Bauhaus-Universität Weimar
Fakultät Medien
Studiengang Mediensysteme

Verbesserte Kaskade zur Suchsitzungs- und Missionserkennung in Suchanfrage-Logs

Bachelorarbeit

Jakob Gomoll
Geboren am 13.02.1988 in Weimar

Matrikelnummer 70003

1. Gutachter: Prof. Dr. Benno Stein
2. Gutachter: Prof. Dr. Sven Bertel
Betreuer: Dr. Matthias Hagen

Datum der Abgabe: 28. März 2012

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Weimar, den 28. März 2012

.....
Jakob Gomoll

Zusammenfassung

Diese Arbeit befasst sich mit verschiedenen Methoden zur automatischen Suchsitzungs- und Missionserkennung in Anfrage-Logs von Websuchmaschinen. Bei der Sitzungserkennung werden aufeinanderfolgende Suchanfragen, die demselben Informationsbedarf dienen, zu Suchsitzungen zusammengefasst. Die anschließende Missionserkennung prüft diese Sitzungen auf thematische Ähnlichkeit und fasst diese bei Zusammengehörigkeit zu übergeordneten Missionen zusammen (beispielsweise können das Buchen eines Hotels und das Auflisten von Sehenswürdigkeiten Teile der Mission *Reiseplanung* sein).

Das beste uns bekannte Verfahren zur Suchsitzungserkennung stellten Hagen et al. im Jahr 2011 vor, welches verschiedene Schritte zur automatischen Sitzungserkennung in einer Kaskade anordnet, sodass zeitaufwändige Schritte nur dann ausgeführt werden, wenn effizientere Methoden keine sichere Entscheidung treffen können [HSR11]. Wir untersuchen diese bereits existierende Kaskade und nehmen einige Verbesserungen an ihr vor, welche gleichermaßen in einer schnelleren Laufzeit und einer etwas besseren Güte resultieren (die ursprüngliche Kaskade erzielt ein F-Measure von 0,8812, welches wir auf 0,8940 steigern können).

Zusätzlich entwickeln wir auf Basis von miteinander verknüpften Informationen im Web (*Linked Open Data*) einen weiteren Schritt für die Kaskade, der in der Lage ist, Zusammenhänge zwischen zwei Anfragen ohne jegliche syntaktische Ähnlichkeit auf semantischer Ebene zu erkennen. Wir zeigen anhand einer eigens erstellten Liste mit Anfragepaaren, dass diese Technik fast viermal soviele Zusammenhänge erkennen kann, wie die bisher eingesetzte Technik zur Bestimmung der semantischen Ähnlichkeit (die Explizite Semantische Analyse), wenn die Anfragen denn präzise genug formuliert sind. Für die Bearbeitung realer Suchanfragen jedoch trifft diese Technik zu viele Fehlentscheidungen, weshalb wir diesen Schritt nicht in die Kaskade aufnehmen.

Darüber hinaus entwerfen wir in dieser Arbeit ein Evaluationsmodell für die Ergebnisse der Missionserkennung und zeigen, dass die Kaskade zur Suchsitzungserkennung unter Anpassung einiger Parameter auch zur automatischen Missionserkennung verwendet werden kann und dabei gute Ergebnisse erzielt.

Da die uns vorliegenden Gold-Standards zur Evaluation der Ergebnisse zahlreiche qualitative Mängel aufweisen, erstellen wir einen eigenen Gold-Standard, in dem Anfragen in Sitzungen gruppiert sind und jede Sitzung einer Mission zugeordnet ist. Damit präsentieren wir den bisher größten Gold-Standard (mit 8840 Anfragen etwa 6-mal so groß wie der von Lucchese et al. [LOP+11]), der zur Evaluation von Verfahren zur Missionserkennung geeignet ist.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Diese Arbeit befasst sich mit verschiedenen Methoden zur automatischen Suchsitzungserkennung (engl. *Query Session Detection*) in Anfrage-Logs von Websuchmaschinen. Als Suchsitzungserkennung bezeichnet man das Zusammenfassen aufeinander folgender Suchanfragen zu inhaltlich zusammengehörenden Blöcken. Solche Suchsitzungen können dem Betreiber einer Suchmaschine helfen, den eigenen Service zu verbessern. Zum einen erlauben sie Rückschlüsse auf die Ursachen von nicht erfolgreichen Suchanfragen (wenige Klicks in einer Reihe von Anfragen zum selben Informationsbedarf weisen oft auf eine nicht erfolgreiche Suche hin). Zum anderen können dem Nutzer ad-hoc Ergebnisse präsentiert werden, die im Kontext zu der aktuellen Sitzung stehen. Sucht man beispielsweise mit der Google-Suchmaschine nach `tokio hotel` verweisen die ersten sieben Suchergebnisse auf Webseiten rund um „eine deutsche Band aus dem Raum Magdeburg“[1], während erst das achte Ergebnis eine Auflistung von Hotels der Stadt Tokio beinhaltet. Unter Zuhilfenahme der Suchsitzungserkennung könnten dieses und ähnliche Ergebnisse weiter oben aufgeführt werden, wenn bekannt ist, dass vorher beispielsweise nach Sehenswürdigkeiten in Tokio oder einem Flug von Frankfurt nach Tokio gesucht wurde.

Basierend auf der Idee eines kaskadierenden Verfahrens zur Suchsitzungserkennung von Hagen et al. [HSR11], wird in dieser Arbeit ein erweitertes kaskadierendes Verfahren vorgestellt und hinsichtlich seiner Effizienz und Güte untersucht. Beide Kriterien können je nach Situation einzeln oder zusammen eine wichtige Rolle spielen. Möchte man einen Anfrage-Log lediglich im Nachhinein untersuchen, so sollen die Sitzungen möglichst genau erkannt werden – die benötigte Zeit spielt hierbei eine untergeordnete Rolle. Sollen dem Nutzer allerdings schon während der Suche kontextabhängige Ergebnisse präsentiert werden, so muss die Sitzungserkennung sowohl genau als auch besonders schnell

[1] `http://de.wikipedia.org/w/index.php?title=Tokio_Hotel&oldid=101277783`, Letzter Zugriff: 27. März 2012

ID	Anfrage	Zeit	Rang	URL
11262510	`new york aquarium`	2006-04-20 23:01:34		
11262510	`new york aquarium`	2006-04-20 23:05:30	1	http://www.nyaquarium.com
11262510	`evelyn's lounge`	2006-04-21 20:41:50	1	http://www.newyorkmetro.com
11262510	`new york zoo`	2006-04-23 21:12:04	1	http://www.centralparkzoo.com
11262510	`mohegansun`	2006-05-06 20:30:24	2	http://mohegansun.casinocity.com
11262510	`mohegansun`	2006-05-06 20:30:24	1	http://www.mohegansun.com
11262510	`weekend getaways`	2006-05-06 20:40:13		

Tabelle 1.1: Typischer Ausschnitt aus dem AOL Query Log von 2006.

arbeiten. Um eine möglichst hohe Effizienz zu erreichen, ordnen Hagen et al. die einzelnen Schritte in einer Kaskade an [HSR11]. So werden rechen- und zeitaufwändige Schritte nur dann ausgeführt, wenn schnellere keine genaue Aussage darüber treffen können, ob zwei Suchanfragen zur selben Sitzung gehören oder nicht (mehr dazu in Abschnitt 4.2).

Neben der Erkennung von Suchsitzungen möchten wir in dieser Arbeit auch versuchen, die übergeordneten Missionen dieser Sitzungen automatisch zu erfassen, d. h. Sitzungen sollen daraufhin untersucht werden, ob ein inhaltlicher Zusammenhang zu einer früheren – unterbrochenen – Sitzung besteht. Dieses zusätzliche Wissen beinhaltet natürlich noch mehr Details über das Suchverhalten der Nutzer und ist entsprechend wertvoll für Analysen seitens der Suchmaschinenbetreiber. Längerfristig wird hier aber das Ziel verfolgt, dass ein Algorithmus entscheiden kann, worin die Mission besteht und welche weiteren Aufgaben dazu gehören könnten, um den Nutzer mit entsprechenden Vorschlägen zu versorgen. Wurden beispielsweise Flüge von Frankfurt nach Tokio gesucht und bereits ein Hotel gebucht, könnten Busverbindungen vom Flughafen zum Hotel oder eine Liste von örtlichen Taxi-Unternehmen in den Suchergebnissen angezeigt werden.

Als Grundlage für die Untersuchungen dient ein Auszug aus dem Anfrage-Log der AOL-Suchmaschine von 2006 [PCT06], den wir manuell in Sitzungen eingeteilt und jeder Anfrage eine *Missions-ID* zugeordnet haben. Einen typischen Auszug aus dem Original-Log kann man der Tabelle 1.1 entnehmen. Neben einer Nutzer-ID wird genau festgehalten, welche Anfrage zu welchem Zeitpunkt gestellt wurde. Hat der Nutzer ein Ergebnis angeklickt, so wurden zusätzlich die Domain und der Rang auf der Ergebnisseite protokolliert.

Man kann gut erkennen, dass die ersten beiden Anfragen zu derselben Sitzung gehören. Bei Evelyn's Lounge handelt es sich um eine ehemalige Bar in New York. Da diese nicht mit dem Aquarium zusammenhängt, kann die Anfrage einer neuen Mission und damit einer neuen Sitzung zugeordnet werden. Die folgende Anfrage kann als alternativer Ausflugsort zum zuvor gesuchten Aquarium verstanden werden und gehört somit zur selben Mission, weil hier eine frühere Sitzung erneut aufgenommen wurde. Die Begriffe `mohegansun` und

ID	Anfrage	Zeit	Rang	URL	Missions-ID
11262510	`new york aquarium`	2006-04-20 23:01:34			1
11262510	`new york aquarium`	2006-04-20 23:05:30	1	http://www.nyaquarium.com	1
11262510	`evelyn's lounge`	2006-04-21 20:41:50	1	http://www.newyorkmetro.com	2
11262510	`new york zoo`	2006-04-23 21:12:04	1	http://www.centralparkzoo.com	1
11262510	`mohegansun`	2006-05-06 20:30:24	2	http://mohegansun.casinocity.com	3
11262510	`mohegansun`	2006-05-06 20:30:24	1	http://www.mohegansun.com	3
11262510	`weekend getaways`	2006-05-06 20:40:13			3

Tabelle 1.2: Der bearbeitete Ausschnitt mit dem zusätzlichen Feld Missions-ID.

