Universität Regensburg

Beziehungen zwischen Dokumenten als Rankingfaktor für PubMed

Bachelorarbeit im Fach Medieninformatik am
Institut für Information und Medien, Sprache und Kultur (I:IMSK)

Vorgelegt von: Jonathan Brem
Adresse: Mühlstraße 16, 94256 Drachselsried
Matrikelnummer: 1632130
Erstgutachter: Prof. Dr. Christian Wolff
Zweitgutachter: Prof. Dr. Rainer Hammwöhner
Laufendes Semester: 6
Abgegeben am: 01.10.2015

Zusammenfassung

Diese Arbeit behandelt verschiedene Rankingstrategien für Dokumente in der medizinischen Fachdatenbank *PubMed*. Den Kern der Arbeit bildet ein neu entwickelter Rankingfaktor, der eine Kombination aus zwei weiteren anderen Rankingstrategien darstellt. Dies sind: die Verwendung des *PageRank* und die Suche nach Synonymen mithilfe von Ontologien.

Durch einen Vergleich von verschiedenen Rankingfaktoren mit dem neuen Faktor wird bestimmt, wodurch sich die Qualität der Suchergebnisse am meisten verbessern lässt. Die Faktoren werden einzeln und in Kombination mit anderen getestet.

Bei der Evaluation mit dem *TREC Genomics Track* von 2006 als Goldstandard kommt ein neu entwickeltes *Learning to Rank*-Verfahren zum Einsatz, das es erlaubt, die Systemvarianten hinsichtlich konkreter Evaluationsmetriken optimal zu konfigurieren. Die Evaluation zeigt, dass die Faktoren, die auf Beziehungen zwischen Dokumenten basieren, also auch der neu entwickelte kombinierte Faktor, keine Verbesserung der Suchergebnisse ermöglichen. Die gleichzeitige Verwendung mehrerer Rankingstrategien kann aber eine Verbesserung bringen und sollte häufiger und genauer untersucht werden.

Abstract

This paper examines ranking strategies for documents in the medical database *PubMed*. A newly developed ranking factor is at the core of the investigation. It combines the use of the *PageRank* and the search for synonyms in ontologies.

Different ranking strategies are compared with the new ranking factor in ordert o determine the factors that offer the most potential improvements for the search results. The factors are evaluated individually and in combination with one another.

A new method for *learning to rank* that allows optimizing the search system with respect to any evaluation metric is used in the evaluation. The 2006 *TREC Genomics Track* serves as the gold standard. The results indicate that the use of factors built on relationships between documents (including the newly developed factor) does not improve search results. Using multiple ranking strategies at once can improve the results and should be investigated more often and in greater detail.

Inhalt

Abbildungen

Begriffsklärungen

Diese Begriffe und Abkürzungen kommen im Verlauf der Arbeit vor und werden an dieser Stelle kurz erklärt.

- *IR* wird als Abkürzung für *Information Retrieval* verwendet.
- *Precision* und *Recall* sind zwei grundlegende Maßzahlen für die systemzentrierte Evaluation von Suchmaschinen:

  $$Precision = \frac{Zahl\ der\ für\ ein\ Thema\ relevanten\ Dokumente\ in\ den\ Suchergebnissen}{Zahl\ der\ Dokumente\ in\ den\ Suchergebnissen}$$

  $$Recall = \frac{Zahl\ der\ für\ ein\ Thema\ relevanten\ Dokumente\ in\ den\ Suchergebnissen}{Zahl\ der\ für\ das\ Thema\ relevanten\ Dokumente}$$

- Eine *Query* ist eine Suchanfrage an eine Suchmaschine.
- *Score* kann verschiedene Bedeutungen haben:
  a. der Wert, den eine Suchmaschine einem Dokument für eine Suchanfrage zuweist (Dokumente werden absteigend nach ihrem Score sortiert.)
  b. der Wert des in dieser Arbeit untersuchten *kombinierten Scores*
  c. die Ergebnisse einer systemzentrierten Evaluation einer Suchmaschine (Das harmonische Mittel von Precision und Recall ist der F-*Score*.)

  Wenn die verschiedenen Bedeutungen nahe beieinander auftreten, wird *a* mit *Retrievalscore* bezeichnet, *b* mit *kombinierter Score* und bei *c* wird die Maßzahl genannt (MAP-*Score* oder F-*Score*)
- *Rankingalgorithmus* bezeichnet nicht das vollständige System, das entwickelt wird, sondern konkrete Modelle wie das Vektorraummodell oder den BM25-Algorithmus.

1 Einleitung

PubMed ist eine Meta-Datenbank für wissenschaftliche Veröffentlichungen aus der Biologie, Medizin und verwandten Disziplinen.

Im Sommer 2015 bietet PubMed Zugriff auf über 25 Millionen Dokumente (U.S. National Library of Medicine, 2015). Die Zahl der Dokumente, die jährlich zu PubMed hinzukommen, betrug in den letzten fünf Jahren durchschnittlich etwa 900 000:

Jahr	**Zahl der Dokumente**	**Davon neu**
2009	17 764 826	884 811
2010	18 502 916	738 090
2011	19 569 568	1 066 652
2012	20 494 848	925 280
2013	21 508 439	1 013 591
2014	22 376 811	868 372

(U.S. National Library of Medicine, 2013)

Diese Menge an bestehenden und neuen Dokumenten macht ein gutes Informationssystem notwendig, wenn man sich mit PubMed in Forschungsfelder einarbeiten und den Überblick über neue Entwicklungen in bestimmten Bereichen behalten können soll.

Neben einer Suchmaschine, die direkt auf der Webseite von PubMed angeboten wird, gibt es diverse andere Indexierungsansätze und Suchmaschinen für die Datenbank, von denen auch einige in dieser Arbeit kurz vorgestellt werden.

Obwohl es bereits Forschung und Arbeiten zu Information-Retrieval-Fragestellungen für PubMed gibt, lohnt sich eine weitere Befassung mit der Datenbank aus IR-Sicht. Einerseits ist PubMed aufgrund der Größe der Kollektion und der Zahl der Anfragen – 2014 waren es etwa drei Millionen am Tag – eine wichtiges System für die Forschung in der Medizin (U.S. National Library of Medicine, 2014).

Andererseits sind über *PubMed Central* große Teile der Kollektion frei verfügbar und es gibt Hilfsmittel, die das Parsen, Indexieren und die Erstellung von Suchmaschinen

erleichtern. Mit den vielen Metainformationen kann man neue Ansätze für die Indexierung und das Ranking in PubMed und allgemein in IR-Systemen implementieren und evaluieren.

Das folgende Kapitel bietet einen groben Überblick über Forschung zu PubMed aus der *Information Retrieval*-Perspektive und zu anderen Informationssystemen, die auf PubMed aufbauen. Dabei wird gezeigt, dass die in verschiedenen Arbeiten untersuchten Ansätze und unterschiedliche Rankingstrategien selten in einem System kombiniert werden, obwohl das potenzielle eine Verbesserung der Retrievalleistung bringt.

Im dritten Kapitel wird ein neuer Faktor für das Ranking von Dokumenten in PubMed vorgestellt: der „kombinierte Score", ein themenabhängiges *Bibliometric*, das mit Unterstützung durch Ontologien berechnet wird. Eine zentrale Frage, die in dieser Arbeit beantwortet wird, ist, inwiefern man durch diesen kombinierten Score die Ergebnisse einer Suchmaschine verbessern kann.

Kapitel vier ist in drei große Punkte unterteilt: Zunächst wird die Kollektion vorgestellt, die in dieser Arbeit als Goldstandard dient, der *TREC Genomics Track* von 2006. Darauf folgt eine Beschreibung der verschiedenen Systemvarianten bzw. Rankingfaktoren, die später verglichen werden. Der dritte Unterpunkt ist die Beschreibung des *Learning to Rank*-Verfahrens, das für diese Arbeit entwickelt wurde.

Das fünfte Kapitel beinhaltet die Ergebnisse der Evaluation der Systemvarianten. Zuvor wird darin noch geklärt, nach welchen Maßstäben diese Evaluation erfolgt.

Den Abschluss bilden eine Diskussion der Ergebnisse sowie ein Ausblick auf zukünftige Forschung zu den untersuchten Rankingfaktoren in PubMed.

2 Forschung zu PubMed als Informationssystem

In diesem Kapitel wird zunächst gezeigt, dass IR-bezogene Forschung zu PubMed das System von verschiedenen Blickweisen her untersuchen kann. Besondere Aufmerksamkeit gilt dabei Arbeiten, die sich mit der systemzentrierten Verbesserung der Suchergebnisse befassen.

Hierbei werden zwei Punkte genauer betrachtet, die die Basis für den in Punkt 3 der Arbeit vorgestellten neuen Rankingfaktor bilden: Ontologien als Hilfsmittel beim Retrieval und *Bibliometrics* als Rankingfaktor im Web und in PubMed.

Abschließend wird argumentiert, dass es zu wenig Forschung zur Kombination von Rankingstrategien für Dokumente in PubMed gibt.

2.1 Verschiedene Forschungsdisziplinen

Im Rahmen dieser Arbeit wurden 28 Suchmaschinen zu PubMed sowie 10 Forschungsarbeiten systematisch verglichen. Der Vergleich ist in einer Tabelle dokumentiert, die sich im Anhang befindet (A2: Überblick von Informationssystemen und Forschungsarbeiten zu PubMed).

Die Tabelle enthält für jedes System und jeden Artikel Informationen darüber, was der jeweilige Fokus ist, welche Kollektion verwendet wird und – falls Suchmaschinen entwickelt wurden und diese Information verfügbar ist – ob und wie sie implementiert wurden und evaluiert wurden.

In diesem Kapitel wird nur eine kleine Auswahl der Systeme und Arbeiten vorgestellt, die zeigen soll, dass die Forschung zu PubMed vielseitig ist und verschiedene Aspekte des Informationssystems untersucht bzw. zu verbessern versucht. Die vollständige Tabelle befindet sich im Anhang.

Es gibt beispielsweise alternative Benutzeroberflächen und Bedienkonzepte für die Suchmaschine, die die Zugänglichkeit erhöhen sollen. Ein Beispiel hierfür ist die Seite *HubMed*, die mit einem minimalistischen *UI* arbeitet und damit „nonexpert searchers“ ansprechen soll (Eaton, 2006). Die Webseite *SLIM* versucht das mithilfe von Slidern, mit denen man Suchparameter einstellen kann (Muin, Fontelo, Liu, & Ackerman, 2005).

Darüber hinaus gibt es Systeme, die die Suchergebnisse aus PubMed automatisch analysieren und aufbereiten. *PubReMiner* zeigt für die Ergebnisse einer Query an PubMed an, wie häufig die Artikel aus bestimmten Journals stammen und wie viele von welchen Autoren verfasst wurden. Außerdem erstellt das System eine Übersicht, welche Wörter häufig in den Abstracts und Titeln vorkommen. Dadurch sollen Forscher weiterführendes Material zu einem Thema finden können und Querys präziser formulieren können (Koster, 2014).

Forschung zum Benutzerverhalten von Wissenschaftlern befasst sich häufig auch mit PubMed. Die Arbeiten widmen sich aber meist dem Informationssuchverhalten von Forschern allgemein und PubMed wird als viel genutzte Datenbank genannt, jedoch

nicht spezifisch untersucht. Zumindest solche Studien, die sich auf das *Informaton Behvaiour* von Ärzten und Forschern aus den Biowissenschaften konzentrieren, können als IR-Forschung zu PubMed angesehen werden. Ein Beispiel ist die Arbeit von Korjonen-Close (2005), in der versucht wird, durch eine Umfrage zu bestimmen, was ein neues Informationssystem für Wissenschaftler aus diesen Bereichen bieten soll. Es zeigt sich, dass Forscher zum Zeitpunkt der Umfrage (2005) unzufrieden mit den bestehenden Informationssystemen waren und nicht über das Know-How verfügt haben, die für sie notwendigen Informationen in den Systemen zu finden.

Studien von Searchlogs der PubMed-Suchmaschine können dazu dienen, das Nutzerverhalten in der Praxis zu erfassen. In einer Logstudie von Dogan, Murray, Névéol, & Lu (2009) wird beispielsweise versucht, die Anforderungen der Benutzer an PubMed basierend auf ihren Queries (Kategorien, durchschnittliche Länge) und Suchsessions (Anzahl der Queries, Dauer) zu bestimmen und dies als Ausgangspunkt für systemseitige Verbesserungen zu bentuzen.

Forschungsarbeiten und Systeme, die darauf abzielen, das Retrieval systemseitig zu verbessern, werden im folgenden Punkt vorgestellt.

2.2 Forschung zur Verbesserung des Retrievals

Seit Oktober 2013 gibt es die Möglichkeit, sich in PubMed die Suchergebnisse nach ihrer Relevanz sortiert anzeigen zu lassen. Vorher wurde nur Boolesches Retrieval angeboten und die Suchergebnisse wurden absteigend nach ihrer Aktualität zurückgegeben, was nach wie vor die Standardeinstellung für Querys ist. Die Beschreibung in einer offiziellen Mitteilung legt nahe, dass die Relevanz vom Erscheinungsdatum der Artikel sowie von der Häufigkeit des Vorkommens der Suchterme in verschiedenen Feldern der Dokumente abhängt (Canese, 2013).

Es gab und gibt von anderen Seiten Ansätze, die solche und ähnliche Rankingmethoden für PubMed verwenden, sei es für IR-Forschungszwecke oder die tatsächliche Verwendung in Informationssystemen für Wissenschaftler und Ärzte.

In den nächsten zwei Unterpunkten werden zwei Rankingstrategien vorgestellt, die mehr Informationen zum Ranking benutzen als die Artikel an sich:

1. die Suche nach Synonymen mithilfe von Ontologien

2. *Bibliometrics*, die Beziehungen zwischen den Dokumenten als Zusatzinformation benutzen

2.2.1 Ontologien als Hilfsmittel

Gruber definiert Ontologien als explizite Spezifikation von Begrifflichkeiten („explicit specification of a conceptualization“) und beschreibt sie folgendermaßen:

> „Pragmatically, a common ontology defines the vocabulary with which queries and assertions are exchanged among agents. Ontological commitments are agreements to use the shared vocabulary in a coherent and consistent manner. [...] In short, a commitment to a common ontology is a guarantee of consistency, but not completeness, with respect to queries and assertions using the vocabulary defined in the ontology.“ (Gruber, 1995)

Eine Ontologie beinhaltet also formalisiertes Wissen einer Domäne.

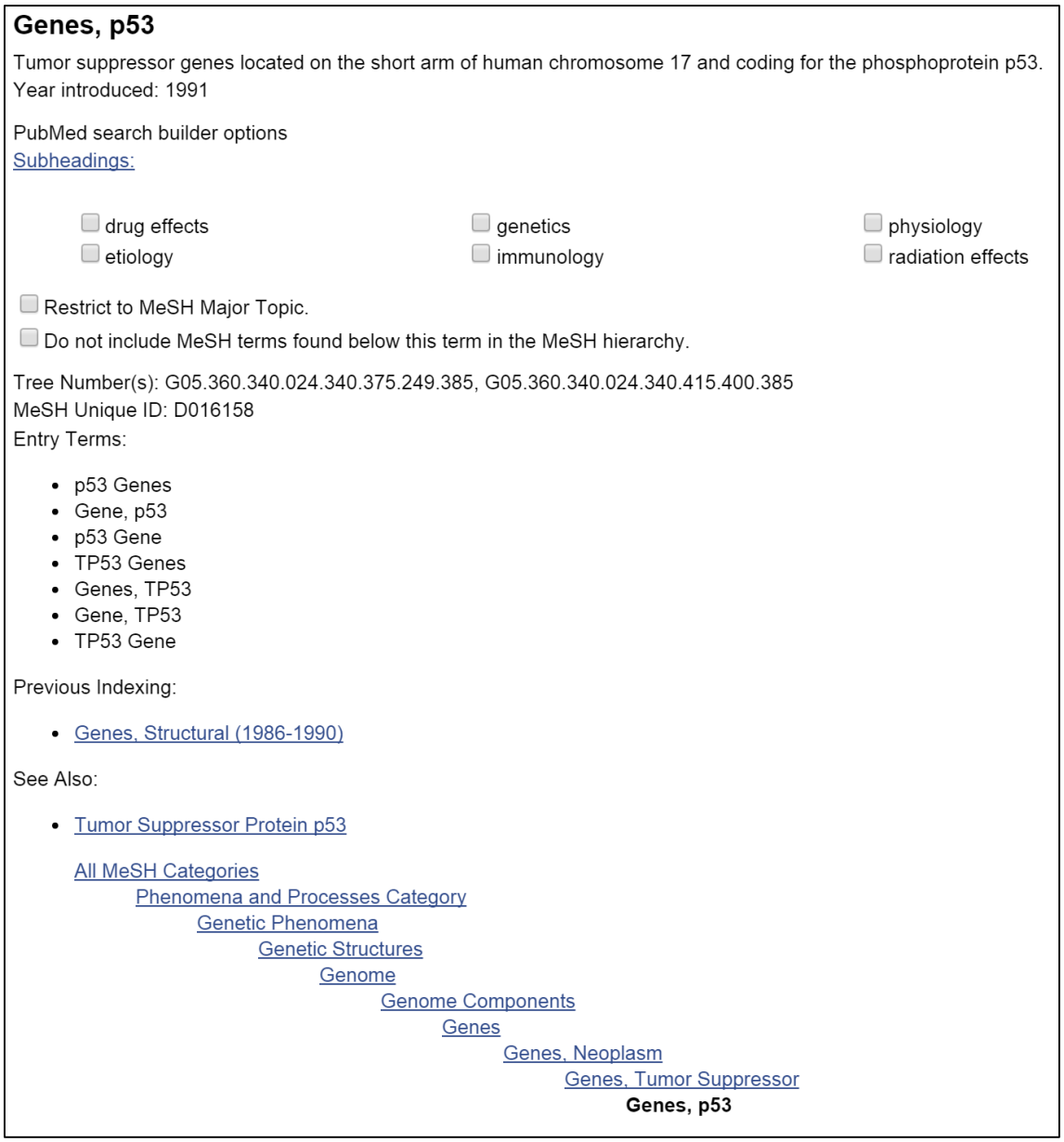
Es gibt zur Unterstützung der Indexierung und des Retrievals von Dokumenten für PubMed die Ontologie *MeSH* (**Me**dical **S**ubject **H**eadings), die genau wie PubMed von der *National Library of Medicine* verwaltet wird.

MeSH-Einträge zu einem Konzept beinhalten folgende Punkte:

- Eine Beschreibung des Konzepts
- Eine Liste von *Subheadings* (mögliche Einschränkungen bei der Suche nach Begriffen oder Artikeln, die im Zusammenhang mit dem Konzept vorkommen, z.B. Arten von Forschungsarbeiten oder Forschungsdisziplien)
- Eine Liste von Synonymen (*Entry Terms*) für das *Heading*
- Die *Tree Number*: MeSH ist hierarchisch, also baumförmig aufgebaut. Über die Nummer kann man die Position im *MeSH-Tree* bestimmen.

All diese Informationen sind in einer XML-Datei verfügbar, sodass man sie in Software integrieren kann. Die *NLM* bietet außerdem auf ihrer Webseite eine Suchmaschine speziell für MeSH-Headings an (U.S. National Library of Medicine, 2015).

Hier ist ein Beispiel für den Eintrag zu p53-Genen in MeSH auf der Webseite (der hier als Beispiel dient, weil er vergleichsweiche kurz ist, aber alle möglichen Elemente beinhaltet):



Genes, p53

Tumor suppressor genes located on the short arm of human chromosome 17 and coding for the phosphoprotein p53.
Year introduced: 1991

PubMed search builder options
Subheadings:

drug effects | genetics | physiology
etiology | immunology | radiation effects

Restrict to MeSH Major Topic.
Do not include MeSH terms found below this term in the MeSH hierarchy.

Tree Number(s): G05.360.340.024.340.375.249.385, G05.360.340.024.340.415.400.385
MeSH Unique ID: D016158
Entry Terms:

- p53 Genes
- Gene, p53
- p53 Gene
- TP53 Genes
- Genes, TP53
- Gene, TP53
- TP53 Gene

Previous Indexing:

- Genes, Structural (1986-1990)

See Also:

- Tumor Suppressor Protein p53

All MeSH Categories
Phenomena and Processes Category
Genetic Phenomena
Genetic Structures
Genome
Genome Components
Genes
Genes, Neoplasm
Genes, Tumor Suppressor
Genes, p53

Abbildung 1: MeSH-Eintrag für p53-Gene (U.S. National Library of Medicine, 1991)

(ein zweiter Eintrag für „See Also“ ist nicht im Bild enthalten)

Neben MeSH gibt es andere Ontologien und Datenbanken für Gene, die häufig in den Systemen verwendet werden. Beispielsweise benutzt die Suchmaschine *GoPubMed* die *Gene Ontology*, um die Dokumente in PubMed nach den darin untersuchten Proteinen zu kategorisieren (Doms & Schroeder, 2005).

