstellung handelt es sich um ein bekanntes und gut erforschtes Problem. Wir werden in dieser Arbeit versuchen, einen bestehenden Lösungsansatz 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 werden in dieser Arbeit nicht behandeln?

Durch den vorgegebenen Zeitraum 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. Auf semantische Analysen der Suchterme[10] zur Spezifizierung der Click-Trough-Daten verzichten wir. Diese sind nicht Kern dieser Arbeit.

1.4 Methodik

Wie in Kapitel 1.3.1 angesprochen, wollen wir das Klickverhalten der User[11] in der Suche analysieren um die Suchergebnisse zu verbessern.

1.4.1 Analyse der Click-Trough Daten

Userrelevante Dokumente durch Click-Trough-Rates identifizieren

Eine Möglichkeit die Userrelevanz eines Suchergebnisses zu bestimmen, ist die Verwendung der Click-Trough-Daten zur Berechnung der Userrelevanz. Durch die Click-Trough-Rates der Suchergebnisse, können wir die für die Nutzer der Suche relevanten Dokumente identifizieren.

Dazu analysieren wir, welche Dokumente im Suchergebnis zu welchen Suchtermen angeklickt worden sind. Wir gehen dabei davon aus, dass jedes Dokument, welches zu einer Suchanfrage angeklickt worden ist, eine gewisse Relevanz für die Suchanfrage hat.

Click-Trough-Daten kommen aus Webtrekk-Analysen

Als Wissensbasis für die Click-Trough-Daten dienen die Webtrekk-Analysen von Springermedizin. Wir können anhand dieser Analysen herausfinden, welches Dokument zu welchem Suchterm angeklickt worden ist. Wie oft es anglickt wurde und auf welcher Position[12] im Suchresultat, sich das Dokument dabei befunden hat.

Suchterm semantisch aufschlüsseln mittels Segmentierung

Ein Suchterm kann aus mehr als einem Wort bestehen. Wir müssen darum davon ausgehen, dass jedes Wort des Suchterms in unterschiedlichen Suchanfragen verwendet werden kann. Außerdem kann auch nach Synonymen eines Wortes gesucht werden. Der Suchterm muss darum semantisch aufgeschlüsselt werden, um alle relevanten Click-Trough-Daten analysieren zu können.

Die Aufschlüsselung des Suchterms in die einzelne Worte, können wir mithilfe einer Segmen-

[10] Als Suchterme werden Suchanfragen bezeichnet
[11] Als User werden die Nutzer der Springermedizin-Suche bezeichnet
[12] Mit Position wird der dargestellte Rang des Dokumentes im Suchresultat bezeichnet

tierung[13] durchführen. Hier könnten wir uns überlegen zusätzlich mit Stoppwörtern[14] zu arbeiten. Wie in Kapitel 1.2 angesprochen, suchen die User der Springermedizin-Applikation oft mit einschlägig, fundierten Fachbegriffen. 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.

Suchterm semantisch erweitern mittels Thesaurus

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

Springer Nature besitzt einen Webservice mit welchem auf den Thesaurus *Unified Medical Language System*[15] (UMLS) zugegriffen werden kann.

Komplexe Auswertungen der Click-Trough-Daten nicht möglich

Diese Analysen bieten uns jedoch 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 komplexe Auswertungen der Click-Trough-Daten nicht zulassen, 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.

Durch Webtrekk kein komplexer Lern-Algorithmus notwendig

Der Vorteil bei der Verwendung von Webtrekk ist, dass der Algorithmus nicht ständig neues Wissen lernen und altes vergessen muss. Der Algorithmus kann zur Laufzeit[16] direkt Analysen gegen Webtrekk über eine frei definierbare Periode machen kann kann. Dadurch kann *overfitting*[17] vermieden werden. Deshalb werden wir von komplexen Lern-Algorithmen wie in [EA06] vorgestellt, absehen.