`weekend getaways` besitzen zwar keine syntaktische Ähnlichkeit, aber man kann erahnen, dass der Nutzer Pläne für das Wochenende schmiedet, weshalb diese Anfragen wohl zur selben Sitzung gehören. Tabelle 1.2 zeigt den bearbeiteten Auszug mit den hinzugefügten Missions-IDs.

Zur Evaluation der teils überarbeiteten, teils neu entwickelten Schritte der Suchsitzungserkennung testen wir unser Verfahren auf einem manuell annotierten Datensatz, dem sogenannten Gold-Standard (engl. *Ground Truth*). Da die uns vorliegenden Gold-Standards von anderen Forschungsgruppen nicht die gewünschte Qualität aufweisen, werden wir im Rahmen dieser Arbeit einen neuen Gold-Standard erstellen. Neben seiner besseren Qualität stellt dieser auch die Grundlage zur Überprüfung der (bisher nur wenig untersuchten) Missionserkennung dar – die meisten anderen Gold-Standards teilen Anfragen nämlich nur in Sitzungen, nicht aber in Missionen ein.

Nach einem kurzen Überblick über bereits veröffentlichte Arbeiten in Kapitel 2 wird in Kapitel 3 beschrieben, warum ein neuer Gold-Standard nötig war und wie dieser entstanden ist. Anschließend wird im Kapitel 4 das kaskadierende Verfahren zur Suchsitzungserkennung und im Kapitel 5 die Missionserkennung beschrieben. Im Kapitel 6 werden die Ergebnisse des entwickelten Verfahrens auf dem Gold-Standard evaluiert und Kapitel 7 schließt die Arbeit mit einem Ausblick auf zukünftige Arbeiten ab.

Kapitel 2

Verwandte Arbeiten

Die ersten Untersuchungen zur Suchsitzungserkennung fanden in den späten 1990er Jahren statt [JSBS98, Spi98]. Diese Arbeiten befassten sich zunächst mit dem 1997 veröffentlichten Suchanfrage-Log der Excite-Suchmaschine. Die Autoren untersuchten das Suchverhalten der Nutzer, indem sie den Datensatz statistisch auswerteten und verwendeten erstmals die Begriffe *session* und *search episode*, um eine Reihe zeitlich nah beieinander liegender Suchanfragen zusammenzufassen. Allerdings sollte sich die Bedeutung dieser Begriffe in den folgenden Jahren mehrfach ändern. Im Jahr 1999 definierten Silverstein et al. eine Sitzung als Reihe von Anfragen in einem kleinen Zeitfenster, die einen einzigen Informationsbedarf decken sollen [SMHM99]. Diese Definition wurde aber von verschiedenen Forschungsgruppen nicht angenommen, weshalb parallel die Definition existierte, dass eine Sitzung alle Anfragen eines Nutzers über den gesamten aufgezeichnten Zeitraum darstellt [JSS00]. Um dennoch Anfragen kontextbezogen gruppieren zu können, führten Radlinski und Joachims im Jahr 2005 den nur noch selten verwendeten Begriff *chain* ein [RJ05], eine Reihe von Anfragen, die demselben Informationsbedarf dienen. Im Jahr 2006 definierten Jansen und Spink eine *searching episode* als Zeitspanne zwischen der ersten und letzten Anfrage eines Nutzers an einem Tag [JS06]. Erst ein Jahr später übernahmen sie die mittlerweile acht Jahre alte Definition von Silverstein et al. und fügten hinzu, dass eine *searching episode* aus mehreren Sitzungen bestehen kann [JSBK07]. Dies ist die heute allgemein anerkannte Definition und auch wir werden uns in dieser Arbeit hierauf beziehen. Dennoch gibt es immer noch einige Arbeiten, in denen der Begriff *Sitzung* mit einer anderen Bedeutung auftritt.

Definition. *Eine Sitzung beschreibt eine Reihe von Interaktionen eines Nutzers mit einer Suchmaschine, die der Erfüllung eines bestimmten Informationsbedarfes dienen.*

2.1 Suchverhalten von Nutzern

Lau und Horvitz bestimmen in ihrer Arbeit einige Nutzer-Verhaltensmuster bei der Benutzung von Suchmaschinen [LH99]. Diese wurden 2002 von He et al. wieder aufgenommen, etwas angepasst und auf das Problem der Suchsitzungserkennung übertragen. He et al. erkennen die acht folgenden Muster [HGH02]:

1. **Blättern** - zwischen den Suchergebnisseiten navigieren
2. **Generalisierung** - einen Term aus der letzten Anfrage entfernen (z. B. `tokio hotel` → `tokio`)
3. **Spezialisierung** - einen Term zu der letzten Anfrage hinzufügen (z. B. `tokio` → `tokio hotel`)
4. **Umformulierung** - Terme durch andere ersetzen (z. B. `tokio hotel` → `tokio accommodation`)
5. **Wiederholung** - die letzte Anfrage wird unverändert erneut geschickt (z. B. `tokio` → `tokio`)
6. **Neu** - die beiden Anfragen behandeln verschiedene Themen (z. B. `tokio hotel` → `super bowl results`)
7. **Suchvorschlag** - Klick auf einen, von der Suchmaschine vorgeschlagenen, ähnlich relevanten Suchbegriff bzw. Rechtschreibkorrektur
8. **Andere** - Interaktionen mit der Suchmaschine, die in keine der genannten Kategorien eingeordnet werden können, wie z. B. einen Filter auf die Suchergebnisse der letzten Anfrage anwenden

Verschiedene Forschungsgruppen bestätigten weitestgehend die aufgezählten Suchmuster [ÖÇ05a, JSBK07], weisen aber darauf hin, dass nur die Punkte 2 bis 6 für die Suchsitzungserkennung relevant sind [Gay09]. Auch für diese Arbeit treffen nur die genannten Punkte zu, da die übrigen Muster nicht im AOL-Log enthalten sind: So entspricht *Blättern* einer *Wiederholung* und bei einem Klick auf einen *Suchvorschlag* steht im Anfrage-Log lediglich die neue Anfrage, aber kein Hinweis darauf, dass es sich hier um einen Vorschlag von der Suchmaschine handelt. *Andere* werden nicht gesondert aufgeführt.

2.2 Suchsitzungserkennung

Um zu entscheiden, ob zwei Anfragen inhaltlich zusammengehören, wurden in der Literatur bereits verschiedene Techniken beschrieben, die im Folgenden vorgestellt werden.

Zeitliche Merkmale

Seit dem Beginn der Untersuchungen zur Suchsitzungserkennung wird der simple Ansatz verfolgt, Anfragen aufgrund ihrer zeitlichen Distanz zu Sitzungen zusammenzufassen bzw. in getrennte Sitzungen einzuordnen. Dabei werden in verschiedenen Arbeiten diverse Schwellwerte verwendet. Diese reichen im Allgemeinen von fünf Minuten [SMHM99, HCO03] bis zu 30 Minuten [RJ05]. He und Göker nehmen in ihrer Arbeit aus dem Jahr 2000 einen Schwellwert zwischen 10 und 15 Minuten an, stellen aber fest, dass zeitliche Ähnlichkeit allein kein gutes Kriterium ist [HG00]. Jones und Klinkner bestätigten 2008 diese Aussage und ermittelten, dass mit festen Schwellwerten für die verstrichene Zeit zwischen zwei Anfragen nur eine maximale Genauigkeit von 70 % erreicht werden kann [JK08]. Auch Murray et al. waren schon 2006 der Meinung, dass feste Schwellwerte nicht über verschiedene Datensätze hinweg für jeden Nutzer allgemein gültig sein können. Daher ermittelten sie einen eigenen Schwellwert für jeden Nutzer anhand dessen Such-Interaktionen über den gesamten aufgezeichneten Zeitraum. Allerdings weisen sie in ihrer Arbeit darauf hin, dass dieser letztlich nur die Nutzer-Aktivität widerspiegelt und auch kein ausreichendes Kriterium für eine Sitzungserkennung darstellt [MLC06].

Die Bestimmung der Sitzungszugehörigkeit zweier Anfragen anhand zeitlicher Distanz birgt im Kern zwei Fehlerquellen: Einerseits werden Anfragen, die zeitlich nah beieinander liegen, inhaltlich aber nicht zusammen gehören, zur selben Sitzung zugeordnet. Da Jones und Klinkner in ihrer Arbeit beobachteten, dass Themenwechsel häufig innerhalb weniger Sekunden stattfinden [JK08], ist davon auszugehen, dass hier die meisten Fehlentscheidungen getroffen werden. Andererseits werden Anfragen, die ein Nutzer vor der Mittagspause gestellt hat, pauschal von denen getrennt, die nach der Pause erfolgten, aber demselben Informationsbedarf dienten. Vor allem dieser Fehler kann minimiert werden, indem die lexikalische Ähnlichkeit der beiden Anfragen untersucht wird.

Lexikalische Merkmale

Auch die Überprüfung auf Ähnlichkeit der Zeichenketten lässt sich in primitivere sowie fortgeschrittenere Techniken unterteilen.

Zu den primitiven Mitteln gehört vor allem die Betrachtung auf Termbasis. Hier wird häufig auf die bereits vorgestellten Suchmuster von He et al. zurückgegriffen, d. h. man prüft, ob die zweite Anfrage eine Wiederholung, eine Spezialisierung oder eine Generalisierung der ersten darstellt. Trifft einer dieser Fälle zu, kann man mit hoher Wahrscheinlichkeit davon ausgehen, dass beide Anfragen derselben Sitzung angehören. Wird kein einziger über-

lappender Term gefunden, so wird (ohne weiterführende Betrachtung) davon ausgegangen, dass die zweite Anfrage einer neuen Sitzung zugeordnet werden sollte [JSBK07, ÖÖS08]. Seco und Cardoso betrachten die zeitliche und die lexikalische Ähnlichkeit zusammen und legen fest, dass eine Anfrage zur vorherigen Sitzung gehört, wenn zwischen der letzten und aktuellen Anfrage weniger als 60 Minuten vergangen sind *und* wenigstens ein Term in den Anfragen der vorherigen Sitzung enthalten ist [SC06].