EBIMed arbeitet ebenfalls mit solchen Datenbanken, um Informationen über Zusammenhänge zwischen Proteinen und Medikamenten oder anderen Proteinen zu finden. Bei einer Suche nach einem Term werden für alle Dokumente, die PubMed als Ergeb-

nisse anzeigt, alle Namen von Termen, die in den unterstützenden Datenbanken vorkommen, extrahiert und es wird angezeigt, welche in den Suchergebnissen häufig miteinander auftreten (Rebholz-Schuhmann, et al., 2007).

Ein von Demner-Fushman & Lin entwickeltes System generiert aus Abstracts von Dokumenten automatisch eine Antwort auf eine Frage. Hier kommt wieder eine Ontologie zum Einsatz, das *Unified Medical Language System* (Demner-Fushman & Lin, 2006).

Man kann mithilfe von MeSH Suchanfragen an PubMed (semi-)automatisch um Synonyme erweitern, was etwa bei der „offiziellen" PubMed-Suchmaschine geschieht. Bei Zhiyong, Kim, & Wilbur (2009) findet man konkrete Werte für die Verbesserung, die durch diese automatische Suche nach Synonymen auftreten sollte. Mit der Kollektion des *TREC Genomics Track* von 2006 (s. Punkt 4.1) bringt die Verwendung von MeSH eine Verbesserung des harmonischen Mittels von Precision und Recall von 0.334 auf 0.406.

In dieser Arbeit werden zwei Ontologien verwendet; MeSH und die Gensuchmaschine *GeneCards*® (www.genecards.org). Bei *GeneCards* handelt es sich um eine Datenbank, die laut eigener Angabe Informationen zu allen bekannten und vermuteten menschlichen Genen beinhaltet (Weizmann Institute of Science, 2015). Zumindest bei den für diese Arbeit nachgeschlagenen Genen schien *GeneCards* umfangreicher zu sein als MeSH. Ursprünglich sollte nur MeSH benutzt werden, aber dort sind nicht zu allen Genen, die in den Querys des *Genomics Track* vorkommen, Einträge vorhanden (Stand: September 2015).

2.2.2 Bibliometrics zur Bestimmung relevanter Artikel

Citation Analysis verfolgt den Grundgedanken, dass man den Einfluss einer wissenschaftlichen Arbeit bestimmen kann, in dem man zählt, wie häufig anderen Arbeiten darauf verweisen. Man kann damit *Bibliometrics* berechnen, also Zahlenwerte, mit denen sich der Einfluss von Arbeiten messen und vergleichen lässt. In dem Zusammenhang sollte der *Science Citation Index* erwähnt werden, den es seit Anfang der 1960er Jahre gibt. Dabei handelt es sich um eine umfassende Zitationsdatenbank, die die Entwicklung diverser *Bibliometrics* ermöglicht hat (Garfield, 1995).

Wenn sich viele Dokumente auf einen bestimmten Artikel berufen, dann kann man das als Qualitätsmerkmal ansehen. Im Information Retrieval bietet es sich an, eine Maßzahl, die auf dieser Grundlage basiert, in das Ranking von Dokumenten einfließen zu

lassen, da ein hoher Einfluss für eine hohe Relevanz spricht. Das gilt für alle Kollektionen, in denen es Zitate bzw. Verweise auf andere Dokumente gibt, also nicht nur für wissenschaftliche Veröffentlichungen, sondern z.B. auch für das Web.

Ein prominentes Beispiel ist der *PageRank*, der u.a. von den Google-Gründern Larry Page und Sergej Brin für die Websuche entwickelt wurde. Dabei wird ein Benutzer modelliert, der zufällig durch das Web surft („idealized random Web Surfer", Page, Brin, Motwani, & Winograd, 1998). Zufällig bedeutet hier, dass die Wahrscheinlichkeit des Besuchs einer Seite davon abhängt, wie viele Links dorthin führen, nicht vom Inhalt der Seite. Dadurch ist es wahrscheinlicher, dass ein Benutzer Seiten besucht, die häufig referenziert werden. Links von diesen „wichtigen" Seiten erhöhen den PageRank einer Seite mehr als Links von anderen Seiten (Page, Brin, Motwani, & Winograd, 1998).

Für PubMed kann man den PageRank direkt übertragen, was von Bernstam, et al. (2006) untersucht wurde. Sie vergleichen verschiedene Rankingalgorithmen für PubMed und den Einfluss von verschiedenen Maßzahlen, die auf *Citation Analysis* basieren, u.a. auch den PageRank. Durch diese Einflussfaktoren sollen mehr Dokumente, die von einer Fachgesellschaft von Chirurgen als „must read" eingestuft wurden, die oberen Ränge der Suchergebnisse einnehmen. Bernstam, et al. konnten zeigen, dass eine signifikante Verbesserung durch die Verwendung von Bibliometrics eintritt.

2.3 Mangel an Forschung zur Kombination von Rankingfaktoren

In den meisten der Systeme und Forschungsarbeiten, die in der Übersichtstabelle der Forschung zu PubMed aufgelistet sind, wurde ein Ansatz (teils in mehreren Variationen) implementiert und – falls ein systemorientierter Test möglich war und erfolgt ist – mit PubMed als Baseline verglichen. Ob der Ansatz eine Verbesserung in existierenden Systemen darstellt und sich gut mit anderen Rankingfaktoren kombinieren lässt wird nicht untersucht.

Dem gegenüber steht der Suchalgorithmus der Google-Suchmaschine (*Hummingbird)*, der einer Schätzung nach über 200 Einflussfaktoren benutzt, um die relevantesten Seiten für eine Query zu bestimmen (Sullivan, 2013). Die Kombination von verschiedenen Ansätzen spielt hier eine große Rolle.

Die Anforderungen an eine Websuchmaschine sind vielseitig und nicht auf die zum Teil stark spezialisierten Systeme für Wissenschaftler übertragbar, weswegen die geringere Zahl der Einflussfaktoren kein Anzeichen für eine geringe Qualität sein muss. Aber Informationssysteme für PubMed wie z.B. die „offizielle" Suchmaschine führen auf der technischen Ebene genauso Textsuche in Dokumenten durch. Es bestehen zwar hinsichtlich der Zielgruppe und des Verwendungszwecks große Unterschiede zu Websuchmaschinen, von der Systemperspektive her sind sie jedoch verwandt: Die Suchmaschine soll in beiden Fällen relevante Artikel für eine Suchanfrage zurückgeben.

Deswegen lohnt es sich, zu untersuchen, ob auch bei der Suche in *PubMed* die Kombination von Rankingfaktoren eine Verbesserung der Suchergebnisse ermöglicht.

Es gibt Arbeiten, die sich mit der Kombination von Faktoren befassen, wie die von Lu, Kim, & Wilbur (2009). Darin werden drei verschiedene Ansätze für das Ranking von Dokumenten in PubMed verglichen und es wird jeweils überprüft, inwiefern eine Verbesserung durch die Suche nach Synonymen aus MeSH eintritt. Es zeigt sich, dass MeSH für eine der Rankingstrategien eine Verbesserung bringt, für die Beste der untersuchten Methoden (die auf dem Vektorraummodell basiert) jedoch nicht.

In dieser Arbeit wird ein Rankingfaktor vorgestellt, der Beziehungen zwischen Dokumenten ausnutzt, um zu bestimmen, welche Seiten für eine Query relevant sind. Dabei spielt der Inhalt der Dokumente auch eine Rolle. Der Ansatz wird in Punkt 3 dieser Arbeit genauer beschrieben.

Der neue Rankingfaktor stellt einen der Hauptpunkte dieser Arbeit dar; der andere ist der Vergleich mit drei anderen Faktoren und die Kombination der verschiedenen Ansätze. Zwei der Ansätze sind der PageRank und die Verwendung von Ontologien. Der kombinierte Score sowie der PageRank-Ansatz und die Verwendung von Ontologien sind verwandte Ansätze, da sie alle über den Vergleich von Querytermen mit den Termen der Dokumente und der Kollektion hinausgesehen.

Deswegen wird als weitere Systemvariante und Vergleichspunkt untersucht, welche Verbesserung man ohne solche Faktoren erreichen kann, also nur durch die Wahl von Indexierungsvariante und Rankingalgorithmus der Suchmaschine.

3 Der kombinierte Score

In diesem Kapitel wird der „kombinierte“ (oder „themenabhängige“) Score eingeführt, der die Rankingfaktoren „*Bibliometrics*“ und „Verwendung von Ontologien“ kombiniert (s. 3.1).

Die Kombination lässt sich durch unterschiedliche Implementierungen realisieren, von denen drei Varianten vorgestellt werden. Zuletzt werden potenzielle Vorteile des Scores aufgelistet.

3.1 Erklärung des Ansatzes

Der PageRank (und andere dokumentbezogene *Bibliometrics*) ordnen Dokumenten einen Wert zu, der von der Suchanfrage unabhängig ist. Der kombinierte Score ist von der Anzahl der Zitate abhängig, die ein Dokument von Artikeln erhält, die für ein Thema relevant sind. Damit ist er indirekt vom Thema der Query abhängig und soll genauer ausdrücken, ob ein Dokument relevant für eine Suchanfrage ist als *Bibliometrics*, die für jedes Dokument immer denselben Wert haben.

Formeller lässt sich das folgendermaßen beschreiben:

Für ein Thema gibt es ein Set an Dokumenten aus PubMed, die sich damit befassen (dieses Set wird fortan mit *d1* bezeichnet). Jedes Dokument, das von einem Dokument aus *d1* zitiert wird, gehört zum Set *d2*. Dokumente in *d2* können auch gleichzeitig in *d1* sein.

Allen Dokumenten aus *d2* wird ein Wert zugewiesen, der angibt, wie relevant sie für das Thema sind, was von den Zitaten aus *d1* abhängt. Für die genaue Berechnung der Werte kann man verschiedene Ansätze verfolgen; Punkt 3.2 beinhaltet drei mögliche Formeln bzw. Verfahren zur Bestimmung des kombinierten Scores der Dokumente. Dieser Wert nimmt dann Einfluss auf den *Retrievalscore* und damit das Ranking jedes Dokuments für eine Suchanfrage zu diesem Thema. Folgende Grafik soll dies veranschaulichen:

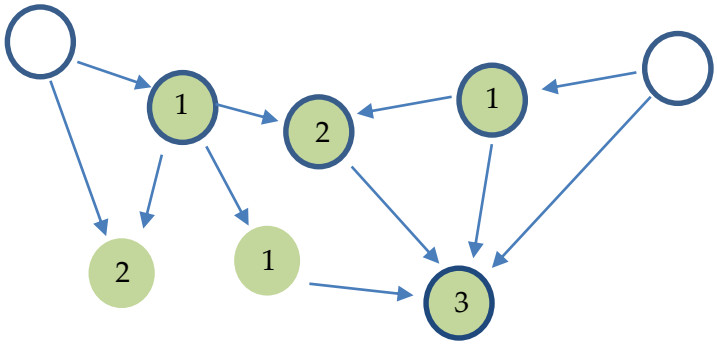


Abbildung 2: „Zitationsnetz" für ein Thema

Jeder Kreis steht für ein Dokument aus den Ergebnissen einer Suche nach einem medizinischen Konzept, z.B. der Entstehung einer bestimmten Krankheit. Die fünf Kreise mit Rand stellen Dokumente dar, die für dieses Thema relevant sind, das ist die Menge *d1*.

Ein Dokument, von dem ein Pfeil ausgeht, zitiert das Dokument, auf das der Pfeil zeigt. Die grün ausgefüllten Kreise bilden damit die Menge *d2*. Die Zahl in den Kreisen gibt an, wie häufig sie von Dokumenten aus *d1* zitiert werden. *d2* enthält nicht nur relevante Dokumente, sondern auch zwei nichtrelevante Dokumente, und nicht alle relevanten Dokumente sind in *d2* enthalten.

Beim PageRank wurde ursprünglich vorgeschlagen, alle Webseiten als Treffer anzugeben, die die Suchbegriffe im Titel enthalten, und sie nach dem PageRank-Wert absteigend zu sortieren (Page, Brin, Motwani, & Winograd, 1998). Der kombinierte Score ist als Ergänzung zu anderen Rankingfaktoren geplant, weswegen hier solche Fehler akzeptiert werden. Es sollte so sein, dass die Mengen *d1* und *d2* eine große Überschneidungsmenge haben, somit sind die meisten Dokumente, die den Score erhalten, relevant für das Thema.

Um die Methode umsetzen zu können ist es wichtig, zu wissen, welche Dokumente relevant für ein bestimmtes Thema sind, also welche Dokumente Teil von *d1* sind. Für diese Arbeit werden darunter alle Dokumente verstanden, die Terme beinhalten, mit denen man nach dem Thema suchen würde. Man muss folglich bereits eine Suche durchführen, um *d1* zu bestimmen.

MeSH und andere Ontologien helfen, die dafür notwendigen Querys zu konstruieren und die Terme zu finden, die bestimmte Konzepte beschreiben. Deswegen wird der Score hier als „kombinierter Score“ bezeichnet: Neben Beziehungen zwischen den Dokumenten (Zitaten) werden auch noch Metadaten der Dokumente benutzt (ihr vermutlicher Inhalt).

d1 enthält mit dieser Vorgehensweise vermutlich in vielen Fällen *False Positives*, also Dokumente, die nicht relevant für das Thema sind, aber bei der initialen Suche zurückgegeben werden, was der Definition von *d1* widerspricht. Einige der in Punkt 3.3 aufgeführten Vorteile wären jedoch nicht mehr vorhanden, wenn man nicht möglichst alle für ein Thema relevanten Dokumente finden würde.

Außerdem erhalten Dokumente einen höheren Score, wenn sie häufiger von den Dokumenten in *d1* zitiert werden, was im Optimalfall die *False Positives* zu einem gewissen Grad ausgleicht; wenn die Menge *d1* präzise genug ist, wirken sich deren Zitate nicht so stark aus.

Hierbei handelt es sich trotzdem um den offensichtlichsten Schwachpunkt des Ansatzes: Man muss davon ausgehen, dass man vor der Berechnung des Scores bereits relevante Dokumente findet, benötigt also schon gute Ergebnisse, um diese noch verbessern zu können.

3.2 Varianten der Berechnung des Scores

Zur Bestimmung des konkreten Werts des Scores eines Dokuments für ein Thema kann man verschiedene Formeln oder Verfahren anwenden. In dieser Arbeit wurden drei Varianten entworfen, implementiert und getestet.

3.2.1 Zahl der Zitate

$$\text{Kombinierter Score}_{\text{Thema}}(d \in d2) = \frac{Zahl\ der\ Dokumente\ \text{in}\ d1,\ \ die\ d\ zitieren}{Zahl\ der\ Zitate\ in\ d1}$$

Für jedes Dokument wird hier gezählt, wie häufig es von Dokumenten aus *d1* zitiert wird. Diese Anzahl wird durch die Summe der Referenzen der einzelnen Dokumente aus *d1* geteilt. Wenn bei einem Thema zwei von zehn relevanten Dokumenten aus *d1* einen Artikel referenzieren, dann bekommt er einen kombinierten Score von 0,2 für das Thema zugewiesen.

Diese Variante des Scores ist im Vergleich zum PageRank relativ simpel gehalten, bei dem Zitate aus anderen Dokumenten, die bereits selber hohe PageRank-Werte haben, stärker gewichtet werden.

3.2.2 *Retrievalscores* der zitierenden Dokumente

Kombinierter $\text{Score}_{\text{Thema}}(d \in d2) = \frac{Summe\ der\ Retrievalscores\ der\ Dokumente\ aus\ d1,\ \ die\ d\ zitieren}{\text{Gesamtretrievalscore der Dokumente in } d1}$

Dieser Variante liegt der Gedanke zugrunde, dass ein Dokument, das einen höheren *Retrievalscore* für eine Query hat, vom System mit einer höheren Wahrscheinlichkeit als relevant für diese Query bzw. das Thema der Query eingestuft wird.

Wenn also ein Dokument von vielen Dokumenten zitiert wird, die gute Treffer für das Thema darstellen, so sollte es bei dieser Variante als wichtiger eingestuft werden.

Hierbei gibt es den Nachteil, dass man vor der Berechnung des kombinierten Scores bereits eine Formel benötigt, um den „gewöhnlichen" *Retrievalsore* eines Dokuments zu bestimmen. Eine hohe Zahl von *False Positives* ist hier im Gegensatz zu den anderen beiden Varianten kein Problem; die Qualität ist trotzdem davon abhängig, wie gut man *d1* in der Praxis bestimmen kann.

3.2.3 An *PageRank* angelehnte Berechnung

Die dritte Variante des Scores ist an den PageRank angelehnt, wobei eine andere Berechnung stattfindet, da es nicht als sinnvoll angesehen wurde, hier von einem *Random Surfer* auszugehen. Es gibt für diese Variante keine Formel sondern stattdessen ein aus mehreren Schritten bestehendes Verfahren:

1. Der Score wird zunächst wie bei Verfahren 1) berechnet.
2. *d1* wird um alle Dokumente aus *d2* erweitert, die noch nicht in *d1* waren.
3. *d2* wird zu allen Dokumenten, die von Dokumenten aus dem „neuen" *d1* zitiert werden.
4. Der Score errechnet sich nun mit folgender Formel:
   Kombinierter $\text{Score}_{\text{Thema}}(d \in d2)$ = (1 – a) * (alter kombinierter Score von d) + a * (Summe der kombinierten Scores der Dokumente aus d1, die d zitieren)
5. Jeder Score wird zur Normalisierung mit $\frac{1}{Summe\ aller\ kombinierten\ Scores}$ multipliziert.

Das Verfahren ist iterativ, die Schritte 2 bis 5 werden wiederholt. *a* gibt die „Lerngeschwindigkeit" des Verfahrens an und liegt zwischen 0 und 1. Werte nahe bei 1 machen den Einfluss der Scores der zitierenden Dokumente bereits bei wenigen Iterationen

größer, während für Werte von *a* nahe 0 der ursprüngliche Score (aus Schritt 1) wichtiger ist. Die Normalisierung (Schritte 5 und 6) erfolgt, damit nicht bei vielen Iterationen Scores entstehen, die in völlig verschiedenen Größenordnungen liegen.

3.3 Erhoffte Vorteile

In der Einleitung wurde bereits die Größe der Kollektion als Motivation für die Erstellung guter Suchmaschinen vorgestellt. Das könnte gerade bei der Einarbeitung in neue Themengebiete Schwierigkeiten verursachen.

Hierbei wäre es von Interesse die Dokumente zu finden, die als erstes ein bestimmtes Thema oder einen Zusammenhang untersucht haben. Spätere wissenschaftliche Arbeiten verweisen häufig auf diese Dokumente und man könnte diese Information nutzen, um das Ranking der ersten Dokumente zu erhöhen und den Einstieg bei der Einarbeitung in ein neues Thema zu erleichtern.

Es gibt auch Grund zur Annahme, dass die Rankingmethode bei vielen klassischen IR-Problemen eine Unterstützung sein kann:

Man kann bei der Entwicklung einer Suchmaschine festlegen, ob verschiedene Schreibvarianten, Synonyme und verschiedene grammatikalische Formen von Wörtern bei der Suche als gleichwertig betrachtet werden und ob Dokumente, die diese Formen (aber nicht die exakten Suchterme) enthalten, Treffer darstellen und einen hohen Score erhalten. Das sind Entscheidungen, die sich auf Recall und Precision und damit die Qualität der Suchergebnisse auswirken. Bei PubMed kommt hinzu, dass für Namen von Proteinen und Genen häufig Kurzschreibweisen verwendet werden. Hier ist u.a. bei der Tokenisierung besondere Vorsicht geboten, was in Punkt 4.2.1.1 genauer erklärt wird.

Es ist in jedem Fall notwendig, für diese Punkte eine Lösung zu finden, die den Anforderungen der Suchenden gerecht wird.

Der kombinierte Ansatz könnte helfen, viele dieser Probleme zu umgehen. Dabei zählt vor allem die Häufigkeit, mit der die Artikel von vermeintlich relevanten Dokumenten zitiert werden. Dies kann ein besseres Anzeichen von Relevanz sein als eine höhere Termfrequenz, mit der bestimmte Begriffe in den Dokumenten vorkommen.

Nicht zuletzt könnte der Ansatz helfen, den Recall zu erhöhen. Beim PageRank wäre es nicht möglich, immer Seiten anzuzeigen, die einen hohen Wert haben, da der PageRank unabhängig von der Suchanfrage ist; beim themenabhängigen Score wäre es möglich, alle Dokumente zur Liste der Ergebnisse hinzuzufügen, die Teil von *d2* sind.

Allerdings gilt hier die Einschränkung, dass *d1* und *d2* relativ ähnlich sein sollten (s. Punkt 3.1). *d1* wird – wie die „normale" Ergebnismenge – durch Suchanfragen bestimmt, weswegen die meisten Dokumente aus *d1* sich bereits in den Suchergebnissen befinden. Eine große Erhöhung des Recalls würde also bedeuten, dass man durch Suchanfragen *d1* nicht genau genug bestimmen kann.