1.4.2 Userrelevanz in Suche einbinden

Es gibt drei verschiedene Eingriffspunkte, um Userrelevanzen in die Springermedizin-Suche einzubauen. In der Aufbereitung der Suchquery auf der Springermedizin-Applikation. In den Suchindex der Solr. Oder in der Aufbereitung der Suchresultate der Springermedizin-Applikation.

Ansatz: Suchindex-Erweiterung in Solr-Suche

Um die Userrelevanzen direkt in die Solr einzubeziehen gibt es zwei Varianten. Wir können das Schema des Suchindexes erweitern über die Schema API[18] und alle Einträge neu indexieren. Oder wir ergänzen den Index um ein externes Feld[19] (ExternalFileField).

[13] Bezeichnet die Aufteilung in Abschnitte, in diesem Fall in einzelne Worte
[14] Stoppwörter sind Wörter, die sehr häufig auftreten und für gewöhnlich keine Relevanz für den Dokumentinhalt besitzen
[15] https://www.nlm.nih.gov/research/umls/
[16] Unter Laufzeit wird in diesem Fall die direkte Abfrage während der Suchanfrage bezeichnet
[17] Überanpassung des Algorithmus durch zu viele (falsche oder veraltete) Daten
[18] https://cwiki.apache.org/confluence/display/solr/Schema+API
[19] https://lucene.apache.org/solr/5_4_1/solr-core/org/apache/solr/schema/ExternalFileField.html

Beide Lösungsansätze machen nur bei der Speicherung einer einfachen Click-Count Popularität[20] Sinn. Sie können für diese Anforderungen nicht verwendet werden (Click-Trough-Rate abhängig vom Suchterm). Der erste Lösungsansatz ist zudem besonders heikel. Bei jeder Änderung des Click-Count-Wertes muss das Dokument in der Solr neu indexiert werden.

Ansatz: Aufbereitung der Suchquery

Die Solr-Suche bietet eine Boost-Funktion namens *DisMax Query Parser*[21]. Mit dieser können basierend auf Feldwerten einzelne Dokumente besser im Suchergebnis positioniert werden. Die Boost-Funktion müssten wir in den Aufbau der Suchquery für die Suche auf der Springermedizin-Applikation einbauen. Dieser Ansatz beinhaltet einige Gefahren die wir beachten müssen.

Abhängigkeiten von anderen Boost-Faktoren. Alle Boost-Faktoren hängen voneinander aber und müssen bei jeder Ergänzung um neue Faktoren normalisiert werden, um kein „über-Boosting“[22] einzelner Faktoren zu riskieren. Zudem besteht die Gefahr des „blinden“ Boosting von Dokumenten. Die Solr-Relevanzberechnung ist eine komplexe Berechnung und der Einfluss des Boosting in die Solr-Relevanzberechnung schwer erkennbar. Auch hat Springer bereits sehr schlechte Erfahrungen mit Boosting gemacht und bevorzugt einen Lösungsansatz ohne Boosting.

Ansatz: Aufbereitung der Suchresultate

Die dritte Möglichkeit ist ein Sortier-Algorithmus bei der Aufbereitung der Suchresultate aus der Solr-Suche. 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[23] zu beeinflussen. Hierbei müssen wir jedoch beachten, dass die Solr durch die Pagination-Funktion[24] nur die Top-N-Ergebnisse (bei Springermedizin sind es 20 Ergebnisse) zurückgibt. Diese Logik liegt in der Springermedizin-Applikation beim Aufbau der Suchquery. Daher können wir diese selber steuern und stattdessen uns 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 Dokumenten und deren Rang. Dadurch haben wir hilfreiches Zusatzwissen, welches wir in den Sortier-Algorithmus einfließen lassen können.

Entscheidung für den Ansatz der Aufbereitung der Suchresultate

Wägen wir die besprochenen Fakten ab, wirkt der Ansatz mit der Aufbereitung der Suchresultate durch einen Sortier-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.