Darüber hinaus existieren Techniken, die über diese simplen Betrachtungen hinausgehen. So werden zum Beispiel in einigen Arbeiten beide Anfragen in Mengen von n-Grammen überführt und der Jaccard-Koeffizient dieser beiden Mengen ermittelt [ZM06], der bestimmt, wie ähnlich sich beide Mengen sind. Alternativ zu diesem Ansatz wird von einigen Forschungsgruppen die Ähnlichkeit durch die sogenannte Levenshtein-Distanz bestimmt [SLY+10], die beschreibt, wie viele Operationen (Einfügen, Löschen oder Ersetzen von Buchstaben) nötig sind, um die erste Anfrage in die zweite zu überführen. Lucchese et al. beziehen in ihre Berechnungen sogar eine Kombination aus den beiden Methoden ein [LOP+11], beschreiben aber nicht konkret, wie sich dies auf die Ergebnisse auswirkt.

Gayo-Avello führte im Jahr 2009 die geometrische Methode ein, die nach seiner eigenen Aussage im direkten Vergleich zu allen vorher bekannten Verfahren bessere Ergebnisse erzielt [Gay09]. Auch die geometrische Methode kombiniert ein zeitliches und ein lexikalisches Merkmal, um eine Kennzahl für die Ähnlichkeit zweier Anfragen zu bilden (siehe Abschnitt 4.4). Gayo-Avello stellt in seiner Arbeit fest, dass n-Gramme zur Untersuchung der lexikalischen Ähnlichkeit besser geeignet sind als ganze Terme, weil sich hierdurch zwei Vorteile ergeben:

1. Das Stemming der Terme (d. h. die Rückführung auf ihren Wortstamm) entfällt.

2. Anfragen, die Rechtschreibfehler enthalten, sind dennoch sehr ähnlich zu ihren Korrekturen, da z. B. bei Buchstabendrehern einzelne verschiedene n-Gramme insgesamt nur wenig an der Ähnlichkeit ändern.

Zur Bestimmung der lexikalischen Ähnlichkeit verwendet Gayo-Avello ein dem Jaccard-Koeffizienten sehr ähnliches Merkmal, welches von Hagen et al. durch die Kosinusähnlichkeit[1] zwischen den n-Gramm-Vektoren der Anfragen ausgetauscht wurde [HSR11].

[1] Bei der Kosinusähnlichkeit wird der Winkel zwischen zwei mehrdimensionalen Vektoren berechnet. Ist dieser sehr klein, sind sich beide Anfragen, die durch den Vektor repräsentiert werden, sehr ähnlich.

Lexikalische Merkmale können (vor allem in Kombination mit einem zeitlichen Merkmal) gegenüber der alleinigen Verwendung zeitlicher Merkmale schon deutlich bessere Ergebnisse bei der Sitzungserkennung erzielen, jedoch können sie im Allgemeinen nur die Verhaltensmuster *Generalisierung*, *Spezialisierung* und *Wiederholung* (siehe Abschnitt 2.1) erkennen, aber nur selten das Muster *Umformulierung*. In solchen Fällen handelt es sich oftmals um eine komplette Neuformulierung (z. B. `wimbledon winner 89` → `boris becker`) oder den Gebrauch von Abkürzungen (z. B. `information retrieval` → `ir`). Wenngleich diese Zusammenhänge leicht von einem Menschen erkannt werden können, stellt diese Aufgabe für die automatische Sitzungserkennung eine große Hürde dar. Aus diesem Grund werden wir in dieser Arbeit zusätzlich Verfahren anwenden, die in der Lage sind, inhaltliche Zusammenhänge zwischen zwei Entitäten zu erkennen.

Semantische Ähnlichkeit

Bisher wurden in der Literatur zwei Verfahren vorgestellt, die die semantische Ähnlichkeit für ein Anfragepaar ermitteln können.

Bei dem ersten Verfahren handelt es sich um die *Explicit Semantic Analysis*, kurz *ESA*, die 2007 von Gabrilovich und Markovitch vorgestellt wurde [GM07]. Dieses Verfahren wurde ursprünglich für die Ähnlichkeitsbestimmung zwischen zwei Dokumenten vorgeschlagen, die auf Ähnlichkeit zu einem Satz von Wikipedia-Artikeln (üblicherweise 100 000 Artikel) geprüft werden. Beide Eingabedokumente werden als Vektor dargestellt, dessen Komponenten die Ähnlichkeiten zu den einzelnen Wikipedia-Artikeln beschreiben. Die Kosinus-Ähnlichkeit zwischen diesen beiden Vektoren beschreibt, wie ähnlich sich beide Dokumente sind. Hagen et al. wandeln die Explizite Semantische Analyse leicht ab und wenden sie auf Suchanfragen statt kompletter Dokumente an, um so inhaltliche Zusamenhänge zwischen zwei Anfragen zu erkennen [HSR11]. Dabei bauen sie darauf, dass inhaltlich verwandte Suchanfragen eine hohe Ähnlichkeit zu denselben Artikeln aufweisen. So zum Beispiel weisen die Suchbegriffe `colosseum` und `pantheon` eine hohe Ähnlichkeit zu dem Artikel *Rome* auf, aber nicht zu *Hong Kong*.

Das zweite Verfahren beruht auf *Linked Open Data* (kurz *LOD*), eine Sammlung von frei verfügbaren Informationen im Internet nach der Idee von Tim Berners-Lee.[2] Hollink et al. beschreiben in ihrer Arbeit, wie sie Suchanfragen aus dem Anfrage-Log einer kommerziellen Bildsuchmaschine auf Entitäten eines solchen LOD-Graphen abbilden und nach Verbindungen zwischen diesen suchen [HTV11]. Um eine Vorstellung eines solchen LOD-Graphen zu

[2] `http://www.w3.org/DesignIssues/LinkedData.html`, Letzter Zugriff: 27. März 2012

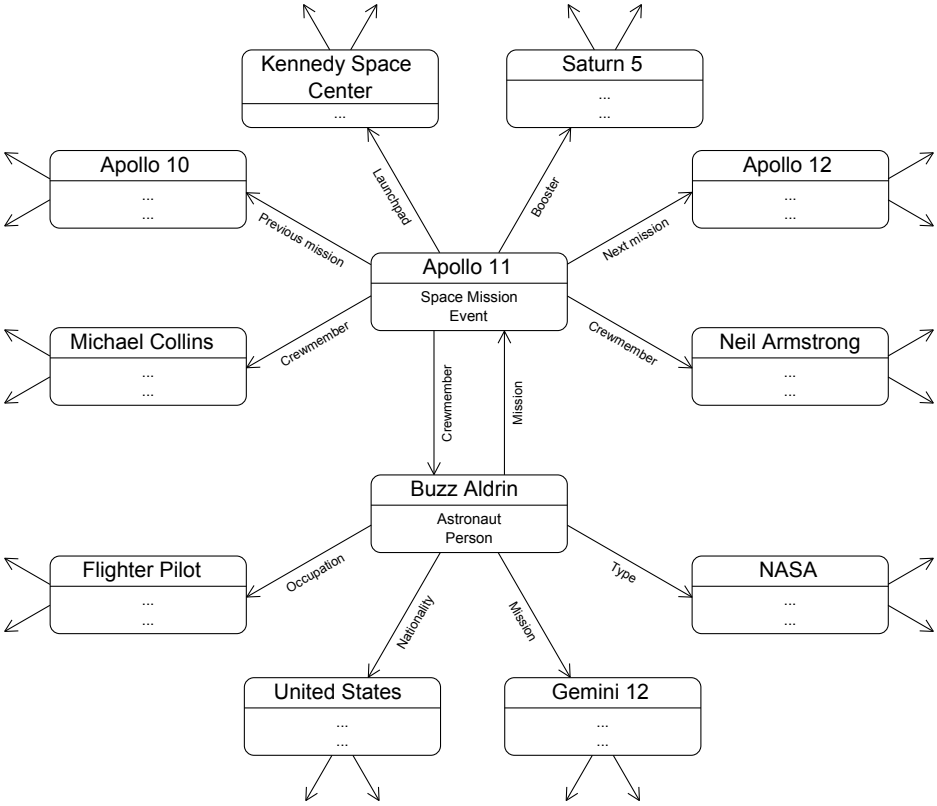


Abbildung 2.1: Teilgraph des DBpedia-Datensatzes.

vermitteln, wird in Abbildung 2.1 ein Teilgraph des verwendeten DBpedia-Datensatzes für die beiden Entitäten *Apollo 11* und *Buzz Aldrin* dargestellt. Da ein solcher Graph schnell sehr komplex werden kann, sind die Beziehungspfeile der äußeren Entitäten nur exemplarisch angedeutet. So zum Beispiel enthalten die Entitäten *Neil Armstrong* und *Michael Collins* ebenfalls die Attribute *NASA*, *Flighter Pilot* und *United States*. Wären auch diese und andere Verbindungen hier eingezeichnet, würden die vielen Pfeile schnell zum Verlust der Übersicht führen, weshalb hier auf sie verzichtet wird.

Hollink et al. sind mittels ihrer vorgeschlagenen Technik in der Lage, Zusammenhänge wie z. B. zwischen Andre Agassi und Boris Becker zu finden (beide sind vom Typ *Tennis Player*). Da sie auf der Basis einer Bildsuchmaschine arbeiten, die häufig nur von Journalisten benutzt wird, tauchen in den Anfragen größtenteils Namen von Prominenten auf (44 %) [HTV11]. Die Suche nach anderen Typen, wie z. B. Technologien oder geografischen Informationen, treten in ihren Untersuchungen vergleichsweise selten auf. Auch waren viele Anfragen konzeptioneller Natur, d. h. es handelte sich um Anfragen, die nicht

auf eine spezifische Entität abzielen, sondern auf allgemeine Begrifflichkeiten. Beispielsweise zielt die Anfrage `skyscraper` auf ein Konzept ab, während der Nutzer bei der Anfrage `empire state building` eine konkrete Vorstellung davon hat, wonach er sucht [HTV11]. Hollink et al. stellen fest, dass ihr Verfahren für solche konzeptionellen Anfragen nicht gut geeignet ist und daher insgesamt nur mit begrenzter Genauigkeit arbeitet. Dennoch ist der Ansatz sehr vielversprechend und soll daher in dieser Arbeit für Anfragen von „normalen“ Suchmaschinen untersucht werden.