4 Durchführung

Dieses Kapitel beschreibt das Vorgehen, mit dem die mögliche Verbesserung durch den kombinierten Score und durch die anderen getesteten Faktoren in dieser Arbeit bestimmt wurden. Bei der Methodik dient andere systemzentrierte Forschung als Vorbild, vor allem der allgemeine Ablauf von Arbeiten im Rahmen der TREC-Konferenzen. Der Sinn der Orientierung an anderen Arbeiten und der Verwendung derselben Testkollektion ist, dass man so Vergleichbarkeit mit anderen Arbeiten und Systemen schafft und die Ergebnisse verallgemeinerbar sind (Harman, 1993).

Zunächst wird der gewählte Goldstandard vorgestellt: der *TREC Genomics Track* von 2006. Dann wird auf die Implementierung der Suchmaschine mit *Solr* und die verschiedenen Systemvarianten eingegangen.

In dieser Arbeit wurde besonderer Wert auf die automatische Optimierung der verschiedenen Systeme gelegt, sodass es einen eigenen Punkt für die verwendete *Learning to Rank*-Methode gibt.

4.1 Der TREC Genomics Track von 2006

Um den mit den beschriebenen Verfahren erzielten Gewinn feststellen zu können, muss man die Qualität einer Suchmaschine messen können, die damit arbeitet, was beispielsweise mit einer Testkollektion mit Relevanzbeurteilungen für bestimmte Themen möglich ist. Für PubMed gab es bis 2007 den *TREC Genomics Track*, d.h. eine Teilmenge von Dokumenten aus PubMed, für die Relevanzbeurteilungen für eine Menge an Themen

vorgenommen wurden. Dieser Punkt erklärt die Inhalte und den Aufbau des *Genomics Tracks* von 2006, der bei der Evaluation als Goldstandard dient.

4.1.1 Allgemeines zum *TREC Genomics Track*

TREC steht für ***T**ext **RE**trieval **C**onference* und ist eine wiederkehrende Konferenz in deren Rahmen sogenannte *Tracks* erstellt und untersucht werden. Hierfür werden Testkollektionen für verschiedene Bereiche zur Verfügung gestellt und auf diverse Fragestellungen hin untersucht, um Aufgaben aus unterschiedlichen IR-Problemstellungen zu bearbeiten.

Bis 2007 gab es dabei einen Track, der mit Dokumenten aus PubMed arbeitete: den Genomics Track. Ziel dieses Tracks war es, Retrievalaufgaben in einer bestimmten Fachdomäne zu untersuchen. Die Ergebnisse der Forschung hierzu sollten also auch auf andere fachspezifische Datenbanken übertragbar sein, wenn diese ähnlich aufgebaut sind. 2006 wurde ein Fokus auf die Themen *Text Mining* und *Information Extraction* gelegt (Hersh W. , Cohen, Roberts, & Rekapalli, 2006). Das wirkt sich so aus, dass die Relevanzbeurteilungen einzelne Textpassagen betreffen, nicht ganze Dokumente. Für diese Arbeit werden alle Dokumente als relevant betrachtet, die diese Passagen beinhalten, da hier nur die Abstracts betrachtet werden und nicht die Volltexte.

Die für den Genomics Track erstellte Kollektion beinhaltet viele Forschungsarbeiten zu Genen, Proteinen, deren Beziehungen untereinander und zu Krankheiten und anderen biomedizinischen Prozessen. Sie besteht aus Dokumenten, die von 1995 bis 2006 veröffentlicht wurden (National Institue of Standards and Technology, 2014).

Der Genomics Track von 2007 war noch mehr auf Fragestellungen zu *Question-Answering* hin ausgelegt und weniger auf klassisches Retrieval von Dokumenten, weswegen hier nur der Track von 2006 verwendet wird.

4.1.1.1 *Anzahl und Aufbau der Dokumente und Topics*

In der Kollektion befinden sich 162 259 Dokumente aus 49 Journals, für die jeweils der Volltext in HTML und Metadaten in einem XML-Format verfügbar sind.

Einige der Dokumente konnten nicht indexiert werden, sodass insgesamt mit 160 472 Dokumenten gearbeitet wird. Die XML-Dateien beinhalten unter anderem die folgenden Informationen:

- PMID (*PubMed* ID) als eindeutige ID der Dokumente
- Autor(en)
- Datum der Veröffentlichung / Wiederveröffentlichung
- Journal
- Abstract
- Liste mit *MeSH*-Headings, die dem Dokument zugeordnet wurden

Die HTML-Dateien haben jeweils die PMID als Dateinamen, sodass eine Zuordnung möglich ist.

Die 28 Topics des Tracks sind Fragestellungen zu den Zusammenhängen zwischen einem oder mehreren Genen und einem anderen biologischen Konzept, wie etwa einer Krankheit oder der Funktionsweise eines Organs. Die Fragen sind in ausformulierter Form verfügbar, wobei die Gene und anderen Konzepte jeweils klar hervorgehoben werden. Hier sind zwei Beispiele:

ID	Gen	anderes Konzept	Frage
160	PRNP	Mad Cow Disease	What is the role of PrnP in mad cow disease?
180	RET and GDNF	kidney development	How do Ret-GDNF interactions affect liver development?

Abbildung 3: Beispiele für TREC-Topics

Für vier der Topics gibt es keine relevanten Dokumente, sie wurden bei der späteren Evaluation ausgeschlossen. Eine Übersicht, die u.a. diese Information beinhaltet, findet sich im Anhang „TREC-Topics“.

4.1.1.2 Extraktion der Zitatinformationen

Für diese Arbeit ist es essentiell zu wissen, welche Dokumente andere Dokumente referenzieren. Diese Information ist nicht direkt in den XML-Dateien enthalten, sondern lediglich über eine ID in den anchor-Tags der HTML-Dateien der Kollektion abrufbar. Die Links sind nur für die Dokumente vorhanden, die sich ebenfalls in der TREC-Kollektion befinden.

Außerdem handelt es sich bei der ID nicht um die PubMed-ID; es gibt eine Datei, die angibt, welche dieser IDs welcher PubMed-ID entspricht. Da auf der offiziellen Seite des TREC-Tracks angemerkt ist, dass einzelne Dokumente fehlen oder die IDs für diese falsch zugeordnet sind, muss man berücksichtigen, dass auch hier Fehler sein könnten

(Roberts, Cohen, & Hersh, 2015). Es ist darüber hinaus nicht sicher, dass jedem Zitat innerhalb der Kollektion ein Link mit der korrekten ID zugeordnet ist.

Es wurden 2 455 316 Zitate innerhalb der Kollektion gefunden.

4.1.2 Erweiterbarkeit des Ansatzes für PubMed und PubMed Central

Da für den TREC Genomics Track die Dokumente in einer anderen Datenstruktur verfügbar sind als für PubMed, ist es wichtig, zu bestimmen, ob man den Ansatz theoretisch für die ganze Kollektion implementieren könnte.

Mit *PubMed Central* sind 3,5 Millionen Dokumente aus PubMed frei verfügbar (U.S. National Library of Medicine, 2015). Zumindest für diese Dokumente könnte man die Scores berechnen, da hier in den XML-Dateien die notwendigen Informationen vorhanden sind.

Auf technischer Ebene benötigt man eine Kollektion von Dokumenten, zwischen denen es Beziehungen (z.B. in Form von Zitaten) gibt und im besten Fall eine Ontologie für Synonyme. Eigentlich muss die Kollektion auch nicht fachspezifisch sein, jedoch kann man nicht davon ausgehen, dass die Ergebnisse dieser Arbeit für diesen Fall aussagekräftig sind.

Man kann in PubMed die Felder der Dokumente angeben, in denen man nach Termen suchen will, und eins dieser Felder ist *References*. Es ist nicht klar, ob hier nur der Text der Referenzangabe indexiert ist oder ob z.B. auch die eventuell vorhandene *PubMed*-ID der Artikel gespeichert wird. Zumindest für die Artikel in PubMed Central ist diese Information vorhanden und der Ansatz wäre für diese Dokumente umsetzbar.

4.2 Beschreibung und Implementierung der Rankingfaktoren

Eine genaue Beschreibung der verschiedenen Einflussfaktoren auf die Berechnung der Scores der Dokumente ist an dieser Stelle essentiell, um das Vorgehen beim Machine Learning erklären zu können. Der zentrale Punkt dieser Arbeit ist zwar der kombinierte Ansatz, aber die anderen Systemvarianten und das Machine Learning sind die Mittel, mit denen man zeigen kann, welche Vorteile der Ansatz im Vergleich zu anderen Schritten bringt.

Die Implementierung der Suchmaschine erfolgte mit Java™ und *Solr 5.1* (http://lucene.apache.org/solr/). Solr ist ein Open-Source-Suchserver, der von der *Apache*

Software Foundation entwickelt wird. Sowohl Indexierung als auch Retrieval wurden mit dem System durchgeführt, eine Benutzeroberfläche wurde nicht entwickelt.

Die genaue Umsetzung der Querys wirkt sich auf das Ranking und damit die Ergebnisse der Evaluation aus, weswegen an dieser Stelle auf das Dokument „Erklärungen zu Solr und Programmen“ im Anhang verwiesen wird. Es beschreibt wichtige Punkte der Implementierung und erklärt, wie man Querys an das System sendet.

Die folgenden Unterpunkte beinhalten Informationen zur Umsetzung und zu Problemen der betrachteten Rankingfaktoren. Diese Rankingfaktoren sind:

1. Die Wahl von Indexierungsvariante und Rankingalgorithmus
2. Der PageRank
3. Die automatische Suche nach Synonymen mithilfe von Ontologien
4. Der kombinierte Score

4.2.1 Indexierungsvarianten und unterschiedliche Rankingalgorithmen

Bei Solr kann man Feldtypen anlegen, die aus einem *Analyzer* und einer *Similarity* bestehen. Ein Analyzer legt fest, wie die Inhalte der Felder (Text, Zahlen, Daten, Koordinaten, etc.) sprachlich verarbeitet werden. Die Similarity entspricht einem Rankingalgorithmus mit einer bestimmten Konfiguration.

Die zwei Felder *Abstract* und *Title* der Dokumente wurden jeweils mit 4 Analyzern und 10 Similaritys indexiert; so erhält man 40 Feldtypen und 80 Felder.

```
<fieldtype name="pubmed_language_remove_nonnumeric_rank_tfidf" class="solr.TextField">
    <analyzer>
        <charFilter class="solr.PatternReplaceCharFilterFactory" pattern="([^a-zA-Z0-9 ])" replacement=""/>
        <tokenizer class="solr.StandardTokenizerFactory"/>
        <filter class="solr.LowerCaseFilterFactory"/>
        <filter class="solr.StopFilterFactory" ignoreCase="true" words="stopwords_medline.txt"/>
    </analyzer>
    <similarity class="solr.DefaultSimilarityFactory">
    </similarity>
</fieldtype>

<fieldtype name="pubmed_language_remove_nonnumeric_rank_bm25_1" class="solr.TextField">
    <analyzer>
        <charFilter class="solr.PatternReplaceCharFilterFactory" pattern="([^a-zA-Z0-9 ])" replacement=""/>
        <tokenizer class="solr.StandardTokenizerFactory"/>
        <filter class="solr.LowerCaseFilterFactory"/>
        <filter class="solr.StopFilterFactory" ignoreCase="true" words="stopwords_medline.txt"/>
    </analyzer>
    <similarity class="solr.BM25SimilarityFactory">
        <float name="k1">1.2</float>
        <float name="b">0.75</float>
    </similarity>
</fieldtype>
```

Abbildung 4: Beispiele für die Definition von Feldtypen in Solr

4.2.1.1 Beschreibung der Indexierungsvarianten

Solr bietet viele Möglichkeiten, Text in Dokumenten zu Tokenisieren und weitere Mittel der Textverarbeitung (*Filter*) darauf anzuwenden.

Gerade für PubMed ist die Auswahl des Tokenizers und der Filter hochrelevant, da Gene und Krankheiten Namen oder Abkürzungen haben können wie „Nurr-77“, „Sec61“, „COUP-TF I“, „ApoE“ oder „Bop-Pes“, wobei es sich um Beispiele aus den Querys im TREC Genomics Track sind. Je nach Implementierung der Tokenizer kann ein Bindestrich zu einer Trennung in zwei Tokens führen. Zahlen können als Teil der Tokens, als eigene Tokens oder gar nicht indexiert werden. Ein Großbuchstabe in einem Wort kann dazu führen, dass das Wort gespalten wird.

Es wurden vier Varianten für die Sprachanalyse angelegt. Bei allen Varianten wird Groß- und Kleinschreibung ignoriert. Bei keiner Variante werden die Stoppwörter, die Solr standardmäßig für die englische Sprache vorsieht, sowie die Begriffe „disease“, „gene“ und „protein“ indexiert. Die Suchmaschine zeigte in ersten Testläufen für Querys, die diese Wörter enthielten, praktisch alle Dokumente als Treffer an. Die Stoppwortentfernung verringerte außerdem die Laufzeit der späteren automatischen Optimierung der Konfiguration der Systemvarianten.

Die vier verwendeten Analyzer sind:

1) „Simple“: Tokens sind durch Leerzeichen getrennt und werden nicht weiter verändert.
2) „Stemming“: Die Indexierungsvariante arbeitet wie die folgenden beiden mit Solrs „Standardtokenizer“. Hierbei kann nicht nur das Leerzeichen Tokens trennen, sondern u.a. auch Satzzeichen. Dafür gibt es Ausnahmen, so dass etwa die Buchstaben in Akronymen wie „I.B.M.“ nicht als einzelne Tokens verstanden werden. Als Stemmingalgorithmus wird Porter-Stemming verwendet.
3) „Remove_nonnumeric“: Bei diesem Analyzer werden alle Zeichen, die keine Buchstaben, Zahlen oder Leerzeichen sind, ersatzlos entfernt.
4) „Worddelimiter“: Dieser Analyzer trennt einzelne Tokens nach bestimmten Regeln weiter auf und entfernt z.B. im Englischen das possessive *s*. Die Tokens werden sonst nicht bearbeitet.

(Apache Software Foundation, 2015)

4.2.1.2 Formeln der Rankingalgorithmen

Das Ranking ist zwar kein Teil der Indexierung, wird bei Solr aber an derselben Stelle festgelegt, weswegen es hier auch an der gleichen Stelle erklärt wird.

Es werden vier Rankingalgorithmen verwendet. Aus Platzgründen folgt an dieser Stelle keine ausführliche Beschreibung der zugrundeliegenden Modelle, aber die von Solr implementierten Formeln werden mit einheitlicher Benennung dargestellt.

1) Die „DefaultSimilarity“ von Solr, die auf dem Vektorraummodell basiert:

$$score(doc, term) = \sqrt{tf} \cdot idf$$

$$idf = (log_e(\frac{numDocs}{df + 1}) + 1)$$

Die Retrievalscores werden für jeden Term separat berechnet und für die einzelnen Dokumente addiert. Das trifft auch auf die folgenden beiden Algorithmen zu. *tf* gibt die Termfrequenz an, also wie häufig der Term im *doc* vorkommt.

numDocs ist die Zahl der Dokumente in der Kollektion, *df* ist die Zahl der Dokumente, die den Term beinhalten. *e* ist die eulersche Zahl.

2) Der BM25-Algorithmus:

$$score(doc, term) = idf \cdot (k1 + 1) \cdot \frac{tf}{tf + (k1 \cdot (1 - b + b \cdot \frac{dl}{avgdl})}$$

$$idf = \left(1 + \frac{numDocs - df + 0.5}{df + 0.5}\right)$$

avgdl ist hierbei die durchschnittliche Länge des Feldes, für das der Score berechnet wird. Der Exaktheit halber sollte man erwähnen, dass *dl* (wohl aus Performancegründen) nur die ungefähre Länge des Feldes ist.

k1 und *b* sind Parameter, die man verändern kann. Folgende fünf Varianten werden verwendet:

a. k1 = 1,0, b = 0,75
b. k1 = 1,2, b = 0,75 (Standardwerte in Solr)
c. k1 = 1,4, b = 0,75
d. k1 = 1,2, b = 0,65
e. k1 = 1,2, b = 0,85

Ein Wert von *k1* von 0 bedeutet, dass nur der *idf* zum Ranking benutzt wird, während ein hoher Wert dafür sorgt, dass der Einfluss des *idf* geringer wird.

b gibt an, inwiefern eine Normalisierung der Scores in Abhängigkeit von der Länge der Dokumente stattfindet. Bei einem Wert von 0 hat die Dokumentlänge keinen Einfluss, bei höheren Werten sinkt der Score mit zunehmender Dokumentlänge bei gleicher Termfrequenz.

3) Language Models basierend auf der Jelinek-Mercer-Smoothing-Methode:

$$score(doc, term) = log_e(1 + \frac{(1 - \lambda) \cdot \frac{tf}{dl}}{\lambda \cdot (\frac{colTf + 1}{colDl + 1})})$$

colTf steht hier für *Collection Term Frequency* und bezeichnet die Häufigkeit des Vorkommens des Terms über alle Dokumente hinweg. Analog bezeichnet *colDl* die Summe der Längen des Feldes in allen Dokumenten.

Man kann hier den Wert von λ angeben; es wurden vier Werte getestet:

a. λ = 0,5
b. λ = 0,6
c. λ = 0,7 (Standardwert in Solr)
d. λ = 0,8

Nach (Zhai & Lafferty, 2001) eignen sich für Querys in langen Feldern λ-Werte um etwa 0,7 am besten und für Felder mit weniger Tokens wie dem Titel eher kleinere Werte.

4.2.2 PageRank

In diesem Unterpunkt wird zunächst das Verfahren beschrieben, mit dem die Werte berechnet wurden. Daraufhin wird erklärt, dass die Verwendung und Berechnung des PageRank bei der Kollektion problematisch sind.

4.2.2.1 Berechnung der Werte

Der PageRank wurde mit einem Verfahren implementiert, das auf der englischen Wikipedia-Seite zum PageRank als „Power Method“ bezeichnet wird:

1) *N***N*-Matrix *M* erstellen, wobei *N* der Anzahl der Dokumente der Kollektion entspricht
2) Für zwei Webseiten i, j ist der Wert von
   $M_{i,j} = \frac{1}{Zahl\ der\ von\ Seite\ j\ referenzierten\ Seiten}$, falls j i referenziert, und 0 falls nicht.

3) Jeder dieser Werte wird mit d multipliziert und zu jedem Wert in M wird $\frac{1-d}{N}$ addiert.
4) Der PageRank-Vektor $x(0)$ hat die Länge N und wird zunächst beliebig definiert.
5) Die Rechnung $x(t + 1) = M \times x(t)$ wird durchgeführt, bis die Summe der Änderungen der Werte von x(t) zu $x(t + 1)$ kleiner als ein bestimmter Grenzwert ist. (Wikipedia Contributors, 2015)

Für d wurde der Wert 0,5 gewählt und es wurden 40 Iterationen von Schritt 5 durchgeführt. Bei der Größe der Kollektion war das einzige Problem die Datenstruktur für die Speicherung der Matrix M.

4.2.2.2 Probleme bei der Verwendung des PageRank

Beim PageRank steht der „Random Surfer" im Mittelpunkt, also ein Benutzer, der zufällig den Webgraphen traversiert. Im Web gibt es einseitige (aber potenziell beidseitige) Links; eine Seite, die vor einer anderen angelegt wurde, kann sich ändern und auf die andere verweisen.

Eine wissenschaftliche Arbeit, die vor einer anderen veröffentlicht wurde, kann die andere nicht zitieren, d.h. Verweise können im Normalfall nur einseitig sein. Insofern ist es zwar aufgrund der identischen Berechnung der PageRank-Wert der Dokumente, der in dieser Arbeit verwendet wird, aber die Bezeichnung „PageRank" ist ohne Webgraphen in dieser Hinsicht unangemessen.

Es ist auch ein Problem, dass man den Einfluss, den neuere Arbeiten später haben werden, noch nicht bestimmen kann. Bernstam, et al., (2006) bezeichnen dies als „*Citation Lag*" und untersuchen, wie die Verbesserung der Retrievalwerte mit dem PageRank aussieht, wenn man die Zahl der Zitate auch für neuere Dokumente berechnet. Dafür muss man Zitate aus Dokumenten verwenden, die nicht in der Testkollektion sind. Damit befinden sich in den ersten 100 Suchergebnissen durchschnittlich etwa vier relevante Treffer mehr, was in ihrer Analyse eine signifikante Verbesserung darstellt.

4.2.3 Ontologien und Synonyme

Um den kombinierten Ansatz in einem echten System umsetzen zu können, wäre es notwendig, automatisch Synonyme für Begriffe zu finden. Zunächst war geplant, dies zu implementieren, allerdings war der Aufwand hierfür zu groß. Die Headings sind in

MeSH in einem Format, das sowohl das automatische Auffinden geeigneter Begriffe kompliziert macht als auch die Einschränkung auf bestimmte Begriffe.