[20] Kennzahl für alle Klicks auf ein Dokument unabhängig des Suchterms
[21] https://cwiki.apache.org/confluence/display/solr/The+DisMax+Query+Parser
[22] Bezeichnet die über-priorisierte Bewertung einzelner Faktoren
[23] Dazu gehört die Aufbereitung der Suchquery für die Solr und die Suche auf der Solr
[24] https://cwiki.apache.org/confluence/display/solr/Pagination+of+Results

Klick-Wahrscheinlichkeit mit Position-based Model 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 Model* (PBM) [Chu+15]. Dieses berechnet die Wahrscheinlichkeit, 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 der Klick-Wahrscheinlichkeiten abhängig der Position im Suchresultat

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

Smoothing Faktor in Position-based Model

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

Multiplikationen mit Null ergeben immer einen Nullwert. Wir müssen darum einen Faktor einbauen, um diese Wahrscheinlichkeitswerte trotz Nullwerten beachten zu können. Ein Ansatz den wir hier benutzen können, ist der Smoothing-Faktor.

1.4.3 Effektive Userrelevanz

Klick-Wahrscheinlichkeit 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. Oder sie klicken unabsichtlich das falsche Dokument an.

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 rücken. Das System fährt sich dadurch nicht auf falschen Annotationen fest.

Zusätzliche Varianz durch Zufallsfaktors

Mithilfe eines Zufallsfaktors kann eine solche Varianz in den Sortier-Algorithmus gebracht werden. Viele Suchresultate werden keine Click-Trough-Daten haben. Diese werden entweder Nullwerte oder nur eine sehr kleine Klick-Wahrscheinlichkeit haben. Der Zufallsfaktor soll darum nur leichte Einflüsse in die Klick-Wahrscheinlichkeitsberechnung haben. Auch hier können wir wieder mit einem Smoothing-Faktor arbeiten.

1.4.4 Evaluation

Suchvarianten mithilfe eines Evaluationssystems vergleichen

Das große Kernproblem der Überprüfung 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 uns Informationen darüber geben, 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.

Evaluationssystem aufbauen

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

Evaluationsdaten für die Auswertung mit Cohens Kappa-Koeffizient selektieren

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]). Cohens Kappa 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.

Evaluationsdaten mittels NDCG auswerten

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

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 beim Einfluss des Zufallsfaktor evaluieren. Bei den Click-Trough-Daten können wir zwischen an der Applikation angemeldeten Benutzern und anonymen Benutzern unterscheiden.

Aus den beiden Einflusswerten beim Zufallsfaktor 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.

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 besprechen. Die Umsetzung selbst, folgt dann in Kapitel 4 (Implementierung).

Erkenntnisse verarbeiten

Um zu prüfen ob der umgesetzte Lösungsansatz die erhofften Verbesserungen erzielt, werden wir in Kapitel 5 (Evaluation und Auswertung) in einer Evaluation diesen 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.

Literatur

[Chu+15] Aleksandr Chuklin, Ilya Markov und Maarten de Rijke. *Click Models for Web Search*. Morgan & Claypool, 2015 (siehe S. 7).

[EA06] Susan Dumais Eugene Agichtein Eric Brill. *Improving Web Search Ranking by Incorporating User Behavior Information*. Report. Microsoft, 2006 (siehe S. 3, 5).

[Gro+07] Grouven, Bender, Ziegler und Lange. „Der Kappa-Koeffizient“. In: Thieme, 2007, S. 111–128 (siehe S. 8).

[TJ05] Bing Pan Thorsten Joachims Laura Granka. *Accurately Interpreting Clickthrough Data as Implicit Feedback*. Report. Random, 2005 (siehe S. 3, 5).

Abbildungs-Verzeichnis

Tabellen-Verzeichnis

Sourcecode-Verzeichnis