Erweiterung der Anfragen

Ein Problem in der Suchsitzungserkennung stellt die Kürze der Suchanfragen dar. Einige Forschungsgruppen erweitern deshalb Anfragen mit Teilen der Ergebnismenge. Wie viele und welche Teile der Ergebnisse in die Erweiterung der Anfrage einfließen, ist wieder von Forschungsgruppe zu Forschungsgruppe verschieden.

Radlinski und Joachims vergleichen die URLs der ersten zehn Suchergebnisse beider Anfragen miteinander [RJ05]. Wird wenigstens eine Übereinstimmung gefunden, gehen sie davon aus, dass beide Anfragen zusammengehören.

Shen et al. verwenden statt der URLs die Titel und Snippets der ersten 50 Suchergebnisse, überführen sie in zwei $tf{\cdot}idf$-gewichtete Vektoren und ermitteln die Kosinus-Ähnlichkeit zwischen diesen [STZ05]. Übersteigt diese einen vorher festgelegten Schwellwert, werden beide Anfragen derselben Sitzung zugeordnet.

Sahami und Heilman verwenden sogar die kompletten Ergebnisdokumente zur Erweiterung von kurzen Textfragmenten [SH06], beziehen sich dabei aber nicht explizit auf die Suchsitzungserkennung.

Gayo-Avello schlägt vor, zusätzlich zu den Snippets auch die ersten vier Kilobyte eines Wikipedia-Artikels in die Erweiterung mit einzubeziehen, falls das erste Suchergebnis auf einen solchen verweist [Gay09].

Auch Hagen et al. greifen auf die Anfrage-Erweiterung zurück und überprüfen, ob mindestens eines der ersten zehn Suchergebnisse beider Anfragen auf die selbe Domain verweist und entscheiden sich in diesem Fall für eine Sitzungsfortführung [HSR11]. Allerdings weisen sie darauf hin, dass die von ihnen implementierte Anfrage-Erweiterung sehr langsam ist und das Gesamtergebnis der Kaskade nur unwesentlich verbessert. Daher schlagen sie vor, zugunsten einer besseren Laufzeit komplett auf diesen Schritt zu verzichten [HSR11].

Maschinelles Lernen

Zuguterletzt kann mittels Verfahren des Maschinellen Lernens (engl. *Machine Learning*) eine Vielzahl verschiedener Merkmale miteinander kombiniert bzw. ein Entscheidungsbaum entworfen werden. Prinzipiell kann eine Kombination verschiedener Merkmale durchaus sinnvoll sein, wie beispielsweise bei der geometrischen Methode, aber maschinelle Lernverfahren liefern häufig Ergebnisse, bei denen es keine intuitive Erklärung dafür gibt, warum sie gute oder schlechte Ergebnisse erzielen. Bei Entscheidungsbäumen können Fälle entstehen, in denen die Schrittreihenfolge schlichtweg keinen Sinn ergibt, z. B. wenn die (zeitaufwändige) Anfrage-Erweiterung vor simplen lexikalischen Überprüfungen ausgeführt wird. Überhaupt wird bei Einsatz maschineller Lernverfahren häufig die Effizienz vernachlässigt und jedes Merkmal mit jedem Merkmal kombiniert. Wir wollen in dieser Arbeit aber neben einer hohen Güte auch eine möglichst niedrige Laufzeit erzielen und sehen daher von solchen Verfahren ab. Dennoch sollen hier einige Ansätze aus der Literatur genannt werden.

He et al. ermitteln unter Zuhilfenahme der Dempster-Shafer-Evidenztheorie für jede mögliche Kombination aus verstrichener Zeit und den acht von ihnen vorgestellten Suchmustern (siehe Abschnitt 2.1) einen Wert, der beschreibt, wie wahrscheinlich eine Sitzungsfortführung bzw. eine neue Suchsitzung vorliegt [HGH02].

Özmutlu und Çavdur nehmen drei Jahre später diesen Vorschlag wieder auf [ÖÇ05a] und läuten damit eine ganze Reihe von Untersuchungen ein, die überprüfen sollen, wie geeignet maschinelle Lernverfahren für die Suchsitzungserkennung sind. Unter anderem werden neuronale Netze trainiert [ÖÇ05b], auf Basis einer Monte-Carlo-Simulation entschieden [ÖÖB07] und verschiedene Merkmale mittels der multiplen linearen Regression kombiniert [ÖÖS08]. Später stellt Gayo-Avello jedoch fest, dass keines dieser Verfahren trotz ihrer deutlich höheren Komplexität bessere Ergebnisse als die von ihm vorgestellte geometrische Methode erzielt [Gay09].

Lediglich Jones und Klinkner erzielen laut ihrer eigenen Aussage mit logistischer Regression von verschiedenen Merkmalen (wie z. B. Levenshtein-Distanz, Jaccard-Koeffizient oder Anzahl übereinstimmender Worte) sehr gute Ergebnisse [JK08]. Jedoch liefen diese Untersuchungen auf einem eigens erstellten und nicht frei verfügbaren Datensatz, sodass hinsichtlich der Überprüfung der Ergebnisse nur wenig Transparenz besteht.

2.3 Missionserkennung

Jones und Klinkner definieren eine Mission als ein übergeordnetes Ziel, zu dessen Erfüllung u. U. verschiedene Informationen benötigt werden, und somit in

mehreren Suchsitzungen resultieren kann [JK08]. Beispielsweise könnten untergeordnete Ziele bei der Planung des nächsten Urlaubes die Suche nach dem günstigsten Flug, die Buchung eines Hotels oder das Sammeln von Informationen über Verkehrsregeln im Reiseland sein. Die einzelnen untergeordneten Ziele können natürlich auch über mehrere Tage oder sogar Wochen hinweg verteilt sein und immer wieder von anderen Sitzungen unterbrochen und später wieder aufgenommen werden. Selbes gilt für umfangreichere Rechercheaufträge (z. B. Autokauf oder die Suche nach verwandten Arbeiten während einer Bachelorarbeit), die häufig auch als Mission zu verstehen sind. Hierbei tritt vor allem die Besonderheit auf, dass Anfragen immer spezieller werden, da sie auf neuem Wissen aufbauen, das durch vorausgegangene Suchanfragen erworben wurde: Behandeln Suchanfragen zu Beginn der Bearbeitung eines Themas oft nur die Grundlagen, gehen diese mit der Zeit immer mehr ins Detail.

Eine Mission besteht immer aus mindestens einer Sitzung, kann aber auch aus beliebig vielen Sitzungen bestehen. Gehören Anfragen zur selben Suchsitzung, so gehören sie auch zur selben Mission. Missionen können, wie bereits erwähnt, unterbrochen und später erneut aufgenommen werden.

Bearbeitet der Nutzer verschiedene Problemstellungen und stellt innerhalb kurzer Zeit Anfragen zu verschiedenen Themengebieten, d. h. er sucht nach Informationen für verschiedene Missionen, sprechen Spink et al. von *Multitasking* [SÖÖ02]. Sie führen allerdings nur den Begriff ein, beschreiben aber nicht, wie dieses Verhaltensmuster automatisch erkannt werden kann.

Erst Jones und Klinkner befassen sich mit einer automatischen Missionserkennung. Sie stellen fest, dass 17 % der Aufgaben eines Nutzers miteinander verzahnt bzw. ineinander verschachtelt und 20 % hierarchisch organisiert sind [JK08]. Daher bauen sie eine Struktur aus *missions* und *goals* auf (ein Ziel entspricht laut unserer Terminologie einer Sitzung), in die sie die Anfragen einordnen. Laut ihrer eigenen Aussage gelingt ihnen dies auch recht gut, jedoch ist dieses Ergebnis aufgrund der mangelnden Transparenz des Gold-Standards und bei der unklaren Kombination der vielen verschiedenen Merkmale fraglich.

Auch in der bisher einzigen anderen verfügbaren Arbeit zur Missionserkennung von Lucchese et al. konzentrieren sich die Autoren auf die Kombination einer Vielzahl von Merkmalen mit maschinellen Lernverfahren [LOP+11] und erreichen damit passable, aber auch nicht sonderlich gute, Ergebnisse. Das Problem mit dieser Arbeit besteht unserer Ansicht nach vor allem in dem mangelhaften Gold-Standard (siehe Abschnitt 3.1). Darüber hinaus betonen die Autoren zunächst, dass ein fester Schwellwert für den zeitlichen Abstand zweier Anfragen keine hohe Aussagekraft für eine Entscheidung über Sitzungsfortführung oder -grenze besitzt, verwenden aber im anschließenden Preprocessing ihres Verfahrens selbst einen festen Schwellwert von 26 Minuten, um die Anfragen des Logs in ihre so benannten *time sessions* einzuteilen [LOP+11].

Kapitel 3

Überarbeiteter Gold-Standard

Um die Ergebnisse von Algorithmen zur Suchsitzungserkennung auf ihre Genauigkeit zu prüfen, benötigt man einen Referenzdatensatz, die sogenannte *Ground Truth*, die im deutschen Sprachgebrauch häufig auch als Gold-Standard bezeichnet wird. Es handelt sich hier um einen Auszug aus einem Suchmaschinen-Anfrage-Log, dessen Anfragen manuell in Sitzungen eingeteilt werden. In der Literatur werden diverse Datensätze von verschiedenen Suchmaschinen verwendet, die aber meist nicht öffentlich verfügbar sind. Zudem wird in einigen Arbeiten nur unzureichend beschrieben, wie und nach welchen Kriterien diese Datensätze überhaupt entstanden sind, z. B. in [BBC+08]. Uns liegen daher nur zwei benutzbare Gold-Standards vor, die allerdings bei genauerer Betrachtung viele Mängel aufweisen, sodass im Rahmen dieser Arbeit ein eigener Gold-Standard entstehen soll. Nach einer kurzen Beschreibung der beiden vorliegenden Datensätze und ihrer Mängel in den nächsten zwei Abschnitten folgt eine ausführlichere Beschreibung unseres eigenen Datensatzes. Zum Abschluss des Kapitels werden Evaluationsmaße vorgestellt, die beschreiben, wie viel Übereinstimmung zwischen den Ergebnissen eines Verfahrens zur Suchsitzungserkennung und dem Gold-Standard besteht.