Zur Veranschaulichung der Problematik ist hier eine Auswahl der Synonyme für die Alzheimer-Krankheit, die Teil des *TREC Genomics*-Topics 161 ist:

- Disease, Alzheimer
- Alzheimer Syndrome
- Syndrome, Alzheimer
- Alzheimer Type Senile Dementia
- Senile Dementia, Alzheimer Type
- Alzheimer's Disease
- Disease, Alzheimer's
- Early Onset Alzheimer Disease

(U.S. National Library of Medicine, 1998)

Bei der Erweiterung der Query erhöht sich der Retrievalscore mit jedem Vorkommen eines Terms, weswegen es sich negativ auswirken kann, dass hier häufig unterschiedliche Reihenfolgen derselben Wörter auftauchen. Das ist eine Teilmenge der Begriffe, in der vollständigen Liste kommt 24 Mal das Wort „Alzheimer“ vor.

Es wäre noch mit vertretbarem Aufwand machbar, solche Begriffe wie „Alzheimer Syndrome“ und „Syndrome, Alzheimer“ zusammenzufassen. Wenn man aber die Terme von „Early Onset Alzheimer Disease“ miteinbezieht, fügt man viele Dokumente zur Ergebnismenge hinzu, die sich mit dem Frühstadium irgendeiner Krankheit befassen, nicht nur mit Alzheimer. Ein System für die automatische Auswahl der Terme wurde im Rahmen dieser Arbeit nicht entwickelt.

MeSH beinhaltet nicht für alle Begriffe der TREC-Topics Einträge. Da die automatische Erweiterung der Querys nicht implementiert wurde, konnten auch andere Datenbanken benutzt werden; der Aufwand, eine Schnittstelle für die Ontologien zu entwickeln, fiel dadurch weg.

Ein Beispiel für ein Gen, das man in MeSH nicht findet, ist der andere Teil des TREC-Topics, in dem Alzheimer als untersuchte Krankheit vorkommt: das IDE-Gen. Das Gen ist in MeSH nur ausgeschrieben enthalten; bei GeneCards findet man das Akronym, das für **I**nsulin **D**egrading **E**nzyme steht. Wenn man MeSH als einzige Ontologie zur Verfügung hätte, würde das dazu führen, dass der Score stärker von Alzheimer beeinflusst würde, weil Synonyme für das Konzept häufiger in der Query vorkommen.

Bei einer Implementierung der Suchmaschine für den „echten“ Einsatz wäre es wichtig, dass man zu einem ergänzenden System mit den verwendeten Ontologien eine passende Schnittstelle erstellen kann, um die Querys automatisch zu erweitern. Ob das mit *GeneCards* möglich ist wurde nicht überprüft. Man darf die Seite für akademische Zwecke kostenlos verwenden, das ist für diese Arbeit ausreichend.

Die Ontologien wurden in dieser Arbeit zusammenfassend so verwendet, dass nach den Termen in den TREC-Topics gesucht wurde. Die in den Ontologien enthaltenen Oberbegriffe und Synonyme stellen zusätzliche Suchterme dar, deren Einfluss separat gewichtet wird.

4.2.4 Kombinierter Score

Der kombinierte Score wurde nur für die Terme berechnet, die in den TREC-Topics enthalten sind; insgesamt sind das 39 Begriffe. Wenn eine Suche nach den Termen oder einem Synonym für die Terme im Titel oder im Abstract der Dokumente (mit den bei der Evaluation als am besten eingestuften Indexierungs- und Rankingmethoden, s. Punkt 5.2.2) einen Treffer ergab, so wurden die Dokumente zu *d1* hinzugefügt.

Als Thema für diese themenabhängigen Scores wurden nicht die gesamten TREC-Topics ausgewählt. Dies hätte bedeutet, dass man die in Punkt 3.2 vorgestellten Werte für die Ergebnisse einer Suchanfrage berechnet. Stattdessen erhielten die Dokumente Scores für die einzelnen Begriffe, die in den Topics enthalten sind; alle Scores, die für ein Topic relevant sind, werden dann pro Dokument aufsummiert und bilden den Wert des kombinierten Scores für das Dokument.

4.2.5 Ausschluss anderer möglicher Rankingfaktoren

Die Faktoren wurden alle so ausgesucht, dass sich damit relativ leicht vergleichbare Scores für die Dokumente berechnen lassen. Die Werte, die Solr berechnet, der PageRank und die kombinierten Scores sind alle von ähnlicher Natur:

- Man kann sie als Zahlen ausdrücken.
- Es ist möglich, ihren Einfluss durch Gewichtungen (die ebenfalls Zahlen sind) bestimmen
- Höhere Werte sollen zu einem höheren Retrievalscore führen.
- Es müssen nur positive Werte betrachtet werden (s. Punkt 4.3.1).

Deswegen werden mögliche Faktoren wie beispielsweise das Veröffentlichungsdatum der Artikel oder der Abstand der Suchterme in den Dokumenten ausgeschlossen.

Einschränkungen auf eine bestimmte Art von Forschungsarbeit oder untersuchter Gruppe sind auch ein Rankingfaktor, der in dieser Arbeit nicht betrachtet wird. Das sollte man erwähnen, weil diese Filter in PubMed verfügbar sind und zum Teil automatisch vorgeschlagen werden. Beispielsweise wandelt PubMed den Suchbegriff „causes“ um in die komplexe (Teil-)Query ("etiology"[Subheading] OR "etiology"[All Fields] OR "causes"[All Fields] OR "causality"[MeSH Terms] OR "causality"[All Fields]).

Es wurde davon abgesehen, so komplexe Suchanfragen zu formulieren. Man hätte damit einen Faktor, dessen Einfluss sich nicht direkt in Zahlen festlegen lässt: Es ist nicht möglich, zu bestimmen, was halb oder doppelt so gravierend wäre wie die Einschränkung auf einen Typ von Studie.

Dieser Ausschluss von einigen potenziellen Rankingfaktoren steht nicht im Widerspruch mit dem Anspruch dieser Arbeit, zu überprüfen, inwiefern die Kombination von diversen Rankingfaktoren die Suchergebnisse verbessern können. Es wird nicht versucht, die besten Rankingfaktoren für PubMed zu bestimmen, sondern es wird untersucht, ob die Kombination von Faktoren grundsätzlich Verbesserungen der Qualität der Ergebnisse bringt.

4.3 *Learning to Rank* zur Optimierung der Systemvarianten

> A new trend has recently arisen in document retrieval, particularly in web search, that is, to employ machine learning techniques to automatically construct the ranking model $f(q, d)$. (Li, 2010)

Das Zitat beschreibt der Einsatz von maschinellen Lernverfahren zur systemzentrierten Optimierung der Suchmaschinen, was auch als *Learning to Rank* bezeichnet wird. Man kann solche Machine Learning-Techniken anwenden, wenn man in einer Suchmaschine mehrere Retrieval- bzw. Rankingfaktoren hat, deren Einfluss auf die Ergebnisse sich festgestellt werden soll. Hierfür benötigt man z.B. eine Testkollektion mit Relevanzbeurteilungen, wie sie für TREC Genomics gegeben sind.

Damit lassen sich *Ranking-Modelle* erstellen und trainieren (formell *f(q, d)*: eine Funktion, die für jedes Dokument und jede Query einen bestimmten Wert zurückgibt). In dieser Arbeit wird *Learning to Rank* verwendet, um sicherzugehen, dass die einzelnen

Systemvarianten jeweils optimal konfiguriert sind, sodass man die mit den Varianten maximal erzielbare Qualität messen kann. Das ist wichtig, um entscheiden zu können, ob eine Variante vom Ansatz her einer anderen über- oder unterlegen ist und nicht, weil die Gewichtungen der Faktoren für eine Variante nicht richtig eingestellt waren.

Letztendlich werden Gewichtungsvarianten für Felder der Dokumente verglichen. So kann man entscheiden, welche dieser Werte man einen Einfluss auf das Ranking der Ergebnisse haben lässt.

4.3.1 Logistische Regression und Support Vector Machines

Zuerst wurde versucht, die Optimierung der Gewichtung mit logistischer Regression durchzuführen. Man kann für jede Query die Relevanz und die Scores der einzelnen Felder eines Dokuments bestimmen, was eine Voraussetzung für die Erstellung und Optimierung eines Regressionsmodells darstellt.

Ein großes Problem dabei ist, dass alle Implementierungen für logistische Regression, die für diese Arbeit in Betracht gezogen wurden, von Grund auf auf eine Minimierung der Fehler 1. und 2. Art hin ausgelegt sind. Es war also nicht möglich, eine Anpassung vorzunehmen, mit der das Programm die Suchmaschine auf bestimmte Evaluationsmaße hin optimiert. Das wäre aber sehr wichtig, wie in den Punkten 4.3.2 und 5.1 genauer erklärt wird.

Ein weiterer Nachteil bei der Verwendung von Regressionsanalyse ist, dass für mehrere Faktoren ihre Werte negativ in die Beurteilung der Relevanz eines Dokuments einfließen sollten. In den hier untersuchten Varianten gibt es allerdings keine solchen Faktoren (wie z.B. den Abstand zwischen Suchtermen in einem Dokument). Was ein negativer Wert aussagt, ist: „Wenn einer der Suchterme häufig in diesem Feld vorkommt, dann senkt das die Wahrscheinlichkeit, mit der das Dokument relevant ist.“

Das mag zwar in der Praxis für die Testkollektion und die Querys stimmen, muss aber hinterfragt werden: Der grundlegende Ansatz von allen verwendeten Rankingalgorithmen ist, dass die Ähnlichkeit von Query und Dokument bestimmen, welchen Rang das Dokument in der Ergebnisliste hat oder ob es überhaupt darin auftaucht. Das negative Vorzeichen würde bei einer hohen Übereinstimmung von Query und Dokument dazu führen, dass das Dokument als nicht relevant eingestuft wird. Deswegen wurde nicht akzeptiert, dass Werte ein negatives Vorzeichen erhalten.

Support Vector Machines wurden ebenfalls als *Learning to Rank*-Verfahren für die Suchmaschinenvarianten in Betracht gezogen. Da es sich beim Berechnen eines Scores um ein Rankingproblem handelt, man aber nur Daten für ein Klassifikationsproblem zur Verfügung hat, wären *Support Vector Machines* theoretisch sauberer (Manning, Raghavan, & Schütze, 2009). Damit würde man für jedes Dokument entscheiden, ob es relevant ist oder nicht und nicht wie relevant es im Vergleich zu anderen ist.

Hierbei ergaben sich jedoch ähnliche Probleme wie bei der Regressionsanalyse. Darüber hinaus gab es mit den bestehenden Implementierungen hierfür massive Performanceprobleme, sodass ein eigenes *Learning to Rank*-Verfahren entwickelt wurde.

4.3.2 Eigener Ansatz für *Learning to Rank*

Bei der systemzentrierten Evaluation soll man eine Maßzahl wählen, die möglichst den Bedürfnissen der Benutzer entspricht (siehe auch: 5.1). Wenn man es schafft, eine Suchmaschine so zu optimieren, dass sie den bestmöglichen Wert für eine Evaluationsmethode erreicht, optimiert man sie nach dieser Logik für die Benutzer des Systems.

Es wurden bereits Methoden untersucht, die basierend auf diesem Grundgedanken mit Machine Learning Suchergebnisse für eine bestimmte Metrik optimal zu ranken. Yue, Finley, Radlinski, & Joachims (2007) haben eine Methode entwickelt, mit der mit *Support Vector Machines* das Rankingmodell einer Suchmaschine so bestimmt wird, dass die *Mean Average Precision*-Maßzahl (s. 5.1.2) möglichst hoch wird. McFee & Lanckriet (2010) beschreiben ein Verfahren, mit dem man ein Rankingmodell so trainieren kann, dass es theoretisch hinsichtlich einer beliebigen Evaluationsmetrik optimal ist.

Keine dieser Methoden konnte in dieser Arbeit verwendet werden, da der Quellcode jeweils nicht frei verfügbar ist und es noch keine Anpassungen für eine der verwendeten Metriken (ein gewichtetes harmonische Mittel) gibt. Deswegen wurde eine neue Methode entwickelt, die denselben Grundgedanken verfolgt, also Suchergebnisse hinsichtlich eines beliebigen Evaluationsmaßes optimieren kann. Das Verfahren wird im folgenden Unterpunkt vorgestellt.

Aus mathematischer Sicht ist das Verfahren unpräziser als das von McFee & Lanckriet, aber es ist für diese Arbeit ausreichend. Das wird in Punkt 4.3.2.2 genauer diskutiert.

4.3.2.1 Verfahren

Das Vorgehen funktioniert wie folgt:

1) Es werden Sets von Feldern bzw. Werten festgelegt, für die jedes Feld mit allen Feldern aus den anderen Sets kombiniert wird.
2) Für eine Kombination von Feldern aus den Sets werden alle Werte über alle Topics hinweg eingelesen.
3) Für die Werte wird der Durchschnitt gebildet und sie werden so sortiert, dass das Feld mit dem höchsten Durchschnitt als „Basis“ dient.
4) Für die anderen Felder werden jeweils 11 Gewichtungen berechnet: Zunächst wird die Potenz von 3* berechnet, die den Durchschnitt des Felds am nächsten zur Basis bringt. Die Gewichtungen sind dann die 5 nächstniedrigeren und nächsthöheren Potenzen von 3.
5) Für alle Kombinationen von Gewichtungen der Felder werden für 18 der 26 Topics die X Dokumente mit den höchsten Scores berechnet und für die Topics werden jeweils Recall und Precision berechnet.
6) Die Kombination von Gewichtungen mit dem höchsten F2-Wert für den durchschnittlichen Recall und die durchschnittliche Precision über alle Topics ist das vorläufige Ergebnis für die Kombination der Felder.
7) Es werden weitere Gewichtungen rund um die optimale Gewichtung getestet, um präzisere Ergebnisse zu erhalten. Wenn 3^X als optimale Gewichtung gefunden wurde, werden noch die Werte für $3^{X-0.8}$, $3^{X-0.6}$, $3^{X-0.4}$, $3^{X-0.2}$, $3^{X+0.2}$, $3^{X+0.4}$, $3^{X+0.6}$ und $3^{X+0.8}$ berechnet.
8) Für die besten Gewichtungen wird nun der Score für die verbleibenden 8 Topics berechnet und notiert. Dies ist der Score der Variante.

Zu „*“ (Punkt 4)): mit „3“ als Basis kann man mit einer überschaubaren Menge an Gewichtungen ein relativ breites Spektrum abdecken (wenn der Durchschnitt eines Feldes etwa gleich mit dem der Basis ist sind die getesteten Gewichtungen 0,0014, 0,0041, 0,012, 0,037, 0,11, 0,33, 1, 3, 9, 27, 81, 243 und 729, jeweils im Vergleich zur Basis).

Um zu vermeiden, dass durch die zufällige Auswahl eines Trainings- und Testsets falsche Schlüsse gezogen werden, sollte man das Verfahren für verschiedene Sets wiederholen.

Im Anhang befindet sich das Dokument „Beispiel für das *Learning to Rank*-Verfahren“, in dem das Verfahren anhand von konkreten Feldern und Zahlenwerten erläutert wird.

4.3.2.2 Diskussion des Verfahrens

Ein Nachteil, der als nicht besonders schwerwiegend angesehen wird, ist die theoretische Unsauberkeit des Verfahrens. Die besten Varianten werden durch Ausprobieren gefunden, da sich der Einfluss der Änderung der Gewichtung eines Feldes auf Maßzahlen wie die MAP dadurch sehr leicht und auf anderen Wegen schwerer bestimmen lässt. Die getesteten Gewichtungen decken jedoch so viele Kombinationen ab, dass man davon ausgehen kann, dass die bestimmten Werte nahe genug an den optimalen Werten sind, um keine falschen Schlüsse zu ziehen.

In Punkt 5 wird häufiger von Performanceproblemen und dem Zeitaufwand der Berechnungen gesprochen. Das Verfahren ist auf Skalierbarkeit ausgelegt und wurde auch so implementiert, dass mehrere Berechnungen parallel laufen können. Die Probleme liegen also nicht am Verfahren an sich, sondern daran, dass die Berechnungen auf einem PC durchgeführt wurden und vermutlich bei mehr als drei Rankingfaktoren für ein Rechencluster angemessen wären.

Einer der großen Pluspunkte des Verfahrens ist die Austauschbarkeit der Evaluationsfunktion. Die Retrievalscores der Ergebnisse werden berechnet und die Dokumente entsprechend sortiert, dann erfolgt die Berechnung der Maßzahl. Man arbeitet also mit Ergebnissen, wie sie tatsächlich auf einer Suchergebnisseite angezeigt würden, und kann bestimmen, wie stark der Einfluss der Rankingfaktoren relativ zueinander sein soll, damit sie das beste Ergebnis für die Evaluationsfunktion erzielen.

Ein weiterer Vorteil gegenüber den anderen untersuchten Optimierungsmethoden ist, dass man die Qualität der Suchmaschine über verschiedene Topics mit einer unterschiedlichen Zahl an Ergebnissen leichter bestimmen kann. Um die Tatsache auszugleichen, dass es für manche Topics mehr Ergebnisse gibt, hätte man für jedes Topic alle nicht gefundenen Dokumente mit dem Wert „0“ allen Feldern hinzufügen müssen. Bei der Größe der Kollektion und der Zahl der Topics hätte das die Performance stark beeinträchtigt.

Der gravierendste Nachteil des *Learning to Rank*-Verfahrens ist, dass keine Möglichkeit vorgesehen ist, das Verfahren zur Klassifikation von Dokumenten als relevant bzw. nicht relevant zu verwenden, sondern nur zum Ranking von Dokumenten. In den verfügbaren Relevanzbeurteilungen sind die Dokumente klassifiziert, nicht gerankt. Mit den verwendeten Faktoren stellt dies kein Problem dar, aber man müsste das Verfahren weiterentwickeln, wenn man andere Arten von Rankingfaktoren benutzen möchte.

5 Evaluation der Systemvarianten

In diesem Kapitel wird beschrieben, mit welchen Maßzahlen die Qualität der Suchergebnisse der verschiedenen Systemvarianten bestimmt wird. Im Anschluss daran werden die Ergebnisse der Evaluation präsentiert.

5.1 Wahl von Maßzahlen basierend auf Nutzerbedürfnissen

Zunächst werden in diesem Unterpunkt zwei mögliche Anforderungen an eine Suchmaschine für PubMed vorgestellt, die jeweils zu anderen Evaluationsmetriken führen, um den Bedürfnissen von Suchenden zu entsprechen.

Zur Orientierung für die Interpretation der Ergebnisse der Systemvarianten wird dann gezeigt, welche Ergebnisse zu welchen Werten bei den Maßzahlen führen und welche Ergebnisse gewünscht sind.

5.1.1 F_2-Score als Qualitätsmaß bei der Erstellung von Systematic Reviews

Ein mögliches Einsatzgebiet für die Suchmaschinen wäre die Suche nach Artikeln zu einem Thema bei der Erstellung von *Systematic Reviews*. Das sind sind Meta-Analysen, die den der Stand der Forschung eines bestimmten Gebiets beschreiben. Sie enthalten Informationen darüber, was schon untersucht wurde, wie es untersucht wurde und sollen z.B. für Ärzte fundierte Schlüsse darüber zulassen, welche Therapien funktionieren und welche nicht. Für diese Systematic Reviews ist es notwendig, möglichst alle relevanten Dokumente zu einem Thema zu finden. Hierfür lautet eine Empfehlung, dass Experten aus den in den Reviews behandelten Fachgebieten mit Retrievalexperten zusammenarbeiten sollen (Uman, 2011).

Insofern kann eine Anforderung für Suchmaschinen zu PubMed sein, einen möglichst hohen Recall zu haben. Da ein sehr hoher Recall-Wert bei einer sehr geringen Precision in der Praxis jedoch nicht ausreicht (weil z.B. eine Suchmaschine, die immer alle

Dokumente als Treffer anzeigt, einen Recall von 1 hat), bietet sich ein gewichtetes harmonische Mittel von Recall und Precision als Kennzahl an:

$$F_\beta = (\beta^2 + 1) * \frac{precision * recall}{(\beta^2 * precision) + recall}$$

(Manning, Raghavan, & Schütze, 2009)

Ein hoher Recall wird als wichtiger angesehen, weswegen mit $\beta = 2$ gerechnet wird; damit hat der Wert des Recall einen höheren Einfluss auf den Score als der der Precision. Es werden hier nicht alle Ergebnisse in die Berechnung miteinbezogen, sondern Precision und Recall werden für die X obersten Ergebnisse berechnet. Normalerweise sind dies X = 50, 100, 150, 200 und 300 Ergebnisse berechnet, bei Systemvarianten mit aufwendigeren Berechnungen nur X = 100.

Das liegt daran, dass keine Klassifikation der Ergebnisse in relevant bzw. nicht relevant erfolgt, sondern „nur" ein Ranking der Dokumente. Das kann dazu führen, dass bei einem Topic ein Dokument mit einem Score von beispielsweise 0,5 als relevant eingestuft wird und bei einem anderen nicht.

Die Einschränkung auf X Ergebnisse (die auch bei 5.1.2 vorliegt) ist keine Schwäche des Evaluationsverfahrens, die hingenommen wird, sondern ist auch bewusst gewählt, da dies das Nutzerverhalten in den jeweiligen Fällen wiederspiegelt: Hier lautet die Annahme, dass selbst eine sehr ausführlichen Suche nicht über das 300. Ergebnis hinausgeht.

5.1.2 Mean Average Precision als Maßzahl für durchschnittliche Benutzung

Für TREC-Studien ist die *Mean Average Precision* (MAP) eine Standardmaßzahl, da man damit die Precision einer Suchmaschine über alle verschiedenen Recall-Stufen hinweg in einer Zahl ausdrücken kann (Manning, Raghavan, & Schütze, 2009).

Die Formel hierfür lautet (für ein Informationsbedürfnis q, rel(q) = Zahl der relevanten Dokumente für q und n = Position des relevanten Dokuments k):

$$AP(q) = \frac{\sum_{k=1}^{rel(q)} Precision@n}{rel(q)}$$

(Manning, Raghavan, & Schütze, 2009)

Die Precision-Werte an den Positionen der relevanten Dokumente werden addiert und durch die Zahl der relevanten Dokumente geteilt. Wenn ein relevantes Dokument

in den Suchergebnissen enthalten ist, wird als Precision „0“ verwendet. Das arithmetische Mittel der *AP* für verschiedene Informationsbedürfnisse ist die *Mean Average Precision*.

Die *MAP* ist ein passendes Maß für die durchschnittliche Verwendung von PubMed. Der Vergleich einer Logstudie für PubMed (Dogan, Murray, Névéol, & Lu, 2009) mit einer Logstudie zu einer Websuchmaschine (Silverstein, Marais, Henzinger, & Moricz, 1999) zeigt, dass folgende Aussagen auf beide Suchkontexte zutreffen:

- Die meisten Suchsessions bestehen aus fünf Querys oder weniger (85% bei Pubmed, mehr als 95% im Web).
- Die Ergebnisse auf der ersten Ergebnisseite erhalten viel mehr Klicks als die auf den folgenden Seiten (mehr als 80% bei PubMed, mehr als 85% im Web)

Diese (und andere) Kennzahlen deuten darauf hin, dass die Suchen in PubMed zwar im Durchschnitt etwas ausführlicher sind als im Web, die Mehrheit der Benutzer jedoch beide Suchmaschinen eher für kurze Suchvorgänge nutzt.

In dieser Arbeit wird nicht exakt die oben gezeigte Formel verwendet, sondern jeweils eine MAP@X berechnet. Hierbei steht im Nenner der Formel X, falls rel(q) > X. PubMed zeigt standardmäßig 20 Treffer an, weswegen die Ergebnisse für X = 20, 40, 60, 80 und 100 berechnet werden, bei Systemvarianten mit aufwendigeren Berechnungen nur für X = 20.

5.1.3 Erwartete Werte

Ein direkter Vergleich mit Werten aus der Literatur ist praktisch unmöglich, da die Systeme, die für den TREC Track 2006 entwickelt wurden, normalerweise mit den vollständigen (HTML-)Artikeln arbeiten, während in dieser Arbeit nur die Abstracts und Titel der Dokumente benutzt werden. Beim TREC Track wurden (als die Konferenz ursprünglich stattfand) 91 Systemvarianten eingereicht. Die höchsten Werte für die MAP waren etwas unter 0,55, der Median lag bei ca. 0,31 (Hersh, Cohen, & Roberts, 2006).

Es folgen einige Überlegungen und Schätzungen, die die Interpretation der in dieser Arbeit erzielten Scores vereinfachen sollten.

Insgesamt sind für alle Topics 1381 Dokumente relevant, das sind durchschnittlich 53 Dokumente pro Query. Wenn diese Dokumente gleichmäßig auf alle Topics verteilt wären und es für jedes Topic mindestens 300 Treffer gibt, könnte die durchschnittliche

Precision bei X = 300 nicht höher als 0,18 sein und bei X = 100 nicht höher als 0,53. Werte jenseits von 0,8 sind für den F_2-Score dadurch praktisch unerreichbar.

Die relevanten Dokumente sind jedoch nicht gleich verteilt, für fünf Topics gibt es mehr als 100 relevante Ergebnisse. Für zwei davon sind mehr als 300 Dokumente wichtig, ein Recall von 100% ist demzufolge (selbst für X = 300) unmöglich. Der für X = 100 maximal erreichbare durchschnittliche Recall liegt bei ca. 0,87.

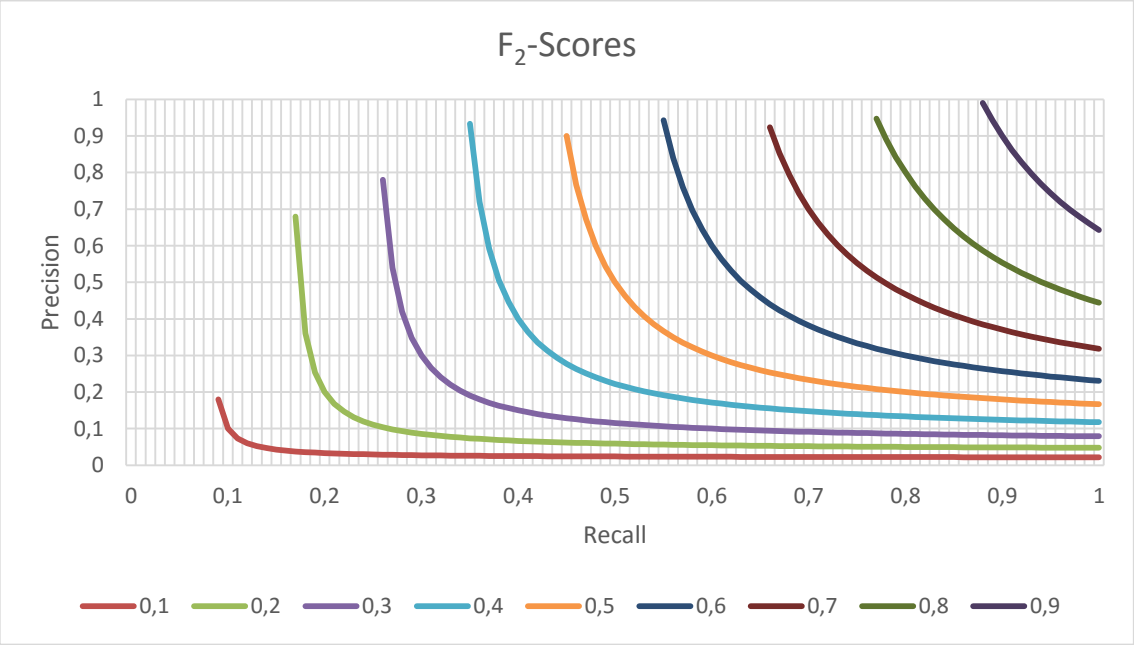


Abbildung 5: F_2-Scores in Abhängigkeit von Precision und Recall

Dieses Diagramm zeigt, welche Precision in Abhängigkeit vom Recall notwendig ist, um bestimmte F_2-Werte zu erreichen. Die neun Kurven entsprechen den Werten 0,1, 0,2, 0,3 usw. für F_2. Für einen F_2-Score von 0,2 (grüne Linie bzw. zweite von links unten) ist bei einem Recall von 0,2 eine Precision von 0,2 notwendig, bei einem Recall von 0,3 nur noch eine Precision von etwas unter 0,1 usw.

Hohe Recall-Werte (ab 0,7) wären notwendig, wenn man mit dem System den Stand der Forschung zuverlässig bestimmen können soll. Bei F_2-Werten von unter 0,4 oder 0,5 sind Precision und/oder Recall so gering, dass man nicht behaupten kann, dass es mit der Suchmaschine möglich ist, den Stand der Forschung zu erfassen.

Wenn der Einfluss eines Rankingfaktor eine Erhöhung um etwa 0,05 bewirkt, kann man das als signifikante Verbesserung betrachten. Verbesserungen zwischen 0,01 und 0,05 sind vermutlich für Benutzer spürbar.

Bei der *MAP@X* wären für geringe *X* (20, 40) theoretisch Werte bis ca. 1 möglich. Werte um 0,3 würden bedeuten, dass von 20 Ergebnissen die ersten sechs relevant sind oder ca. jedes zweite Ergebnis relevant ist. Während immer noch große Verbesserungen möglich sind, wäre 0,3 also ein Wert, mit dem man bei oberflächlichen Suchen zufriedenstellende Ergebnisse haben kann. Theoretisch kann es auch sein, dass ein Wert von beispielsweise 0,017 ausreicht, um ein Informationsbedürfnis zu befriedigen (wenn ein relevantes Dokument an der dritten Position der Suchergebnisse erscheint, das dem Suchenden ausreicht). Werte von mindestens 0,3 sind aber wünschenswert, da es sich um Durchschnittswerte handelt und sich bei dem Wert vermutlich für die meisten Querys mehrere relevante Dokumente in den ersten Rängen befinden.

Werte um 0,5 wären bei X = 20 sehr gut, bei X = 40 bräuchte man hierfür (falls es 40 oder mehr relevante Ergebnisse für das Topic gibt) mindestens 20 relevante Ergebnisse, wenn diese an den Positionen 1-20 sind. Die *MAP@X*-Scores sind hier für kleine X leichter zu interpretieren. Für X >= 50 bedeuten Werte jenseits von 0,3 oder 0,4, dass man die Informationsbedürfnisse der Nutzer häufiger befriedigen kann, die Werte sind jedoch in den meisten Fällen ausreichend.

5.2 Ergebnisse der Tests

In diesem Punkt werden der Reihe nach die Systemvarianten *Vanilla*, *Feldtypen*, *PageRank*, *Ontologien* und *Kombinierter Score* und die Ergebnisse ihrer jeweiligen Evaluation vorgestellt. Es folgt eine Zusammenfassung der Ergebnisse und eine Überprüfung, ob die Kombination der Rankingfaktoren noch weitere Verbesserungen bringt.

Die Ergebnisse für die MAP- und die F_2-Scores stammen aus vier Tests mit verschiedenen Trainings- und Testsets; es sind jeweils das arithmetische Mittel sowie die niedrigsten und höchsten erzielten Werte pro Variante angegeben. Das arithmetische Mittel zeigt an, wie die Werte durchschnittlich bei Suchanfragen in der Praxis ausfallen würden.

5.2.1 Vanilla

Die *Vanilla*-Variante dient als Referenzwert für die anderen Varianten. Es wird nach den Begriffen der TREC-Queries im Titel und Abstract der Dokumente gesucht, wobei als Indexierungsvariante „simple“ und als Rankingalgorithmus der TF*IDF verwendet werden.

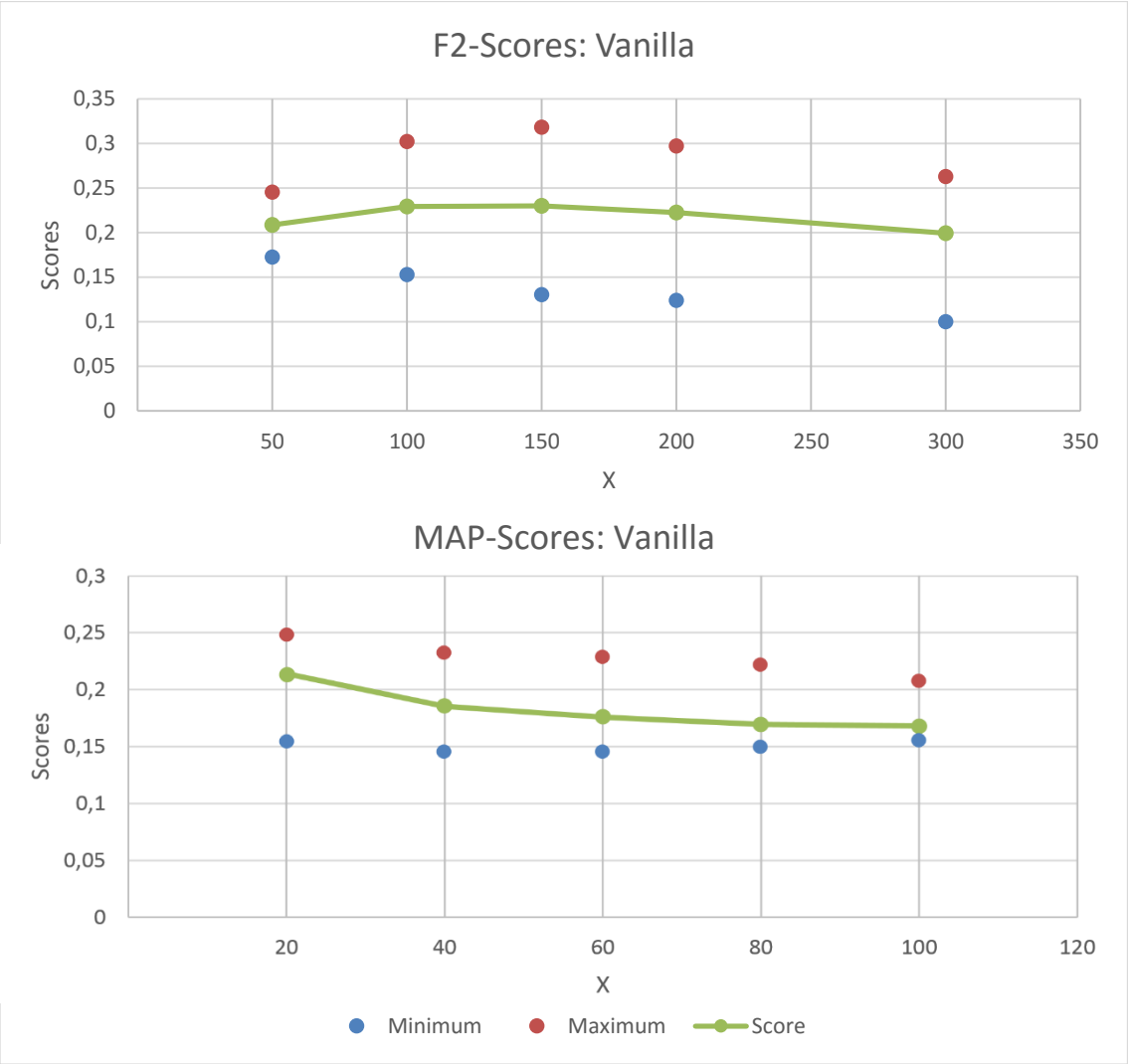


Abbildung 6: Ergebnisse der Evaluation der Vanilla-Variante

Den Verlauf der Kurve des F_2-Scores kann man folgendermaßen erklären: Der Recall wächst mit der Zahl der betrachteten Dokumente, was zunächst einer Erhöhung des Scores führt. Ab 150 betrachteten Dokumenten beginnt jedoch die Precision so gering zu werden, dass die Scores nicht weiter steigen.

Dass die MAP@X-Werte zunächst sinken und dann einen bestimmten Wert nicht mehr unterschreiten, ist nicht überraschend. Der Nenner steigt irgendwann nicht mehr für alle Topics mit X (wenn X größer wird als die Zahl der relevanten Dokumente), es kommen aber weitere gefundene Dokumente hinzu, die den Wert geringfügig erhöhen. Für sehr hohe X (jenseits von 300) müsste eine leichte Steigerung des Wertes erfolgen.

Insgesamt sind die Werte beider Maße nicht hoch genug, dass man davon ausgehen könnte, dass die entsprechenden Anforderungen der Nutzer erfüllt wären. Vor allem

die F_2-Werte sind zu niedrig, es befinden sich bei weitem nicht alle relevanten Dokumente in den Suchergebnissen (der Recall liegt für X = 300 etwa bei 0,35 bei einer Precision von ca. 0,08).

5.2.2 Wahl der optimalen Feldtypen

Bei dieser Systemvariante wird versucht, durch unterschiedliche Indexierungsvarianten und Rankingalgorithmen die bestmöglichen Ergebnisse zu erzielen, ohne dass der *PageRank* oder *MeSH* verwendet wird. Technisch gesehen werden damit zwei Rankingfaktoren auf einmal untersucht, aber da diese in Solr zu einem Feldtyp zusammengefasst werden, bietet es sich an, das an dieser Stelle auch zu tun.

Alle Feldtypen des Abstracts wurden bei den Querys mit allen Feldtypen des Titels kombiniert. Es schien nicht sinnvoll, verschiedene Indexierungsvarianten eines Feldes gleichzeitig in das Ranking miteinzubeziehen, auch unter Anbetracht der Tatsache, dass das die Laufzeit um ein Vielfaches verlängert hätte und diese durchaus ein Problem darstellte:

Es mussten für diesen Rankingfaktor Scores für 1600 Systemvarianten für 4 Trainings- bzw. Testsets berechnet werden, was zu einer hohen Laufzeit führte und dazu, dass nur Werte für F_2@100 bzw. MAP@20 berechnet wurden.

Feld	Feldtyp Indexierung	Ranking	max. F_2@100-Score	max. MAP@20-Score
abstract	**worddelimiter**	BM25	0.273	0.299
		LM	**0.275**	**0.304**
		Vektor	0.266	0.265
	stemming	BM25	0.272	0.281
		LM	0.266	0.294
		Vektor	0.264	0.249
	simple	BM25	0.238	0.264
		LM	0.243	0.287
		Vektor	0.235	0.241
	remove_nonnumeric	BM25	0.250	0.269
		LM	0.247	0.303
		Vektor	0.247	0.251
title	**worddelimiter**	BM25	0.272	0,3029
		LM	**0.275**	0,3028
		Vektor	0.273	0,301
	stemming	BM25	0.2731	0,302
		LM	0.2733	0,303
		Vektor	0.2725	**0,304**
	simple	BM25	0.2723	0,298
		LM	0.2721	0,301
		Vektor	0.2725	0,299
	remove_nonnumeric	BM25	0.274	0,3030
		LM	0.274	0,3033
		Vektor	0.271	0,3030

Abbildung 7: Ergebnisse der Evaluation mit verschiedenen Feldtypen

(LM = *Language Model*);

Vergleichswerte (Vanilla): F_2@100 = 0,229, MAP@20 = 0,214

Die Daten in dieser Tabelle sind folgendermaßen dargestellt:

Ganz links steht jeweils ein Feld der Dokumente. In den nächsten beiden Spalten wird der Feldtyp des Dokuments, bestehend aus Indexierungsvariante und Rankingalgorithmus, festgelegt. Die Scores in den zwei rechten Spalten sind jeweils das arithmetische Mittel der Scores der vier Durchläufe des Machine Learning-Algorithmus mit unterschiedlichen Trainingssets. Die grün markierten Zeilen zeigen, für welche Kombination von Indexierung und Ranking des Felds in der linken Spalte der höchste Score erzielt wurde.

Für den BM25-Algorithmus und Language Models wurden jeweils mehrere Varianten (mit unterschiedlichen Werten für die notwendigen Parameter) getestet. Beim BM25

erzielten die Varianten b) und e) in nahezu allen Fällen die höchsten Werte, bei Language Models meist die Variante b). Das bedeutet für beide Rankingmethoden, dass sich eine große Länge der Felder der Dokumente negativer auswirken sollte als in den Standardeinstellungen von Solr.

Der Abstract ist das „wichtigere" Feld bzw. die Qualität des Rankings hängt stärker davon ab als vom Titel. Das sieht man daran, dass für jeden Feldtyp des Titels eine Kombination gefunden wurde, mit der F_2@100-Scores von über 0,27 erzielt wurden, aber nur für drei Feldtypen des Abstracts. Bei der MAP@20 verhält es sich ähnlich.

Die Verbesserung der Scores ist hoch genug, um behaupten zu können, dass die Wahl der Indexierungsvariante und des Rankingalgorithmus wichtig sind und einen hohen Einfluss auf die Qualität der Ergebnisse haben.

Da bei beiden Feldern das Maximum für den F_2-Score mit dem *Analyzer* „worddelimiter" und Language Models als Rankingalgorithmus erzielt wurde und für die *MAP@20* nur eine Verschlechterung von ca. 0,001 gegenüber dem Maximum eintritt, wenn man diesen Feldtyp für Abstract und Titel übernimmt, wird dies als optimaler Typ für beide Felder bestimmt.

5.2.3 PageRank

Beim PageRank als Rankingfaktor gibt es die Optionen, ihn zum Score eines Dokuments zu addieren oder den Score damit zu multiplizieren. Da die Werte, die für den *PageRank* auftreten, z.T. um einige Zehnerpotenzen auseinander liegen, bestimmt der *PageRank* bei einer Multiplikation fast alleine den Score eines Dokuments; selbst hohe Werte in allen anderen Feldern können einen schlechten Score nicht ausgleichen. Dies hat dazu geführt, dass die erzielten Werte viel schlechter waren (< 0,1) und hier nicht aufgeführt werden.