3.1 Datensatz von Lucchese et al.

Lucchese et al. stellten im Februar 2011 einen Gold-Standard vor [LOP+11], der online bezogen werden kann.[1] Da der Datensatz etwas unübersichtlich organisiert ist, muss dieser zur weiteren Untersuchung zunächst in eine sinnvolle Form überführt werden. Dies gestaltet sich allerdings problematisch, da aus dem Original-AOL-Log von 2006 lediglich die Nutzer-ID und die eigentliche Anfrage übernommen wurden; Anfrage-Zeitpunkt, geklickte URL und Rang

[1] `http://miles.isti.cnr.it/~tolomei/?page_id=36`, Letzter Zugriff: 27. März 2012

ID	Anfrage	(Zeit)
117514	`for gump 1994`	(2006-04-09 15:24:24)
117514	`forest gump 1994`	(2006-04-09 15:26:18)
117514	`forest gump 1994`	(2006-04-09 15:32:04)
117514	`forest gump 1994`	(2006-04-09 15:32:52)

Tabelle 3.1: Grober Fehler im Gold-Standard von Lucchese et al.

wurden im Gold-Standard von Lucchese et al. verworfen, wobei vor allem der Zeitstempel sehr wichtig für die Suchsitzungserkennung ist. Eine anschließende maschinelle Analyse zeigt, dass lediglich 13 Nutzer mit gerade einmal 1424 Anfragen in dem Log enthalten sind – von diesen ohnehin schon wenigen Nutzern wurden insgesamt 44 715 Anfragen entfernt. Der Datensatz spiegelt daher nur schlecht den originalen AOL-Log wider. Dies zeigt sich zum Beispiel auch in der Anzahl der wiederholten Anfragen: Während im AOL-Log 42,3 % aller Anfragen genauso lauten wie ihre vorausgegangene, sind es im Lucchese-Datensatz gerade einmal 3,7 %. Desweiteren sind 345 URL-Anfragen (siehe Abschnitt 3.3) enthalten, die aber unserer Meinung nach herausgefiltert werden sollten, da der Nutzer hier keine Suchanfrage im herkömmlichen Sinne gestellt, sondern die URL vermutlich versehentlich in das Suchfeld der Suchmaschine statt in die Adressleiste des Browsers eingetippt hat. Teilweise sind auch grobe Fehler in Bezug auf die Sitzungseinteilung enthalten, die bei manueller Analyse eigentlich nicht vorkommen sollten. Ein Beispiel für einen solchen Fehler kann man der Tabelle 3.1 entnehmen: Zwischen zwei exakt gleich lautenden Suchbegriffen liegen nur wenige Minuten (die Zeitstempel wurden aus dem AOL-Log rekonstruiert) und trotzdem wurden sie zwei verschiedenen Suchsitzungen zugeordnet.

Letztlich ist der Datensatz von Lucchese et al. für eine sinnvolle Untersuchung zu klein und weist neben den fehlenden Informationen (URL, Rang und Zeitstempel) zu viele Mängel auf, sodass dieser den Original-AOL-Log nur schlecht repräsentiert.

3.2 Der Gold-Standard von Gayo-Avello

Daniel Gayo-Avello erstellte im Mai 2009 einen deutlich repräsentativeren Datensatz [Gay09]. Dieser beinhaltet 11 484 Anfragen von 215 Nutzern,[2] die einem geeigneten Querschnitt durch den AOL-Log entsprechen. So resultieren beispielsweise 58,5 % der Anfragen in einem Klick auf ein Ergebnis (im AOL-

[2]In der Quelle selbst stehen zwar 223 Nutzer, aber unsere maschinelle Analyse zählt nur 215 verschiedene Nutzer.

Log sind es 53,4 %) und 39,7 % aller Anfragen sind Wiederholungen (im AOL-Log 42,3 %).

Allerdings wird schnell klar, dass der Datensatz nicht unseren Vorstellungen enstpricht. Gayo-Avellos und unsere Ansichten unterscheiden sich in einem wichtigen Punkt: Liegt zwischen zwei gleich lautenden Anfragen ein gewisser zeitlicher Abstand (Gayo-Avello wählt 24 Stunden [Gay09]), so werden diese von ihm getrennt. Wir hingegen sind der Meinung, dass zwei aufeinanderfolgende Anfragen unabhängig von ihrem zeitlichen Abstand ganz klar zu derselben Sitzung gehören, wenn ein inhaltlicher Zusammenhang besteht, und fassen diese daher zusammen. Beispielsweise könnte eine Recherche am Freitagabend enden, weil der Nutzer Feierabend hat, und am Montagmorgen fortgesetzt werden. Nur weil zwei Tage zwischen beiden Anfragen liegen, sollte man unserer Meinung nach nicht kategorisch einen Sitzungszusammenhang ausschließen.

Neben diesen unterschiedlichen Ansichten weist Gayo-Avellos Gold-Standard einige Schwächen auf:

- Auch hier fehlen die Zeitstempel. Diese können zwar rekonstruiert werden, doch stellt sich die Frage, warum diese überhaupt entfernt wurden, zumal sie eine wichtige Rolle bei der Entscheidungsfindung der geometrischen Methode spielen.

- Die Reihenfolge der Anfragen wurde in einigen Fällen nicht vom AOL-Log übernommen. In dieser Arbeit soll allerdings das originale Anfrageverhalten der Nutzer untersucht werden.

- Es sind 19 Nutzer enthalten, die nur eine Anfrage stellten. Eine Sitzungserkennung für diese Nutzer ist schlichtweg nicht sinnvoll.

- URL-Anfragen blieben, wie auch bei Lucchese et al., erhalten. Wie bereits erwähnt, gehen wir davon aus, dass es sich lediglich um eine Verwechslung des Suchfeldes mit der Adressleiste handelt, und sind daher der Meinung, dass diese Anfragen ignoriert werden sollten.

- 154 Anfragen von 48 Nutzern wurden ohne erkennbaren Grund entfernt. Uns ist aber eine hohe Transparenz hinsichtlich der Entstehung des Datensatzes sehr wichtig.

Darüber hinaus haben wir eine Reihe an inhaltlichen Fehlentscheidungen festgestellt. Beispiele hierfür können der Tabelle 3.2 entnommen werden: Im ersten Auszug wurde fäschlicherweise für eine neue Suchsitzung gestimmt, obwohl es sich in beiden Fällen um Bars in Minneapolis handelt und nur wenig Zeit zwischen den Anfragen liegt. Wir können also mit großer Sicherheit davon ausgehen, dass hier jemand seinen Abend geplant hat und würden daher

ID	Anfrage	Zeit	URL
258919	`joe senser's sports bar`	2006-03-30 20:43:10	
258919	`joe senser's sports bar`	2006-03-30 20:43:44	http://www.findit.com
258919	`dustys bar mpls`	2006-03-30 20:46:23	http://travel.yahoo.com
258919	`dustys bar mpls`	2006-03-30 20:53:16	

ID	Anfrage	Zeit	URL
1379196	`julius caesar`	2006-03-05 12:07:18	http://www.perseus.tufts.edu
1379196	`julius caesar`	2006-03-05 12:07:18	http://www.perseus.tufts.edu
1379196	`weather`	2006-03-05 12:33:22	http://www.weather.com
1379196	`julius casar`	2006-03-05 12:36:00	
1379196	`julius caesar`	2006-03-05 12:36:16	http://www.lausd.k12.ca.us

Tabelle 3.2: Fehler im Gold-Standard von Gayo-Avello: Der obere Auszug enthält eine falsch gesetzte Sitzungsrenze,[4]der untere zwei falsche Sitzungsfortführungen.

die beiden Anfragen derselben Sitzung zuordnen. Im zweiten Auszug hingegen wurde die Anfrage `weather` in dieselbe Sitzung der umgebenden `julius caesar` Anfragen eingeordnet, obwohl eindeutig kein inhaltlicher Zusammenhang besteht. In dem gezeigten Auszug sollten also eigentlich zwei Sitzungsgrenzen enthalten sein, die hier aber fehlen.

Zwar verfügt Gayo-Avellos Gold-Standard über eine sehr gute Grundsubstanz, enthält unserer Meinung nach aber zu viele Fehler (auch wegen unserer unterschiedlichen Ansichten bei gleichlautenden Anfragen, zwischen denen mehr als 24 Stunden vergangen sind), die in einem Gold-Standard nicht enthalten sein sollten. Immerhin wird die Güte eines Verfahrens zur Suchsitzungserkennung hieran gemessen und der Gold-Standard sollte daher möglichst fehlerfrei sein.

3.3 Entstehung des neuen Gold-Standards

Da wir aus den oben genannten Gründen keinen der beiden vorliegenden Datensätze zur Evaluation verwenden wollen, entscheiden wir uns dafür, einen eigenen Gold-Standard zu erstellen. Zunächst muss entschieden werden, ob ein komplett neuer Datensatz entstehen oder einer der vorliegenden überarbeitet werden soll. Da der Gold-Standard von Gayo-Avello, wie bereits erwähnt, über eine sehr gute Grundsubstanz verfügt, entscheiden wir uns für die zweite Variante.

[4]In diesem Auszug war ursprünglich eine URL-Anfrage (`www.joesenser.com`) enthalten, die hier entfernt wurde, weil sie von der falsch gesetzten Sitzungsgrenze ablenken würde.

http	https	www	.com	␣com (mit voranstehendem Leerzeichen)
.org	.us	.co.uk	.net	.edu
.gov	.biz	.htm	.html	20+*Buchstabe* (Kodierungs-Artefakt)

Tabelle 3.3: Erweiterbare Liste mit Hinweisen auf URL-Anfragen.

Im Folgenden wird erläutert, nach welchen Regeln der neue Gold-Standard bearbeitet wird, und anschließend werden einige statistische Eigenschaften beschrieben.

Regeln zur Erstellung

Zur Überarbeitung des Datensatzes werden im ersten Schritt alle Nutzer-IDs aus Gayo-Avellos Gold-Standard herausgeschrieben, die anschließend verwendet werden, um *alle* Anfragen für die entsprechenden Nutzer aus dem Original-AOL-Log zu extrahieren.

Anfragen, die nur aus „`-`“ bestehen, fassen wir als fehlende Information im AOL-Log auf und entfernen diese daher im nächsten Schritt. Ebenso werden alle Anfragen verworfen, die nur aus einem Punkt bestehen oder im weitesten Sinne Ähnlichkeit zu einer URL aufweisen. Tabelle 3.3 enthält alle Zeichenketten, deren Vorkommen eine Anfrage unserer Ansicht nach als URL-Anfrage klassifiziert. Ein Großteil der URL-Anfragen kann anhand der Einträge in der Tabelle bereits entfernt werden, jedoch bleiben Anfragen wie `myspacecom` enthalten. Diese werden in einem weiteren Durchlauf manuell herausgefiltert, da der Punkt vermutlich vor der Veröffentlichung des AOL-Logs entfernt wurde,[5] es sich im Kern aber nach wie vor um eine URL-Anfrage handelt.

Zusätzlich zu den URL-Anfragen existieren einige Sonderfälle, die ebenfalls aus dem Log herausgefiltert werden. Darunter fällt z. B. die Anfrage `type search words keywords or web addresses here`. Hier handelt es sich um einen voreingetragenen Standardtext im Suchfeld und somit nicht um eine Anfrage im herkömmlichen Sinne.

In einem letzten Schritt werden alle Nutzer mit drei oder weniger Anfragen entfernt, da eine Suchsitzungserkennung hier nur wenig Sinn ergibt.

Nach dem Ausführen der geschilderten Schritte bleiben 8840 Anfragen von 127 verschiedenen Nutzern übrig, die nun manuell in Sitzungen eingeteilt werden müssen. Hierbei wird stets darauf geachtet, möglichst nur den Anfragetext selbst zu betrachten, da Zeitstempel und geklickte URL unter Umständen übereilte – falsche – Rückschlüsse zulassen. Diese Felder werden daher nur bei Anfragepaaren betrachtet, für die eine Entscheidung über Sitzungsfortsetzung

[5] „[...] case shifted with most punctuation removed.“, Quelle: `http://search-logs.com/aol_readme.txt`, Letzter Zugriff: 27. März 2012

oder -unterbrechung sehr schwierig ist. Im Zweifelsfalle entscheiden wir uns immer für eine neue Suchsitzung, da wir der Meinung sind, dass ein nicht erkannter Sitzungszusammenhang nicht so schwerwiegend ist wie eine falsche Sitzungsfortführung: Wenn die automatische Sitzungserkennung keinen Zusammenhang zwischen zwei Anfragen herstellen kann, würde der Nutzer lediglich mit jenen Suchergebnissen versorgt, die er ohnehin, d. h. ohne Anwendung der Sitzungserkennung, bekommen würde. Hingegen würde ein Nutzer bei einer falschen Sitzungsfortführung Suchergebnisse angezeigt bekommen, die u. U. überhaupt nicht der gestellten Anfrage entsprechen.

Während der manuellen Bearbeitung wird zusätzlich jede Anfrage mittels der *Missions-ID* einer Mission zugeordnet, um später Aussagen darüber treffen zu können, ob eine frühere Sitzung erneut aufgenommen wurde. Welche ausformulierten Missionen hinter den Missions-IDs stehen, kann man Anhang A.1 exemplarisch für den Nutzer #1936169 entnehmen.

Doch auch wenn man versucht, klare Regeln zur Bearbeitung zu erstellen und einzuhalten, entstehen immer wieder Sonderfälle, in denen eine Entscheidung sehr schwierig ist. Einige dieser Sonderfälle sind im Anhang A.2 aufgelistet, die komplette Liste kann der dem Datensatz beigefügten ReadMe-Datei entnommen werden.

Statistische Auswertung des Datensatzes

Der Ausgangsdatensatz beinhaltete 11 932 Anfragen von 215 Nutzern, wovon während der Vorverarbeitung 3092 Anfragen und 88 Nutzer entfernt wurden (35 Nutzer stellten von vornherein nur drei oder weniger Anfragen und bei 53 weiteren Nutzern blieben nur noch drei oder weniger Anfragen übrig, nachdem alle URL-Anfragen entfernt wurden). In den verbleibenden 8840 Anfragen von 127 Nutzern sind 2279 Sitzungsgrenzen markiert. Somit besteht eine Sitzung durchschnittlich aus 3,88 Anfragen, wohingegen Gayo-Avellos Gold-Standard nur 2,84 Anfragen je Sitzung zählt (Gayo-Avello hat in 11 484 Anfragen von 215 Nutzern 4039 Sitzungsgrenzen markiert). Keiner der beiden Werte ist richtig oder falsch, sie dienen lediglich zur Veranschaulichung der beiden Datensätze.

Die Anfrage-je-Nutzer-Verteilung folgt im Wesentlichen einer umgekehrt logarithmischen Funktion und kann der Abbildung 3.1 entnommen werden: Es gibt wenige Nutzer mit sehr vielen Anfragen und verhältnismäßig viele Nutzer mit wenigen Anfragen.

Betrachten wir nur die 8753 Anfragen, die in beiden Datensätzen enthalten sind, zählen wir für unseren Gold-Standard 2257 und für den von Gayo-Avello 2867 Sitzungsgrenzen, von denen 2175 exakt übereinstimmen. Je nach Interpretation erreichen wir also eine Deckungsgenauigkeit von 75,9 % $\left(\frac{2175}{2867}\right)$

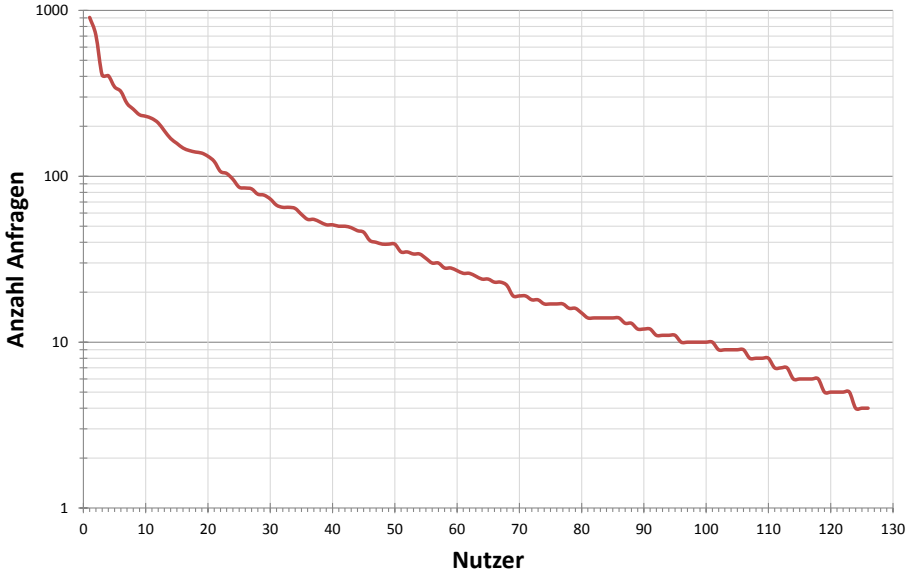


Abbildung 3.1: Anzahl der Anfragen für jeden Nutzer.

mit dem Gold-Standard von Gayo-Avello bzw. arbeiten mit einer Genauigkeit von 96,4 % $\left(\frac{2175}{2257}\right)$, da gerade einmal 82 der von uns gesetzten Sitzungsgrenzen nicht mit denen von Gayo-Avello übereinstimmen. Darüber hinaus können wir feststellen, dass wir an über 500 Stellen Sitzungen zusammengefasst haben, an denen Gayo-Avello getrennt hat. Da es sich gerade hier um eine kritische Stelle handelt (zur Erinnerung: eine falsche Sitzungsfortführung ist schwerwiegender als ein nicht erkannter Zusammenhang), prüfen wir diese Fälle besonders aufmerksam. Die meisten dieser Unstimmigkeiten können aber als Fehler in Gayo-Avellos Gold-Standard verstanden werden, die wir lediglich nach unseren Vorstellungen ausgebessert haben. Der Großteil von Gayo-Avellos Fehlentscheidungen resultiert nämlich aus der unterschiedlichen Sichtweise bei gleich lautenden Anfragen, zwischen denen aber viel Zeit vergangen ist. In den restlichen Fällen sind wir der Meinung, dass wir die richtige Entscheidung getroffen haben und Gayo-Avello den Zusammenhang nicht erkannt hat. Demgegenüber stehen etwa 70 Fälle, in denen wir zwei aufeinander folgende Anfragen zu verschiedenen Sitzungen zuordnen und Gayo-Avello diese zusammenfasst. Hier handelt es sich durchweg um Fälle, in denen wir uns auch nach sorgfältiger Prüfung unsicher waren, ob ein Zusammenhang besteht oder nicht. Da Gayo-Avello für solche schwierigen Entscheidungen keine Begründungen geliefert hat, an denen man sich orientieren könnte, haben wir uns – nach oben genannter Regel – im Zweifelsfalle für eine neue Sitzung entschieden.

Der neue Gold-Standard enthält 1397 verschiedene Missionen, damit ent-

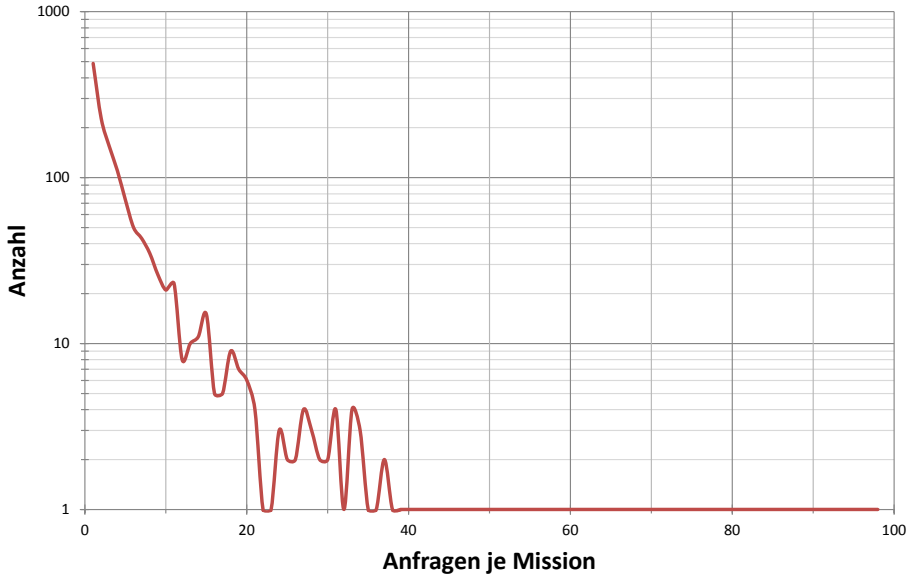


Abbildung 3.2: Häufigkeitsverteilung der Anfragen je Mission. Gezeigt wird nur der kennzeichnende Ausschnitt des tatsächlichen Verlaufs, das Maximum liegt eigentlich bei 251 Anfragen je Mission.

fallen innerhalb der drei aufgezeichneten Monate exakt 11 Missionen auf jeden Nutzer. Eine Mission enthält durchschnittlich 1,63 Sitzungen und somit etwa 5,80 Anfragen. In Abbildung 3.2 kann man erkennen, dass auch die Anfragen-je-Mission-Verteilung stark an eine umgekehrt logarithmische Funktion erinnert: Viele Missionen enthalten nur wenige Anfragen (so enthalten z. B. 488 Missionen nur eine Anfrage) und nur ganz wenige Missionen enthalten sehr viele Anfragen (eine einzige Mission enthält 251 Anfragen).