Die Systemvariante „PageRank" zeigt im Vergleich zur „Vanilla"-Variante folgende Veränderung:

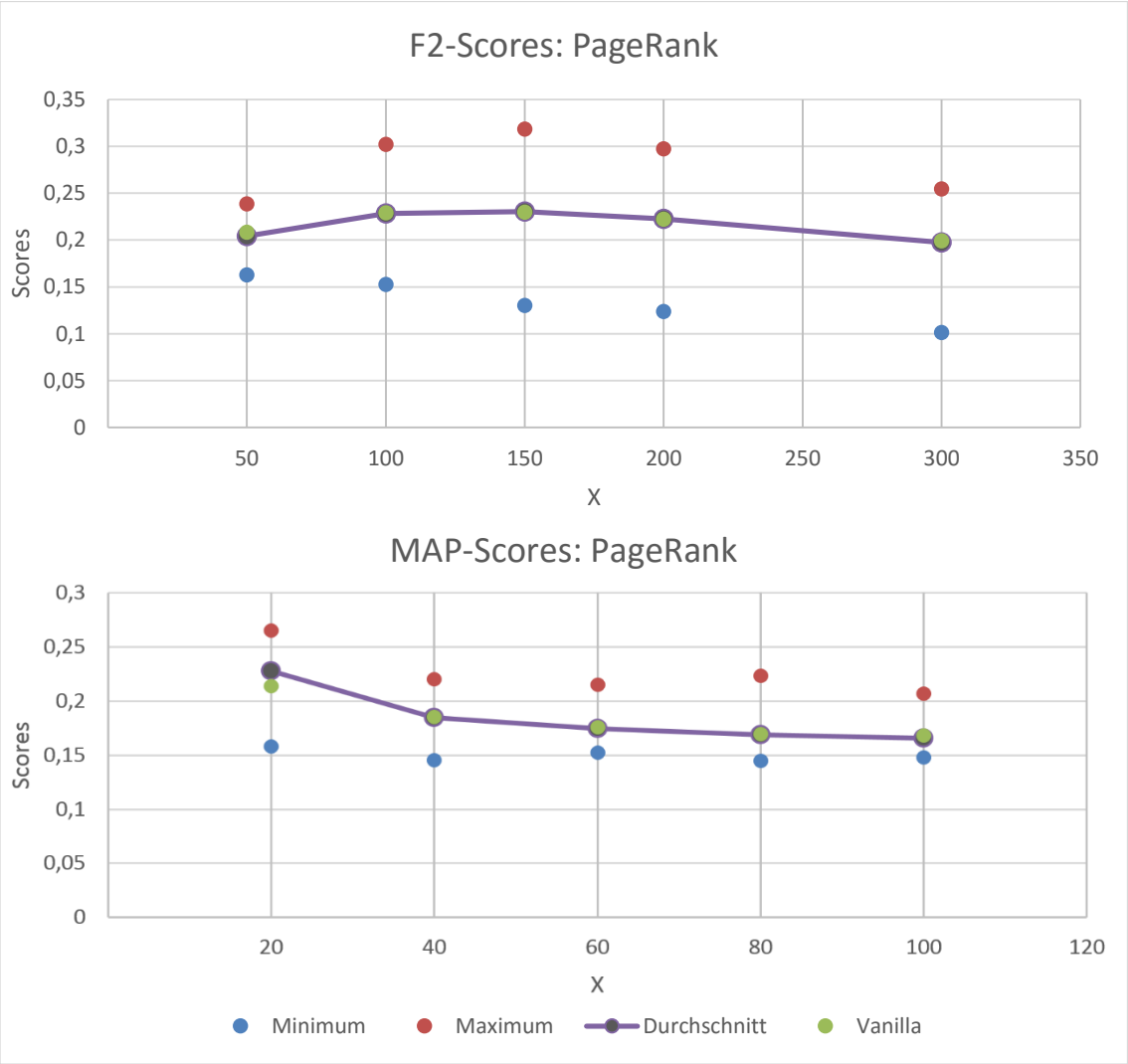


Abbildung 8: Ergebnisse der Evaluation der „PageRank"-Variante

Es wurden (mit Ausnahme der MAP@20) praktisch dieselben Werte erzielt wie ohne die Verwendung des PageRank. Im Machine Learning-Verfahren ist nur eine geringfügige Verschlechterung durch einen unpassenden Faktor möglich, weil für diesen dann eine sehr kleine Gewichtung bestimmt wird. Dies ist hier für die meisten Werte von X eingetreten.

Mit diesen Ergebnissen müsste man sich dagegen aussprechen, dass der PageRank als Rankingfaktor in PubMed verwendet wird. Dies bringt die Frage auf, wieso in der Arbeit von Bernstam, et al. (2006), die als Inspiration für die Verwendung des PageRank in dieser Arbeit diente, damit eine signifikante Verbesserung erzielt wurde und eine Empfehlung für die Verwendung gegeben wurde.

Es sollte angemerkt werden, dass die Suchmaschinen in der anderen Arbeit einem Test unterzogen wurden, bei dem eine sehr kleine Menge an Artikeln (457), die von der

Society of Surgical Oncology als wichtig bestimmt wurden, als relevant eingestuft wurde. Es gibt jedoch keine Garantie dafür, dass andere Artikel nicht auch für die getesteten Querys relevant gewesen wären (Bernstam, et al., 2006). Bei der Suche nach „den relevantesten" Artikeln zu einem Thema macht es zumindest Sinn, dass der PageRank oder ähnliche Bibliometrics eine Verbesserung bringen; mit den in dieser Arbeit festgelegten Qualitätsmerkmalen für die wissenschaftliche Suche muss dies nicht mehr zutreffen.

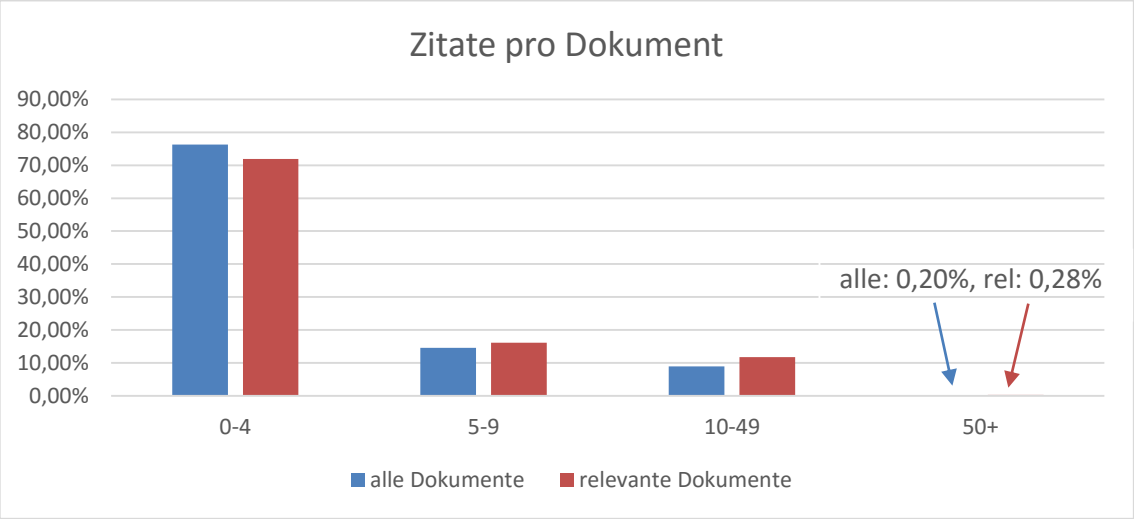


Abbildung 9: Häufigkeiten, mit denen Dokumente zitiert werden

In dieser Grafik werden Dokumente in vier Kategorien eingeordnet, je nachdem, wie häufig sie von anderen Dokumenten aus der Kollektion zitiert werden. Die Anzahl der Zitate steht auf der X-Achse. Die Höhe der Balken gibt an, wie viel Prozent der Dokumente aus den Gruppen (der gesamten Kollektion bzw. den relevanten Dokumenten für die Topics) in die Kategorien fallen.

Die grundsätzliche Tendenz, dass Dokumente, die für mindestens eins der Topics relevant sind, häufiger zitiert werden als die anderen spricht eigentlich dafür, dass der PageRank eine mögliche Verbesserung darstellt, wobei die Zahl der Zitate nicht dem PageRank-Wert eines Dokuments entspricht.

Betrachtet man jedoch die absoluten Zahlen der Dokumente in den Kategorien, so muss man feststellen, dass der Unterschied nicht ausgeprägt genug ist. Selbst die Zahl der Dokumente in der dritten Kategorie (10 bis 49) ist noch fast doppelt so groß wie die der relevanten Dokumente in allen Kategorien. Insofern ist es nicht überraschend, dass hier keine Verbesserung eintritt, und dies steht nicht im Widerspruch zu den Ergebnissen von (Bernstam, et al., 2006).

5.2.4 Ontologien

Die Systemvariante „Ontologien“ erzielte folgende Werte:

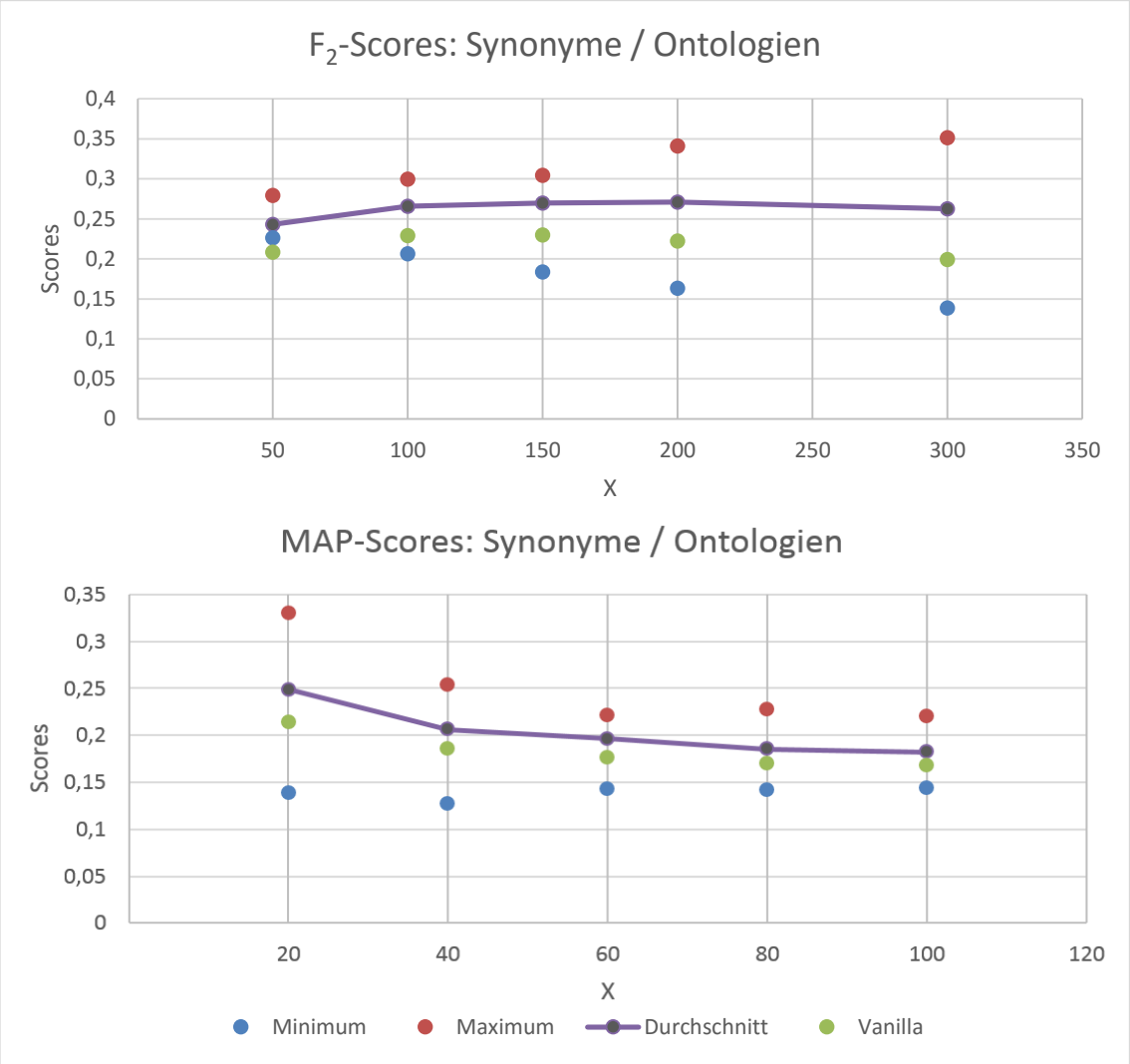


Abbildung 10: Ergebnisse der Evaluation der „Ontologien“-Variante

Für die F_2-Werte befinden sich Verbesserungen der Werte des Scores im Bereich von 0,04 bis 0,06, was eine deutliche Erhöhung darstellt. Die Werte steigen außerdem länger an, d.h. hier werden auch in höheren Rängen mehr relevante Dokumente gefunden als bei der Vanilla-Variante. Beim Maximum für X = 300 beträgt der Recall 0,48 bei einer Precision von 0,17.

Die MAP-Werte sind ebenfalls besser als bei der Vanilla-Variante, wobei die Steigerung (mit Ausnahme von X = 20) nicht so groß ist. Da sie für alle X-Werte vorhanden ist, kann man jedoch annehmen, dass es sich nicht um eine zufällige Verbesserung handelt.

Die Suche nach Synonymen bringt folglich eine Verbesserung der Ergebnisse und ist zu empfehlen.

5.2.5 Kombinierter Ansatz

Es sollte hier zunächst angemerkt werden, dass die Scores nur für Dokumente gewertet wurden, die bereits ohne die Verwendung des Ansatzes gefunden wurden. Die Alternative dazu wäre, alle Dokumente mit in die Ergebnismenge aufzunehmen, die jeweils Werte für den kombinierten Rank für einen relevanten Term beinhalten. In Punkt 3.3 wurde als möglicher Vorteil des kombinierten Scores erwähnt, dass er den Recall erhöhen könnte.

Das ist nicht eingetreten, stattdessen wurde die optimale Gewichtung des Ranks dann als so gering wie möglich bestimmt und trotzdem trat bei allen Varianten eine Verschlechterung der Ergebnisse um bis zu 0,03 auf.

5.2.5.1 Ergebnisse

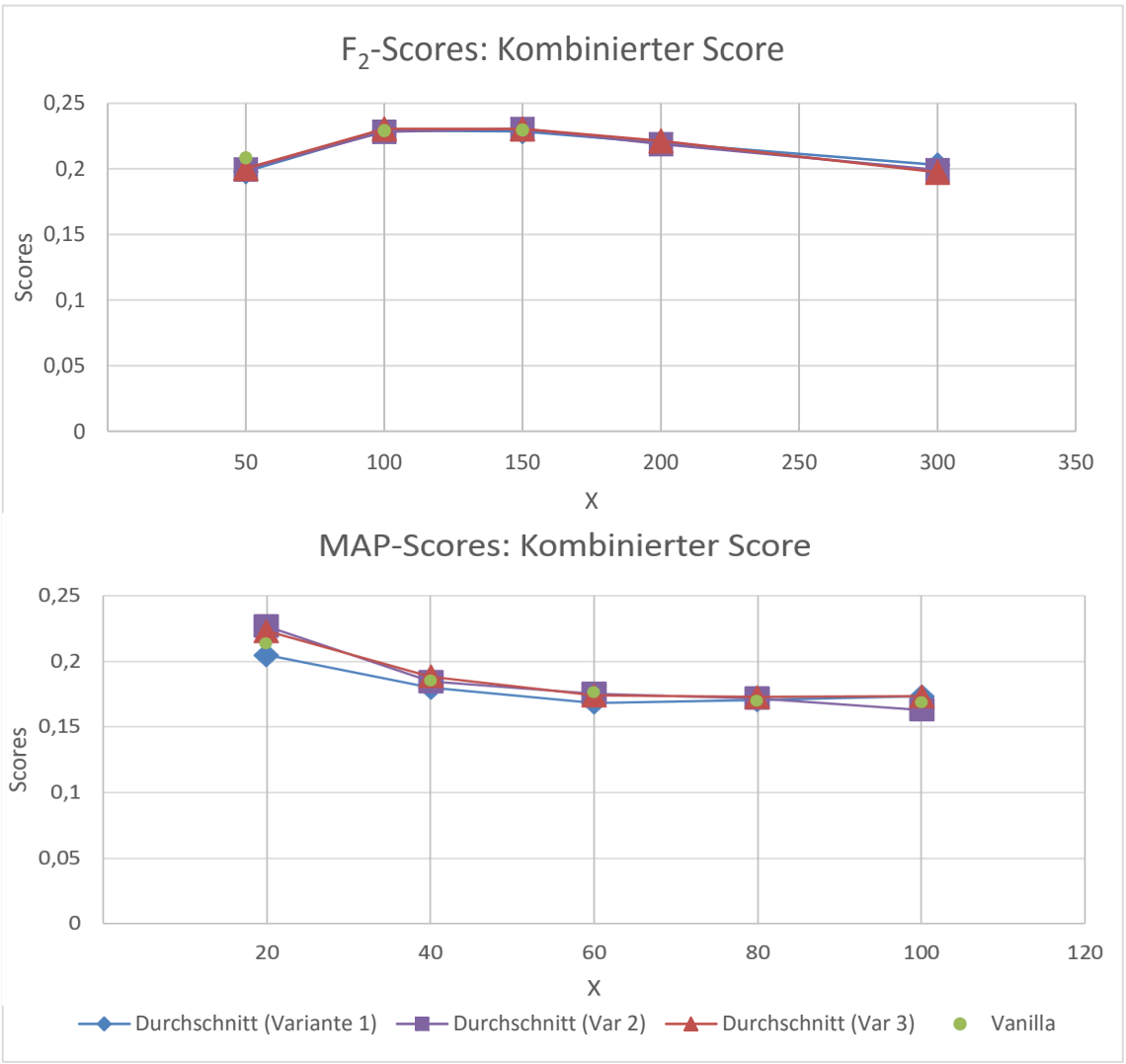


Abbildung 11: Ergebnisse der Evaluation der Systemvariante „Kombinierter Score"

Der Übersicht halber wurden Minimum und Maximum in den Graphen weggelassen. Die Maxima sind vor allem für kleine X geringer als bei der „Vanilla"-Variante, die Minima dafür meist etwas höher. Die erzielten Werte für die drei Berechnungsvarianten des kombinierten Scores sind etwa gleich.

Von allen untersuchten Rankingfaktoren gibt es hier bei beiden Maßzahlen die geringsten Verbesserungen bzw. es ist keine Verbesserung feststellbar. Wie beim PageRank ist es auch hier so, dass die Ergebnisse für eine Multiplikation weitaus schlechter waren als für eine Addition und deswegen nicht aufgeführt werden.

5.2.5.2 Interpretation der Ergebnisse

Für eine Bestätigung und Begründung der Ergebnisse kann man die MeSH-Begriffe verwenden, die den Dokumenten der Kollektion zugeordnet sind. Es ist zwar nicht immer möglich, die richtigen Headings für die einzelnen TREC-Querys zu bestimmen (weil nicht alle der gesuchten Begriffe in MeSH vorhanden sind), aber diese Begriffe bieten trotzdem die Möglichkeit, die Ergebnisse für die einzelnen TREC-Queries zu optimieren. Es liegt die Vermutung nahe, dass aufgrund der allgemein geringen Precision der Suchergebnisse *d1* nicht genau genug bestimmt werden konnte. Diese Analyse sollte zeigen, ob die schlechte Performance des kombinierten Scores daran liegt oder ob der Gewinn, den man mit dem Score erreichen kann, allgemein niedrig ist.

Hierfür wurden zunächst alle Dokumente gesucht, die die relevanten Dokumente *R* zitieren; diese zitierenden Dokumente bilden die Menge *Z*. Die Menge *B* sind alle MeSH-Headings, die für *Z* indexiert sind. Da man schließlich nur mit den Begriffen in *B* die relevanten Dokumente mit dem kombinierten Ansatz boosten kann, werden diese weiter untersucht.

Für jeden Begriff in *B* wird dann die Menge der Dokumente bestimmt, die das Heading als Deskriptor haben (nach obiger Benennung: *d1)*. *Z* ist eine Teilmenge von *d1*. *R* ist also auch eine Teilmenge der Dokumente, die von *d1* zitiert werden (*d2*). Die Frage ist nun, wie viel größer die Menge *d2* im Vergleich zu *R* ist und ob die Dokumente in *R* besonders häufig von Dokumenten aus *d1* zitiert werden; das würde dafür sprechen, dass der kombinierte Ansatz so funktionieren kann.

Aus Performancegründen wurde das nicht für alle Headings in *B* untersucht, sondern nur für diejenigen, die in mindestens fünf der Dokumente aus Z vorkommen.