Im Mittel erstreckt sich eine Mission über 7,66 Tage. Zwar erscheint dieser Zeitraum recht groß, jedoch werden 1065 Missionen noch am selben Tag abgeschlossen. Lediglich die Anfragen der 332 verbleibenden Missionen verteilen sich weitestgehend gleichmäßig über einen Zeitraum von 2 bis 90 Tagen. In dem von uns entwickelten Gold-Standard kehrten 80 Nutzer insgesamt 1009 Mal zu einer früheren Mission zurück und beweisen damit, dass eine Missionserkennung durchaus sinnvoll ist.

3.4 Evaluation der Ergebnisse

Um die Güte der Ergebnisse von Verfahren zur Suchsitzungserkennung geeignet beschreiben zu können, werden zuerst folgende Variablen (in Anlehnung

an [HGH02]) festgelegt:

- N_{shift}: Anzahl der vom Verfahren gesetzten Sitzungsgrenzen
- N_{true_shift}: Anzahl der Sitzungsgrenzen im Gold-Standard
- $N_{shift\&correct}$: Anzahl übereinstimmender Sitzungsgrenzen
- N_{cont}: Anzahl der vom Verfahren gesetzten Sitzungsfortführungen
- N_{true_cont}: Anzahl der Sitzungsfortführungen im Gold-Standard
- $N_{cont\&correct}$: Anzahl übereinstimmender Sitzungsfortführungen

Um nun ein Maß für die Genauigkeit unserer Ergebnisse zu bestimmen, konzentrieren wir uns auf einige weitverbreitete Maßzahlen des Information Retrievals [MRS08]. Bei den ersten beiden handelt es sich um *precision* und *recall*, die (auf unser Problem übertragen) folgendermaßen errechnet werden:

$$\begin{aligned} precision &= \frac{N_{shift\&correct}}{N_{shift}} \\ recall &= \frac{N_{shift\&correct}}{N_{true\_shift}} \end{aligned}$$

Aus diesen beiden Werten kann das sogenannte *F-Measure* berechnet werden, was eine gewichtete Kombination aus *precision* und *recall* darstellt:

$$F - Measure_{\beta} = \frac{(1 + \beta^2) \cdot precision \cdot recall}{\beta^2 \cdot precision + recall}$$

Trifft unser Verfahren eine falsche Entscheidung, kann es sich nur um einen von zwei möglichen Fehlern handeln: Zum einen kann sich das Verfahren für eine Sitzungsgrenze zwischen zwei Anfragen entscheiden, die aber eigentlich zur selben Sitzung gehören (im Folgenden als *Typ-A-Fehler* bezeichnet); zum anderen kann eine Sitzung fortgeführt werden, obwohl beide Anfragen nichts miteinander zu tun haben (*Typ-B-Fehler*). Da wir Fehler vom Typ B als schwerwiegender erachten als Fehler vom Typ A (siehe Abschnitt 3.3), müssen wir für β einen Wert größer 1 bestimmen, damit diese Fehler stärker ins Gewicht fallen und das F-Measure bei häufigem Auftreten von Typ-B-Fehlern sinkt. Wir orientieren uns an He et al. und Gayo-Avello [HGH02, Gay09], die $\beta = 1,5$ setzen.

Tino Rüb führte in seiner Bachelorarbeit ein weiteres Maß für die Genauigkeit (engl. *accuracy*) für die beiden Fälle Sitzungsgrenze und -fortführung ein, nämlich ACC_{shift} und ACC_{cont} [Rüb11]. Er beschreibt damit das Verhältnis korrekter Entscheidungen in Bezug auf den *Gold-Standard*, welches aber unserer Meinung nach schon ausreichend durch die bereits vorgestellten Gütemaße beschrieben wird. Daher möchten wir das Verhältnis korrekter Entscheidungen in Bezug auf die Anzahl der Entscheidungen vom *Verfahren* bestimmen. Das heißt, es wird ermittelt, wie viele der Entscheidungen, die das Programm überhaupt getroffen hat, eigentlich korrekt waren. Wir bestimmen formal:

$$
\begin{aligned}
ACC_{cont} &= \frac{N_{cont\&correct}}{N_{cont}} \\
ACC_{shift} &= \frac{N_{shift\&correct}}{N_{shift}} \\
ACC_{avg} &= \frac{N_{true\_shift} \cdot ACC_{shift} + N_{true\_cont} \cdot ACC_{cont}}{N_{true\_shift} + N_{true\_cont}}
\end{aligned}
$$

ACC_{cont} beschreibt also prozentual, in wie vielen Fällen es sich laut Gold-Standard tatsächlich um eine Sitzungsfortführung handelt, wenn sich das Verfahren dafür entscheidet. Äquivalent hierzu gibt ACC_{shift} ein Verhältnis von korrekt gesetzten Sitzungsgrenzen zu allen gesetzten Sitzungsgrenzen an, was der *precision* entspricht. Die durchschnittliche Genauigkeit (ACC_{avg}) normiert diese beiden Werte auf die Anzahl der im Gold-Standard vorhandenen Sitzungsfortführungen bzw. Sitzungsgrenzen und bildet somit einen Aussagewert darüber, wie viele Entscheidungen insgesamt korrekt sind.

Diese Maßzahlen erleichtern zwar den Vergleich mit anderen Vorschlägen zur Suchsitzungserkennung auf demselben Gold-Standard, jedoch ist es nicht immer einfach, sich vorzustellen, was diese Zahlen überhaupt aussagen. Absolute Zahlen können hier ein besseres Vorstellungsbild vermitteln, weshalb wir in dieser Arbeit bei allen Ergebnissen zusätzlich eine kleine Tabelle angeben, die vier ebenfalls aus dem Information Retrieval bekannte Zahlen enthält, nämlich *true-positive* (TP), *true-negative* (TN), *false-positive* (FP) und *false-negative* (FN). Diese Zahlen beschreiben, wie häufig ein Verfahren zur Suchsitzungserkennung welche Entscheidung getroffen hat und wie häufig diese Entscheidung korrekt war. Tabelle 3.4 zeigt zunächst den Aufbau dieser Tabelle, der anschließend erläutert wird.

Da die beiden Standard-Gütemaße *precision* und *recall* auf Basis von Sitzungsgrenzen errechnet werden, wird eine Entscheidung für eine Sitzungsgrenze als *positive* angesehen. Ist diese Entscheidung korrekt, also *true*, so wird die

		SG im Gold-Standard	
		ja	nein
Für SG gestimmt	ja	TP	FP
	nein	FN	TN

Tabelle 3.4: Tabelle zur Fehleranalyse. Die Abkürzung SG steht für Sitzungsgrenze.

Zählvariable für *true-positive* um 1 erhöht. In der nachfolgenden Aufzählung wird für jeden der vier Werte beschrieben, welche Situation gegeben ist und was dies bedeutet.

- *true-positive*:
  - Kaskade: Sitzungsgrenze
  - Gold-Standard: Sitzungsgrenze

    → Sitzungsgrenze wurde richtig erkannt
- *false-positive*:
  - Kaskade: Sitzungsgrenze
  - Gold-Standard: Sitzungsfortführung

    → Zusammenhang zwischen beiden Anfragen wurde nicht erkannt (Typ-A-Fehler)
- *true-negative*:
  - Kaskade: Sitzungsfortführung
  - Gold-Standard: Sitzungsfortführung

    → Zusammenhang zwischen beiden Anfragen wurde korrekt erkannt
- *false-negative*:
  - Kaskade: Sitzungsfortführung
  - Gold-Standard: Sitzungsgrenze

    → es wurde ein falscher Zusammenhang erkannt (**Typ-B-Fehler!**)

Kapitel 4

Suchsitzungserkennung

In diesem Kapitel werden die implementierten Techniken zur Erkennung von Suchsitzungen und ihre erzielten Ergebnisse vorgestellt. Um die optimale Konfiguration für die Parameter der einzelnen Schritte zu ermitteln, wird ein Viertel des von uns erstellten Gold-Standards als sogenannter Trainingsdatensatz verwendet. Dieser besteht aus exakt 25 % der Anfragen (also $0{,}25 \cdot 8840 = 2210$ Stück) von 25 % aller Nutzer ($0{,}25 \cdot 127 = 31{,}75 \rightarrow 32$ Nutzer). Alle in diesem Kapitel genannten Ergebnisse beziehen sich ausschließlich auf diesen Trainingsdatensatz. Zur Evaluierung der ermittelten Parameter testen wir das Verfahren auf den verbleibenden Anfragen, dem sogenannten Testdatensatz, bestehend aus 6630 Anfragen von 95 Nutzern. Die Ergebnisse auf dem Testdatensatz werden in Kapitel 6 präsentiert. Neben dem vollständigen Gold-Standard werden auch der Trainings- und Testdatensatz zur freien Verfügung online gestellt.[1]

4.1 Aufbau eines Suchanfrage-Logs

Ein Suchanfrage-Log für einen Nutzer besteht aus n Interaktionen mit der Suchmaschine. Wir bezeichnen diese als $a_1, a_2, \ldots, a_i, \ldots, a_n$. Jede dieser Aktionen beinhaltet neben der Nutzer-ID die gestellte Anfrage, den exakten Zeitpunkt und, falls der Nutzer auf ein Suchergebnis geklickt hat, dessen Domain und Rang auf der Ergebnisseite. Wir haben zwar auch damit experimentiert, die angeklickten Domains in die Suchsitzungserkennung miteinzubeziehen, erzielten aber keine verwertbaren Ergebnisse, weshalb wir in unserem Verfahren nur den Anfragetext q und den Zeitstempel t betrachten. Eine Aktion a_i lässt sich für unsere Zwecke also als folgendes Tupel darstellen: $a_i = \langle q_i, t_i \rangle$. Alle Aktionen eines Nutzers sind nach ihrem Zeitstempel sortiert, also gilt $t_{i-1} \leq t_i$ für alle a_i. Ähnlich zu einer Sitzung definieren wir eine Mission m, wobei eine