Zur Veranschaulichung der Ergebnisse dient die nachfolgende Grafik:

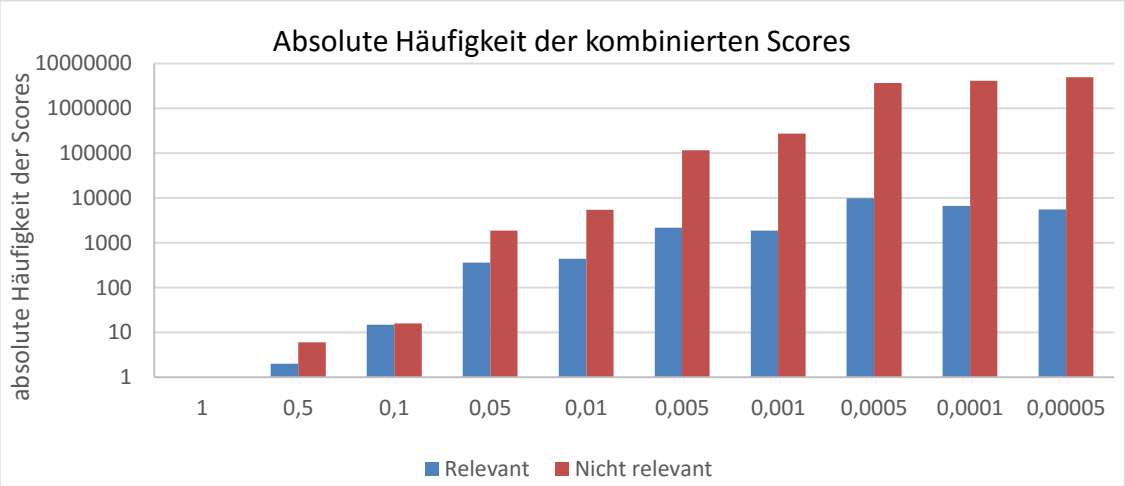


Abbildung 12: Absolute Häufigkeit der kombinierten Scores

Der Score (x-Achse) gibt die Werte für die erste Berechnungsvariante des kombinierten Ansatzes an. Die höhe der Baken gibt dabei an, wie viele Dokumente den Wert haben, der beim Balken steht, oder einen Wert bis zu dem des Balkens rechts davon (bei 0,00005: 0 < Wert <= 0,00005, bei 0,0001: 0,00005 < Wert <= 0,0001 usw.).

Wenn man die Werte in der Grafik normalisiert (also die Balken für die relevanten Dokumente durch die Gesamtzahl der relevanten Dokumente teilt und dasselbe mit den irrelevanten Dokumenten macht), sieht man, dass grundsätzlich eine wünschenswerte Tendenz vorhanden wäre:

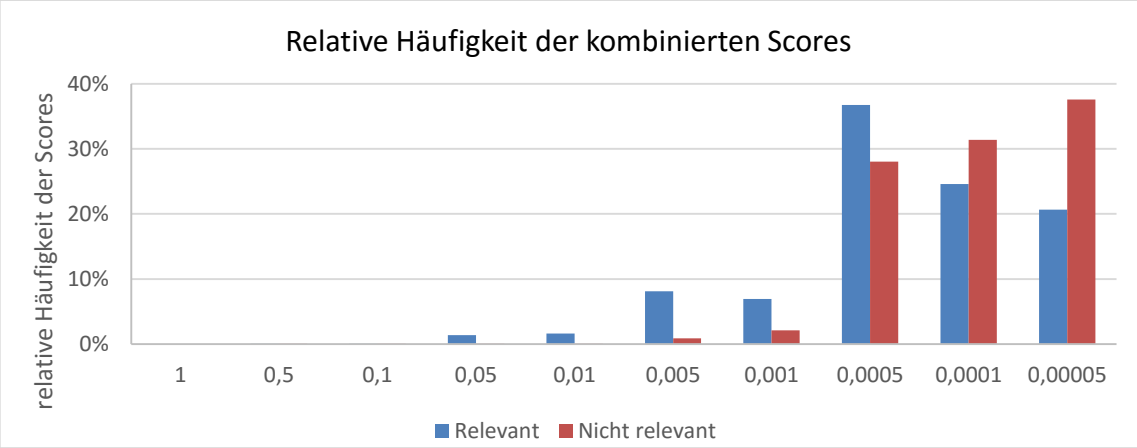


Abbildung 13: Relative Häufigkeit der kombinierten Scores

Die relevanten Dokumente erhalten also im Schnitt bessere Scores als die nichtrelevanten.

Es liegt trotzdem die Vermutung nahe, dass durch die absolut höhere Zahl der nichtrelevanten Dokumente viel *Noise* entsteht, der die Möglichkeiten des Scores stark einschränkt. Diese beiden Diagramme zeigen einen Überblick über alle MeSH-Headings; es sollte jedoch versucht werden, die besten zu finden.

„Die besten" werden hier definiert als MeSH-Headings, für die für Scores größer als ein Grenzwert S absolut mehr relevante als nichtrelevante Dokumente gefunden werden. Eine andere Definition ist vorerst nicht notwendig; dies soll überblicksmäßig dabei helfen, herauszufinden, ob es genügend solcher Begriffe gibt, um den Ansatz theoretisch weiter verfolgen zu können.

In Abhängigkeit von S gibt es dafür verschiedene Ergebnisse. Je nachdem, wie sehr man den kombinierten Ansatz später in den Score des Dokument mit einfließen lassen würde, wären unterschiedliche Werte für S in dieser Analyse angemessen. Wenn der Einfluss bzw. der Wert von S höher ist, dann können sich geringere Werte bereits stark auswirken.

Für S = 0,0005 gibt es für genau eine der TREC-Querys fünf MeSH-Headings, für die die Bedingung erfüllt ist, für alle anderen gibt es keines. Für S = 0,005 gibt es bei zwei anderen Querys solche Headings, insgesamt sind es hier zehn. Für Werte von S um 0,01 scheint das Maximum zu sein, hier gibt es 16 passende Headings in vier Querys.

Obwohl diese Analyse mit vielen Vereinfachungen gemacht wurde und nur die Berechnungsvariante 1 des Scores betrachtet wurde, zeigt dies doch klar, dass die Möglichkeiten des Ansatzes beschränkt sind. Die Begründung der Ergebnisse ist also, dass bei praktisch jedem Thema zu viele nichtrelevante Dokumente die Scores erhalten und es kaum Begriffe gibt, bei denen absolut mehr relevante Dokumente hohe Scores erhalten als Nichtrelevante.

5.2.6 Übersicht & Kombination von Rankingfaktoren

Rankingfaktoren				**F_2@100**	**MAP@20**
Vanilla (Vergleichswerte)				0,229	0,214
Feldtyp				0,275	0,304
	Ontologien			0,266	0,249
		PR		0,228	0,228
			KS	0,228	0,227
Feldtyp +	**Ontologien**			**0,287**	**0,323**
Feldtyp +		PR		0,273	0,295
Feldtyp +			KS	0,270	0,300
	Ontologien +	PR		0,270	0,223
	Ontologien +		KS	0,280	0,231
		PR +	KS	0,230	0,232

Abbildung 14: Übersicht der Ergebnisse der Faktoren (einzeln und in Kombination)

- Feldtyp: „Language Models“ als Rankingalgorithmus und „Worddelimiter“ als Indexierungsvariante (in den Zeilen, in denen dies nicht verwendet wird, werden der TF*IDF und die Indexierungsvariante „Simple“ verwendet.)
- PR = *PageRank*
- KS = kombinierter Score (Berechnungsvariante 2)

Die linke Spalte der Tabelle gibt an, welche der Einflussfaktoren für das Ranking benutzt werden. Rechts stehen die Ergebnisse für die beiden verwendeten Maßzahlen.

Varianten mit drei oder mehr Faktoren werden nicht aufgelistet, da jeweils eine Verschlechterung im Vergleich zu den Varianten besteht, die die Feldtypen und / oder Ontologien benutzen.

Mit der Kombination der beiden Rankingfaktoren, die individuell am besten abgeschnitten haben, erhält man eine weitere Verbesserung um knapp 0,02 bei der MAP@20 und um etwas mehr als 0,01 für den F_2@100-Score (im Vergleich zu den Ergebnissen, die mit der Wahl der Feldtypen erreicht wurden). Man kann insgesamt eine Empfehlung für die Verwendung der Rankingfaktoren „Feldtyp“ und „Ontologien“ in Kombination aussprechen.

6 Interpretation der Ergebnisse und Ausblick

Es lohnt sich wohl nicht, den vorgeschlagenen kombinierten Ansatz weiter zu verfolgen. Mit den vielen Metadaten, die man dafür benötigt, kann man auf andere Art und Weise (z.B. über die Erweiterung der Suche durch MeSH-Begriffe) bessere Ergebnisse erzielen.

Der zweite Kernpunkt der Arbeit neben dem kombinierten Score war die Untersuchung der möglichen Verbesserung der Retrievalergebnisse durch die Kombination von Rankingfaktoren. Diese Analyse hat ein positiveres Ergebnis: Die Kombination führt in vielen Fällen zu einer weiteren systemzentrierten Verbesserung der Suchmaschine. Da das aber nicht in allen Fällen so ist, kann man keine Empfehlung für die Kombination aller möglichen Faktoren aussprechen, sondern es bedarf jeweils einer Evaluation des Systems mit allen Faktoren.

Eine wichtige Erkenntnis ist, dass viel von der Wahl von passenden Indexierungsmethoden und Rankingalgorithmen abhängt. Ursprünglich sollte die Verbesserung durch die Wahl der optimalen Feldtypen als Referenzwert für die anderen untersuchten Faktoren dienen, doch nur die Verwendung von Ontologien kam in die Nähe der Verbesserung, die durch die optimalen Feldtypen möglich ist.

Die Anforderungen an die Systeme, die im Punkt 5.1 und insbesondere 5.1.3 festgelegt wurden, können nur von den Systemvarianten teilweise erfüllt werden, die die optimalen Feldtypen verwenden. Für oberflächliche Suchanfragen sind die Treffer vermutlich präzise genug.

Allgemein lässt sich festhalten, dass es sich lohnt, diverse Rankingfaktoren und -methoden zu untersuchen. Eine weitere Befassung mit dem kombinierten Score in der Zukunft wird als nicht sinnvoll erachtet. Für künftige Forschung gibt es die zwei Erkenntnisse, dass man neue Faktoren vor allem auch in Kombination untersuchen sollte und dass man die grundlegende Konfiguration der Suchmaschine (wie die gewählten Feldtypen) optimal bestimmen sollte.

Da in Punkt 4.2.5 einige Punkte beschrieben wurde, die nicht als Rankingfaktoren miteinbezogen wurden, würde es sich anbieten, diese als nächsten Schritt als weitere Faktoren zu verwenden und zu untersuchen, wie man damit die Suchergebnisse verbessern kann. Dafür müsste das entwickelte Learning to Rank-Verfahren weiterentwickelt

werden. Man könnte auch nach Methoden suchen, den Recall zu erhöhen, da für ausführliche Suchvorgänge noch nicht genug relevante Dokumente gefunden werden.

Literaturverzeichnis

Apache Software Foundation. (20. August 2015). *Tokenizers.* Von Confluence - Apache Software Foundation: https://cwiki.apache.org/confluence/display/solr/Tokenizers abgerufen

Bernstam, E., Herskovic, J., Aphinyanaphongs, Y., Aliferis, C., Sriram, M., & Hersh, W. (2006). Using Citation Data to Improve Retrieval from MEDLINE. *Journal of the American Medical Informatics Association: JAMIA*, S. 96-105.

Canese, K. (22. Oktober 2013). *PubMed Relevance Sort*. Von U.S. National Library of Medicine: http://www.nlm.nih.gov/pubs/techbull/so13/so13_pm_relevance.html abgerufen

Demner-Fushman, D., & Lin, J. (2006). Answer Extraction, Semantic Clustering, and Extractive Summarization for Clinical Question Answering. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL* (S. 841-848). Association for Computational Linguistics.

Dogan, R., Murray, G. C., Névéol, A., & Lu, Z. (2009). *Understanding PubMed® user search behavior through log analysis.* Oxford: Database.

Doms, A., & Schroeder, M. (2005). GoPubMed: exploring PubMed with the gene ontology. *Nucleic acids research*, W783-W786.

Eaton, A. D. (2006). HubMed: a web-based biomedical literature search interface. *Nucleic acids research*, W745-W747.

Garfield, E. (21. August 1995). New International Professional Society Signals The Maturing Of Scientometrics And Informetrics. *The Scientist*, S. 11.

Gruber, T. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, S. 907-928.

Harman, D. (1993). Overview of the first TREC conference. *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '93)* (S. 36-47). New York: ACM.

Hemminger, B. M., Lu, D., Vaughan, K. T., & Adams, S. (2007). Information seeking behavior of academic scientists. *Journal of the American Society for Information Science and Technology*, S. 2205-2225.

Hersh, W. R., Cohen, A. M., & Roberts, P. M. (2006). *TREC 2006 genomics track overview.* TREC.

Hersh, W., Cohen, A. M., Roberts, P., & Rekapalli, H. K. (2006). *TREC 2006 Genomics Track Overview.* Von Oregon Health & Science University: http://skynet.ohsu.edu/trec-gen/trec-06-genomics.pdf abgerufen

Korjonen-Close, H. (2005). The information needs and behaviour of clinical researchers: a user-needs analysis. *Health Information & Libraries Journal*, S. 96-106.

Koster, J. (2014). *PubReMiner*. Von Academisch Medisch Centrum, Department of Oncogenomics: http://hgserver2.amc.nl/cgi-bin/miner/miner2.cgi abgerufen

Li, H. (Oktober 2010). A Short Introduction to Learning to Rank. *IEICE Trans. Inf. & Syst.*

Lin, J., & Wilbur, W. J. (2007). PubMed related articles: a probabilistic topic-based model for content similarity. *BMC Bioninformatics*, S. 423.

Lu, Z., Kim, W., & Wilbur, J. W. (2009). Evaluating relevance ranking strategies for MEDLINE retrieval. *Journal of the American Medical Informatics Association*, S. 32-36.

Manning, C. D., Raghavan, P., & Schütze, H. (2009). *Introduction to information retrieval.* Cambridge: Cambridge university press.

McFee, B., & Lanckriet, G. R. (2010). Metric Learning to Rank. *Proceedings of the 27th International Conference on Machine Learning* (S. 775-782). ICML.

Muin, M., Fontelo, P., Liu, F., & Ackerman, M. (2005). SLIM: an alternative Web interface for MEDLINE/PubMed searches – a preliminary study. *BMC medical informatics and decision making*, S. 37.

National Institue of Standards and Technology. (10. April 2014). *Tracks*. Von Text REtrieval Conference - National Institue of Standards and Technology: http://trec.nist.gov/tracks.html abgerufen

Page, L., Brin, S., Motwani, R., & Winograd, T. (29. Januar 1998). The PageRank citation ranking: bringing order to the Web. Stanford.

Rebholz-Schuhmann, D., Kirsch, H., Arregui, M., Gaudan, S., Riethoven, M., & Stoehr, P. (2007). EBIMed - text crunching to gather facts for proteins from Medline. *Bioinformatics*, S. e237-e244.

Roberts, P., Cohen, A., & Hersh, W. (6. April 2015). *TREC Genomics Track.* Von TREC Genomics Track: http://skynet.ohsu.edu/trec-gen/ abgerufen

Salton, G., Wong, A., & Yang, C. S. (November 1975). A vector space model for automatic indexing. *Communications of the ACM*, S. 613-620.

Sarkar, I. N., Schenk, R., Miller, H., & Norton, C. N. (2009). LigerCat: Using "MeSH Clouds" from Journal, Article, or Gene Citations to Facilitate the Identification of Relevant Biomedical Literature. *AMIA Annual Symposium Proceedings*, (S. 563-567).

Silverstein, C., Marais, H., Henzinger, M., & Moricz, M. (September 1999). Analysis of a very large web search engine query log. *SIGIR Forum*, S. 6-12.

Smalheiser, N. R., Zhou, W., & Torvik, V. I. (2008). Anne O'Tate: A tool to support user-driven summarization, drill-down and browsing of PubMed search results. *Journal of Biomedical Discovery and Collaboration*, 2.

Sullivan, D. (23. September 2013). *FAQ: All About The New Google "Hummingbird" Algorithm*. Von Search Engine Land: http://searchengineland.com/google-hummingbird-172816 abgerufen

U.S. National Library of Medicine. (1998). *Alzheimer Disease - MeSH - NCBI*. Abgerufen am 12. September 2015 von MeSH - NCBI: http://www.ncbi.nlm.nih.gov/mesh/68000544

U.S. National Library of Medicine. (17. Dezember 2013). *2014 MEDLINE®/PubMed® Baseline Database Distribution*. Von U.S. National Library of Medicine: http://www.nlm.nih.gov/bsd/licensee/2014_stats/baseline_med_filecount.html ("2014" in URL durch Werte von 2009 bis 2014 ersetzen) abgerufen

U.S. National Library of Medicine. (17. December 2014). *Key MEDLINE Indicators*. Von National Library of Medicine: http://www.nlm.nih.gov/bsd/bsd_key.html abgerufen

U.S. National Library of Medicine. (25. Juni 2015). *Fact SheetMEDLINE, PubMed, and PMC (PubMed Central): How are they different?* Von U.S. National Library of Medicine: http://www.nlm.nih.gov/pubs/factsheets/dif_med_pub.html abgerufen

U.S. National Library of Medicine. (2015). *Genes, p53 - MeSH*. Von MeSH: http://www.ncbi.nlm.nih.gov/mesh/?term=gene%2C+p53 abgerufen

U.S. National Library of Medicine. (20. April 2015). *Home - PMC*. Von U.S. National Library of Medicine - PubMed Central: http://www.ncbi.nlm.nih.gov/pmc/ abgerufen

U.S. National Library of Medicine. (kein Datum). *Genes, p53 - MeSH - NCBI*. Abgerufen am 11. September 2015 von MeSH - NCBI: http://www.ncbi.nlm.nih.gov/mesh/?term=genes%2C+p53

Uman, L. (Februar 2011). Systematic Reviews and Meta-Analyses. *Journal of the Canadian Academy of Child and Adolescent Psychiatry*, S. 57-59.

Weizmann Institute of Science. (2015). *GeneCards*. Von GeneCards: www.genecards.org abgerufen

Wikipedia Contributors. (4. September 2015). *PageRank.* Von Wikipedia, The Free Encyclopedia: https://en.wikipedia.org/w/index.php?title=PageRank&oldid=679393024 abgerufen

Yue, Y., Finley, T., Radlinski, F., & Joachims, T. (2007). A support vector method for optimizing average precision. *SIGIR 07* (S. 271-278). ACM.

Zhai, C., & Lafferty, J. (2001). A study of smoothing methods for language models applied to Ad Hoc information retrieval. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '01)* (S. 334-342). New York: ACM.

Zhiyong, L., Kim, W., & Wilbur, W. J. (2009). Evaluation of query expansion using MeSH in PubMed. *Information Retrieval*, S. 69-80.

Alle Abbildungen wurden selbst erstellt.

Das letzte Aufrufdatum für alle URLs ist der 14.09.2015.

Plagiatserklärung

Ich habe die Arbeit selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und bisher keiner anderen Prüfungsbehörde vorgelegt.

_____ _____

Ort, Datum Unterschrift

Anhang

A1: USB-Stick

Alle Anlagen befinden sich auf einem USB-Stick, der mit der Arbeit abgegeben wurde. Darauf sind auch der Code und der Solr-Server gespeichert, die für die Arbeit entwickelt wurden. Außerdem befinden sich der Code und die Dokumente im GitHub-Repository https://github.com/JonBrem/PubMed-Bachelorarbeit, die Anlagen sind hier im Ordner „docs“.

Anlage A2 wurde mit einem Tabellenkalkulationsprogramm erstellt und ist damit vermutlich angenehmer zu lesen.

A2: Überblick von Informationssystemen und Forschungsarbeiten zu PubMed

(Links zu den Suchmaschinen und Forschungsarbeiten sind in der digitalen Version enthalten)

Name	Autor(en) / Entwickler	Datengrundlage	Goldstandard	Rankingalgorithmus / Systemvarianten	Nutzung von Systemen wie Solr	Was ist besonders?
Anne O'Tate	Neil R Smalheiser, Wei Zhou and Vetle I Torvik	PubMed (max. 25 000 Dokumente pro Suche)		TF / IDF wird für Keywords berechnet	PubMed API; NIH MetaMap (Semantik)	Fokus liegt auf "More like this" in mehreren Dimensionen (Autoren, Keywords, aber auch 'alle aktuellen Treffer')
BibliMed	Basset, H	PubMed (25 Mio. Dokumente); z.T. noch nicht indexiert			PubMed API + MeSH	Links zu Wikipedia, Amazon, ...
BITOLA	Dimitar Hristovski & Borut Peterlin	PubMed, MeSH, UMLS	Vorhersagen von Zusammenhängen, die bestätigt wurden		PubMed, Datenbank für Assoziationen	Automatische Verknüpfung von Ursachen / Heilmitteln mit Krankheiten
BioIE	A Divoli, T K Attwood	PubMed, eigene Uploads (txt)	Set an fest definierten Regeln für Sätze		PubMed, eigene Uploads	Extrahiert wichtige Sätze aus Abstracts
EBIMed	D Rebholz-Schuhmann, H Kirsch, M Arregui, S Gaudan, M Riethoven, P Stoehr	Medline, Protein- und Gen-datenbanken	Ein Gen, das häufig untersucht wird		Lucene	Co-occurence von verschiedenen Genen in einem Satz --> Zusammenhänge
Evidentista	D Richards	Zahnarzt-sachen, nirgends steht was von Medline				Speziell für Zahnärzte
eTBLAST	J Lewis, S Ossowski, J Hicks, M Errami, H R Garner	Medline (und andere Datenbanken)	Usertests, TREC	unterschiedliche Retrieval- und Rankingalgorithmen	"We implemented [...]", auf eTBLAST	Mehr eine Machbarkeitsstudie (ob man mit eTBLAST PubMed indexieren kann)
Go3R	U G Sauer, T Wächter, B Grune, A Doms, M R Alvers, H Spielmann, M Schroeder		manueller Test, keine Maßzahlen angegeben	Maximum Enthropy Method	Basiert auf GoPubMed; von einer Uni-Firma weiterentwickelt	
GoPubMed	A Doms, M Schroeder	Medline, www.geneontology.org		Terme in GO-Datenbank aus Abstracts werden indexiert	Selbstgeschrieben	Gene Ontology als Hilfsmittel
FACTA+	Y Tsuruoka, J Tsujii, S Ananiadou	Medline		pointwise mutual information (vom Konzept her ähnlich wie TF / IDF)	Selbstgeschrieben	Rankingalgorithmus für PubMed-Dokumente
HubMed	A Eaton	PubMed & "selbst" indexiert: Medline		Auswahl: Relevance oder Date	Auswahl: Lucene oder PubMed (+ MeSH)	Auswahl, nur 1 Eingabefeld; UI wie Websuchmaschinen

Fortsetzung auf den folgenden Seiten

Name	Autor(en) / Entwickler	Datengrundlage	Goldstandard	Rankingalgorithmus / Systemvarianten	Nutzung von Systemen wie Solr	Was ist besonders?
LigerCat	I N Sarkar, R Schenk, H Miller, C N Norton	PubMed, lokal und im Web		generiert MeSH-Queries für PubMed	Selbstgeschrieben; gibt selber keine Ergebnisse zurück	Nur für Gene
MedlineTrend	?	?	?	?	?	Zeigt an, wie sich PubMed verändert
MEDSUM	?	?	?	?	PubMed	fasst Ergebnisse zusammen, z.B. "wie viel hat Person X veröffentlicht"
MiSearch		Medline	cross-validation	Relevance Feedback, Terme: probabilistisch, mehrere Faktoren (z.B. Veröffentlichungstermin)	Datenbank, PubMed e-Utils (aber zum Großteil wohl handgeschrieben)	Relevance Feedback & Ranking
POPLINE						Alternatives / spezialisierteres System als PubMed
PubAnamoty	W Xuan et al	PubMed		"lokale Suchmaschinen", kA	Selbstgeschrieben, PubMed e-Utils	Visualisierung
PubAtlas	D S Parker, W W Chu, F W Sabb, A W Toga, R M Bilder	PubMed			PubMed: Match von Ergebnissen von mehreren Termen --> Übereinstimmungen	Vergleich von Ergebnismengen
PubGet	?	?	?	?		kommerzielles Produkt; vermutl. Verbesserungen im Ranking
PubReMiner	J Koster	PubMed			PubMed, dann Weiterverarbeitung	Nicht die Treffer selber sind im Vordergrund, sondern Terme / Konzepte zu finden
PubNet	S M Douglas, G T Montelione, M Gerstein	PubMed			PubMed	Visualisierung, mehrere Queries auf eimal
Quertle	J Saffer, V Burnett	Medline + ähnliche Datenbanken			Selbstgeschrieben	Ranking
SLIM	M Muin, P Fontelo, F Liu, M Ackerman	PubMed	Beta-Tests mit Ärzten		Pubmed E-Utils	Slider-UI
XplorMed	C Perez-Iratxeta, P Boork, M A Andrade	PubMed	manuelle Berechnung der Precision, Vergleich mit PubMed	word chains	PubMed, Weiterverarbeitung mit eigener Sprachverarbeitungssoftware	Exploratory: Soll schnell anzeigen, was die "allgemeinen" Ergebnisse einer Query waren
RefMed	H Yu, T Kim, J Oh, S Kim	Medline		relevance Feedback	Eigene SQL-Implementierung	Relevance Feedback

Fortsetzung auf der folgenden Seite

Titel	Autor	Datengrundlage	Goldstandard	Nutzung von Systen	Mini-Beschreibung / Besonders
Answer Extraction, Semantic Clustering, and Extractive Summarization for Clinical Question Answering	D Demner-Fushman, J Lin	Medline	Auswertung durch Experten und anhand von Datenbanken	Selbstgeschrieben	Question-Answering-System
Evaluation of query expansion using MeSH in PubMed	Z Lu, W Kim, J Wilbur	ca. 160 000 Dokumente in Medline (Trec Genomics)	Trec Genomics (2006, 2007); Automatisierte Generierung von Fragen	Eigenes System, das Boolesches Retrieval und Probabilist. Retrieval unterstützt	ganz klassische Maßzahlen
PageRank without hyperlinks: Reranking with PubMed related article networks for biomedical text retrieval	J Lin	10 Jahre von Medline (Trec Genomics)	Trec Genomics (2005)	Terrier + eigener PageRank	PageRank innerhalb Ergebnissen einer Query
Evaluating Relevance Ranking Strategies for MEDLINE Retrieval	Z Lu, W Kim, J Wilbur	ca. 160 000 Dokumente in Medline (Trec Genomics)	Trec Genomics (2007)	Lucene	Vergleich von TF-IDF, Okapi BM25, ... mit klassischen Maßzahlen
Ranked retrieval of Computational Biology models	R Henkel, A Peters, N Le Novère, D Waltemath	Biomodels Database		Lucene	Nicht für medizinische Veröffentlichungen, wenn ich das richtig verstehe
How Do Users Find Things with PubMed? Towards Automatic Utility Evaluation with User Simulations	J Lin, M D Schmucker	ca. 160 000 Dokumente in Medline (Trec Genomics)	Trec Genomics (2005); Benutzer-Simulation als Testmöglichkeit		Nur Test, keine eigentliche Entwicklung.
PubMed related articles: a probabilistic topic-based model for content similarity	J Lin, W J Wilbur	ca. 160 000 Dokumente in Medline (Trec Genomics)	Trec Genomics (2005)	steht nichts da	Vergleich mehrerer Ranking-algorithmen für ähnliche Dokumente (u.a. BM25)
Modeling actions of PubMed users with n-gram language models	J Lin, W J Wilbur	8 Tage an LogDaten			Log-Studie zu Benutzerverhalten
Retrieving Clinical Evidence: A Comparison of PubMed and Google Scholar for Quick Clinical Searches	Shariff SZ, Bejaimal SA, Sontrop JM, Iansavichus AV, Haynes RB, Weir MA, Garg AX	PubMed / Google Scholar	Reviews von Experten		Google gewinnt
GeneView: a comprehensive semantic search engine for PubMed	S Shariff et al	PMC Open Access Subset; Spezielle Datenbanken (GNAT, LINNAEUS)		Lucene	Artikel, die bestimmte "biological entities" enthalten suchen

A3: TREC-Topics

Für jedes der Topics wird Information in folgendem Format aufgelistet:

[ID] [Frage]
[Gen(e), anderes Konzept]: [Synonyme, nach denen gesucht wurde]
[Zahl der relevanten Dokumente (0 = wurde bei der Evaluation nicht benutzt)]

160. What is the role of PrnP in mad cow disease?
PrnP: prion protein, g1-dependent
Mad cow disease: Bovine Spongiform Encephalopathy, BSE
214 relevante Dokumente

161. What is the role of IDE in Alzheimer's disease
IDE: Isulin Degrading Enzyme, Insulin Protease, Insulinase
Alzheimer's disease: Alzheimer Sclerosis, Senile Dementia
40 relevante Dokumente

162. What is the role of MMS2 in cancer?
MMS2: DDVit1, UEV-2, UBE2V2, ubiquitin-conjugating enzyme, EDPF-1
Cancer: neoplasm, tumor
1 relevantes Dokument

163. What is the role of APC (adenomatous polyposis coli) in colon cancer?
APC: adenomatours polyposis coli (in Query enthalten)
Colon Cancer: colon neoplasm, colon tumor
99 relevante Dokumente

164. What is the role of Nurr-77 in Parkinson's disease?
Nurr-77: Nur77, NGFI-B, Nerve Growth Factor-inducible B-Protein, Oprhan Nuclear Receptor HMR, Early Response Protein NAK1
Parkinson: Paralysis Agitans
4 relevante Dokumente

165. How do Cathepsin D (CTSD) and apolipoprotein E (ApoE) interactions contribute to Alzheimer's disease?
Cathepsin D (CTSD): keine Synonyme
Apoliprotein E (ApoE): Apo E Isoproteins, Aspartic Acid Endopeptidases
Alzheimer: s. 161
7 relevante Dokumente

166. What is the role of Transforming growth factor-beta1 (TGF-beta1) in cerebral amyloid angiopathy (CAA)?
(keine Synonyme; Akronyme werden als Synonyme gewertet)
2 relevante Dokumente

167. How does nucleoside diphosphate kinase (NM23) contribute to tumor progression?
nucleoside diphosphate kinase (NM23): Non-Metatastatic Cells 1 Protein, Granzyme A-activated DNase
tumor regression: neoplasm, cancer
158 relevante Dokumente

168. How does BARD1 regulate BRCA1 activity?
BARD1: Ubiquitin-Protein Ligases
BRCA1: keine Synonyme
56 relevante Dokumente

169. How does APC (adenomatous polyposis coli) protein affect actin assembly
APC: adenomatours polyposis coli (in Query enthalten)
Actin: isoacton
54 relevante Dokumente

170. How does COP2 contribute to CFTR export from the endoplasmic reticulum?
COP2: Coat Protein Complex II
CFTR: Cystic Fibrosis Transmembrane Conductance Regulator
28 relevante Dokumente

171. How does Nurr-77 delete T cells before they migrate to the spleen or lymph nodes and how does this impact autoimmunity?
Nurr-77: Nur77, NGFI-B, Nerve Growth Factor-inducible B-Protein, Oprhan Nuclear Receptor HMR, Early Response Protein NAK1
[zweites Konzept ungeeignet für die Angabe von Synonymen
14 relevante Dokumente

172. How does p53 affect apoptosis?
P53: phosphrotein 53, tumor suppressor gene
Apoptosis: programmed cell death, intrinsic pathway apoptosis
305 relevante Dokumente

173. How do alpha7 nicotinic receptor subunits affect ethanol metabolism?
(nicht benutzt)

174. How does BRCA1 ubiquitinating activity contribute to cancer?
BRCA1: ubiquitin-protein ligases
Cancer: s. 162
18 relevante Dokumente

175. How does L2 interact with L1 to form HPV11 viral capsids?
(nicht benutzt)

176. How does Sec61-mediated CFTR degradation contribute to cystic fibrosis?
(keine Synonyme)
4 relevante Dokumente

177. How do Bop-Pes interactions affect cell growth?
Pes: pescadillo
6 relevante Dokumente

178. How do interactions between insulin-like GFs and the insulin receptor affect skin biology?
(keine Synonyme)
3 relevante Dokumente

179. How do interactions between HNF4 and COUP-TF1 suppress liver function?
(keine Synonyme)
1 relevantes Dokument

180. How do Ret-GDNF interactions affect liver development?
(nicht benutzt)

181. How do mutations in the Huntingtin gene affect Huntington's disease?
Huntingtin gene: HAPP
Huntington's disease: Huntington Chorea
418 relevante Dokumente

182. How do mutations in Sonic Hedgehog genes affect developmental disorders?
(keine Synonyme)
94 relevante Dokumente

183. How do mutations in the NM23 gene affect tracheal development?
(nicht benutzt)

184. How do mutations in the Pes gene affect cell growth?
Pes: pescadillo
3 relevante Dokumente

185. How do mutations in the hypocretin receptor 2 gene affect narcolepsy?
Hypocretin receptor 2 gene: orexin receptor, HCRT receptor
Narcolepsy: Raoxysmal sleep, Gelineaus Syndrome
17 relevante Dokumente

186. How do mutations in the Presenilin-1 gene affect Alzheimer's disease?
Presenilin-1 gene: psen1
Alzheimer's disease: Senile Dementia, Alzheimer Sclerosis
281 relevante Dokumente

187. How do mutations in familial hemiplegic migraine type 1 (FHM1) gene affect calcium ion influx in hippocampal neurons?
Familial hemiplegic migraine type 1 (FHM1) gene: migraine with Auras
1 relevantes Dokument

A4: Erklärungen zu Solr und Programmen

Solr

Der Solr-Server befindet sich im Ordner *solr-5.1.0* auf dem mit abgegebenen USB-Stick. Man startet den Server, indem man in der Kommandozeile den Befehl „java –jar start.jar“ im Ordner solr-5.1.0/server/ ausführt.

AdvancedQuery: Bestimmung der Scores in allen Feldern

Die Klasse, mit der in dieser Arbeit die meisten Querys an Solr gesendet werden, ist die *AdvancedQuery*-Klasse. Auf ihre Funktionsweise wird hier im Detail eingegangen, weil diese bestimmt, wie die Scores berechnet werden und das hat einen Einfluss auf der Ergebnisse der Evaluation. Eine „normale“ Query mit der solrj-API (die eine Java-Schnittstelle zu Solr bietet) heißt *SolrQuery*.

Die *AdvancedQuery* bietet folgende Funktionalität:

1. Man kann für ein Feld mehrere Querys angeben; die damit jeweils erzielten Scores werden einzeln abgefragt aufsummiert.
2. Man kann mehrere zusätzliche Felder angeben, deren Werte abgefragt werden, ohne einen Einfluss auf den Score zu nehmen.
3. Es ist möglich, die Ergebnisse zu einem anderen Feldnamen zu speichern als den eigentlich abgefragten.

Beispiel:

Bei Topic 161 geht es um die Rolle des IDE-Gens in der Alzheimer-Erkrankung. Man erstellt dafür ein *AdvancedQuery*-Objekt und sagt, dass man die Teilquerys „ide“ und „alzheimer“ jeweils in folgenden Feldern durchführen möchte (Funktion 1):

language_worddelimiter_rank_lm_1_abstract

language_worddelimiter_rank_lm_1_title

language_worddelimiter_rank_lm_2_abstract

language_worddelimiter_rank_lm_2_title

Darüber hinaus möchte man für dieselben Felder die Werte für die Querys „insulin protease“ und „dementia“ erhalten, den Einfluss dieser Werte auf die Ergebnisse jedoch separat bestimmen können, indem man einen anderen Feldnamen angibt, unter dem das gespeichert wird (Funktion 3).

Außerdem will man den PageRank für jedes Dokument abfragen, das man findet (Funktion 2).

Die *AdvancedQuery* würde dann eine „normale“ Query für „ide“ im Feld „language_worddelimiter_rank_lm_1_abstract“ an Solr senden und für alle Ergebnisse den Score sowie den Wert für den PageRank speichern. Dann würde sie die Query „alzheimer“ im selben Feld durchführen und dasselbe machen; wenn sich ein Dokument bereits in den Ergebnissen befindet, dann wird der neue Score addiert.

Das Gleiche passiert nun mit den Querys „insulin protease“ und „dementia“, nur dass die Ergebnisse hierfür unabhängig von denen von „ide“ und „alzheimer“ zwischengespeichert werden.

Das wird für alle Felder wiederholt, die Ergebnisse der Felder sind unabhängig voneinander.

Am Ende könnte man die Ergebnisse anhand von bestimmten Regeln sortieren, indem man Gewichte für die verschiedenen Felder angibt, oder man lässt sich die Ergebnisse im .csv-Format ausgeben.

Wichtig ist insgesamt, dass keine Query nach „role of ide in alzheimer’s disease“ oder „role ide alzheimer“ oder „ide alzheimer“ durchgeführt wird, sondern zwei Querys: eine nach „ide“ und eine nach „alzheimer“. Die Scores werden dann addiert. Das kann andere Scores ergeben als eine Query nach beiden Termen auf einmal.

Erstellte Programme

Die entwickelten Java-Programme befinden sich auf dem USB-Stick, der mit abgegeben wurde, sowie auf dem GitHub-Repository https://github.com/JonBrem/PubMed-Bachelorarbeit. Der Code kann nicht ausgedruckt abgegeben werden, da etwa 6500 Zeilen Code erzeugt wurden. Es würde außerdem keinen Sinn machen, weil darüber hinaus die mehrere GB große Kollektion sowie der Solr-Server notwendig sind, um die Programme ausführen zu können.

Es gibt kein UI für die Programme. Es wurde auf Codequalität geachtet und es sind Kommentare zur Unterstützung vorhanden, jedoch wurde ebenfalls berücksichtigt, dass die Programme in den meisten Fällen etwas sehr Spezielles durchführen müssen.

Es deswegen nicht möglich, Parameter an die Programme zu übergeben, sondern die Konfiguration erfolgt in der *main*-Methode im Code der Dateien.

Um eine echte Suchmaschine oder Such-API zu entwickeln, wären nur die Dateien im Package *de.ur.jonbrem.pubmed.advanced_querying* sowie *de.ur.jonbrem.pubmed.solr-connection* notwendig. Der Vollständigkeit halber wird noch auf ein paar weitere Programme bzw. Klassen verwiesen, die zentrale Punkte der Arbeit darstellen:

- Package *de.ur.jonbrem.pubmed.indexing*:
  - „Indexer“ und „MeshIndexer“ erstellen die Kollektion, indem sie die XML-Dateien lesen und die Informationen an den Solr-Server schicken.
  - Im Subpackage „citations“: die Klasse „HighwireCitationFinder“ beinhaltet die Funktionalität, die Links innerhalb der Kollektion zu bestimmen.
  - Im Subpackage „pagerank“ werden mit dem Programm „CitationRank-Builder“ die PageRank-Scores der Dokumente berechnet.
- Package *de.ur.jonbrem.pubmed.test.machine_learning.own_learning*:
  - Das Programm „FastOptimization“ beinhaltet den Machine Learning-Algorithmus.
  - Die eigentliche Berechnung der Ergebnisse findet aber im „CalculatorThread“ statt.

A5: Beispiel für das *Learning to Rank*-Verfahren

Zunächst legt man Sets von Faktoren fest. Jeder Faktor eines Sets wird mit allen Kombinationen von Faktoren der anderen Sets getestet, nicht aber mit den anderen Faktoren seines Sets.

Zum Beispiel könnte es folgende drei Sets geben:

1) citation_rank
2) title_worddelimiter_bm25, title_stemming_bm25, ...
3) abstract_worddelimiter_bm25, abstract_stemming_bm25, ...

Dann wählt man sich eine Kombination aus, die man noch nicht getestet hat (was am Anfang alle sind). Die erste Kombination von Feldern wäre: citation_rank, title_worddelimiter_bm25, abstract_worddelimiter_bm25; bei der zweiten würde man anstatt abstract_worddelimiter_bm25 abstract_stemming_bm25 verwenden usw.

Angenommen, der Durchschnitt für das citation_rank Feld wäre 0,0001, für title_worddelimiter_bm25 0,3 und für abstract_worddelimiter_bm25 0,9. Die Felder sind bereits richtig geordnet, also den Durchschnittswerten der Dokumente nach aufsteigend.

Nun berechnet man die „Basis" für die Felder (bis auf das mit dem höchsten Durchschnitt), um die Potenzen von 3 zu bestimmen, die man als Gewichtungen ausprobiert. Die Basis für den citation_rank wäre 8, weil $\log_3 \frac{0.9}{0.0001}$ gerundet 8 ist. Die Gewichtungen für diesen Faktor wären also 3^3, 3^4 ... 3^{13}, die für den title_worddelimiter_bm25 wären 3^{-4}, ... 3^6.

Man berechnet nun für die erste Kombination von Gewichtungen (citation_rank: 3^3, title: 3^{-4}, abstract: 1) die Scores für das Retrievalmaß, das man zur Evaluation verwendet. Dann wiederholt man das für die nächste Kombination von Gewichtungen (citation_rank: 3^3,title: 3^{-3}, abstract: 1) und für alle anderen Kombinationen und notiert die Kombination mit dem höchsten Score.

Angenommen, die optimale Gewichtung lautet: 3^3 für den citation_rank und 1 (3^0) für den title. Dann würde man noch den Score für alle Kombinationen der Gewichtungen $3^{2.2}$ ... $3^{3.8}$ für den citation_rank und $3^{-0.8}$ ... $3^{0.8}$ für den title berechnen. Mit der Kombination mit dem höchsten Score berechnet man dann den Score für die Topics, die nicht zum Training verwendet wurden und notiert diesen.