[1] http://www.webis.de/research/corpora

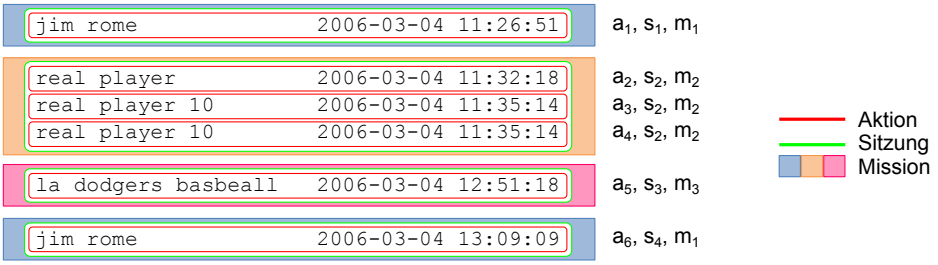


Abbildung 4.1: Struktur eines Anfrage-Logs: Eine Mission kann mehrere Sitzungen enthalten (hier enthält die Mission m_1 die Sitzungen s_1 und s_4), die wiederum aus mehreren Aktionen bestehen kann (wie in s_2). Eine Aktion ist ein Tupel aus Anfragetext und Zeitstempel.

Mission eine oder mehrere Sitzungen enthalten kann. Allerdings unterscheiden sich Sitzungen und Missionen in einem wichtigen Punkt: Während alle Anfragen in einer Sitzung direkt aufeinander folgen *müssen* (d. h. wenn a_1 und a_4 zur selben Sitzung gehören, so gehören zwangsläufig auch a_2 und a_3 dazu), können zwischen den Sitzungen einer Mission beliebig viele andere Sitzungen liegen (beispielsweise wäre $m_1 = \{s_1, s_7, s_{10}\}$ gültig). Abbildung 4.1 zeigt einen Auszug aus dem Gold-Standard, in dem alle Aktionen, Sitzungen und Missionen markiert sind.

4.2 Anordnung der Schritte in einer Kaskade

Wie bereits erwähnt, gibt es im Kern zwei Anwendungsfälle für die Suchsitzungserkennung: den Offline- und den Online-Anwendungsfall.

Im Offline-Anwendungsfall teilt man einen bereits gespeicherten Suchanfrage-Log in Sitzungen ein. Die benötigte Zeit für die Erfüllung dieser Aufgabe spielt hierbei eine untergeordnete Rolle; hier kommt es vor allem auf die Genauigkeit an.

Im Online-Anwendungsfall soll der Nutzer schon während seiner Interaktion mit der Suchmaschine mit kontextabhängigen Ergebnissen versorgt werden, das heißt man muss schon während der Bearbeitung einer Suchanfrage feststellen, ob sie zur letzten Sitzung gehört oder nicht.

Beispielsweise benötigt die Erweiterung einer Suchanfrage durch Suchergebnisse etwa eine halbe Sekunde je Anfragepaar, was im Vergleich zu anderen Verfahren zur Sitzungserkennung sehr langsam ist. Im Online-Anwendungsfall ist eine Wartezeit von einer halben Sekunde aber nicht vertretbar, weshalb man hier bevorzugt auf die Anfrage-Erweiterung verzichtet, da diese prozentual betrachtet ohnehin nur wenige Zusammenhänge erkennt. Im Offline-

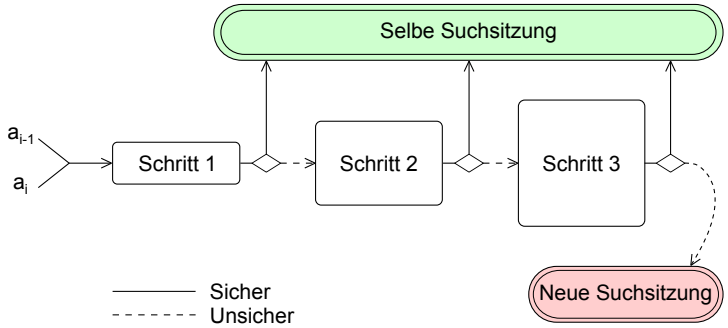


Abbildung 4.2: Die Idee hinter der Kaskade: Kann ein schnelles Verfahren bereits einen Zusammenhang erkennen, so müssen die zeitaufwändigeren Schritte nicht mehr ausgeführt werden.

Anwendungsfall hingegen, bei dem es darauf ankommt, möglichst viele Zusammenhänge zu erkennen, können diese wenigen erkannten Zusammenhänge unter deutlicher Zunahme der Laufzeit die Gesamtgüte der Ergebnisse aber noch verbessern.

Wir betrachten in unserer Untersuchung also für jeden Schritt der Kaskade neben der Güte seiner Entscheidungen auch die Laufzeit je untersuchtem Anfragepaar. Anhand dieses Wertes ordnen wir die verschiedenen Techniken nach dem Vorschlag von Hagen et al. in einer Kaskade an [HSR11], sodass wenig zeitaufwändige Schritte vor solchen ausgeführt werden, die mehr Zeit benötigen, um eine Entscheidung zu treffen. Wird ein Zusammenhang für ein Anfragepaar erkannt, so wird das Ausführen der nachfolgenden Schritte obsolet. Somit kann bei bestmöglicher Güte immer eine minimale Laufzeit garantiert werden. Abbildung 4.2 stellt das abstrahierte Grundgerüst der Kaskade dar. In dieser Abbildung beschreibt die Größe jedes Rechtecks die Komplexität eines Schrittes, d. h. je mehr Zeit ein Schritt benötigt, umso größer ist das Rechteck. Kann auch der letzte Schritt keine klare Aussage darüber treffen, ob zwei Anfragen zur selben Sitzung zugeordnet werden sollten, entscheidet sich unser Verfahren für eine neue Suchsitzung (gestrichelte Linie).

In den folgenden Abschnitten werden wir Schritt für Schritt die Kaskade von Hagen et al. rekonstruieren [HSR11], einige Verbesserungen vornehmen und einen neuen Schritt, basierend auf dem *Linked-Open-Data*-Ansatz von Hollink et al. [HTV11], in die Kaskade aufnehmen. Wir werden die in den einzelnen Schritten verwendeten Techniken beschreiben sowie Aussagen zur Güte und Laufzeit treffen und auf Schwächen hinweisen.

4.3 Lexikalische Ähnlichkeit

Bei einem großen Teil der Anfragen im AOL-Log handelt es sich um reine Wiederholungen der vorausgegangenen Anfrage. Für den Trainingsdatensatz mit 2210 Anfragen ermitteln wir 1004 Wiederholungen; es können also 45,4 % aller Anfragen ohne großen Aufwand zur selben Sitzung zugeordnet werden. Da die Wahrscheinlichkeit äußerst gering ist, dass ein Nutzer zweimal die gleiche Anfrage direkt hintereinander stellt, dabei aber verschiedene Intentionen verfolgt, gehen wir davon aus, dass Fehler für das Zusammenfassen aufgrund von Anfragewiederholungen nahezu ausgeschlossen sind.

Aufgrund ihrer großen Ähnlichkeit zu dem in Abschnitt 2.1 beschriebenen Suchverhaltensmuster *Wiederholung* wird auch geprüft, ob einer der beiden Fälle *Generalisierung* oder *Spezialisierung* vorliegt, bei deren Auftreten beide Anfragen zu einer Sitzung zusammengefasst werden [HSR11]. Formal können diese drei Muster für zwei konsekutive Aktionen a_{i-1} und a_i folgendermaßen dargestellt werden:

$$\begin{aligned} q_{i-1} = q_i &\quad\rightarrow\quad \text{Wiederholung} \\ q_{i-1} \subset q_i &\quad\rightarrow\quad \text{Spezialisierung} \\ q_{i-1} \supset q_i &\quad\rightarrow\quad \text{Generalisierung} \end{aligned}$$

Grundsätzlich stehen für die Bestimmung der lexikalischen Ähnlichkeit zweier Anfragen verschiedene Techniken, wie z. B. die Ermittlung der Levenshtein-Distanz oder des Jaccard-Koeffizienten, zur Verfügung. Diese und andere Techniken wurden von Tino Rüb gegenübergestellt und evaluiert, wobei er die Überprüfung beider Anfragen auf Termgleichheit als zuverlässiges und dabei schnellstes Mittel erachtet [Rüb11]. Für den Termvergleich werden beide Anfragetexte in ihre jeweiligen Terme zerlegt und anschließend geprüft, ob entweder alle Terme in beiden Anfragen enthalten sind (Wiederholung) oder zumindest eine Anfrage *alle* Terme der anderen enthält (Generalisierung oder Spezialisierung). Aufgrund der Untersuchungen von Tino Rüb verwenden Hagen et al. diese Technik als ersten Schritt ihrer Kaskade [HSR11].

Wir schlagen stattdessen vor, einen simplen String-Vergleich zwischen beiden Anfragetexten auszuführen und erreichen damit eine enorme Laufzeitverbesserung ($0,0009$ ms je Anfragepaar gegenüber $0,0133$ ms). Allerdings entstehen für diese Technik Fehler bei sehr kurzen Anfragen. Beispielsweise werden die beiden Anfragen `t` und `pharmaceutical companies` zusammengefasst, weil die zweite Anfrage die erste komplett enthält (nämlich den Buchstaben `t` im Wort `pharmaceutical`) und somit als Spezialisierung gewertet wird. Da der Performance-Vorteil des String-Vergleiches aber deutlich überwiegt und wir daher bevorzugt diesen verwenden wollen, führen wir den String-Vergleich

String-Vergleich		SG im Gold-Standard	
		ja	nein
Für SG gestimmt	ja	555	510
	nein	0	1113

Term-Vergleich		SG im Gold-Standard	
		ja	nein
Für SG gestimmt	ja	555	497
	nein	0	1126

Tabelle 4.1: Fehleranalyse für die beiden Techniken String- und Term-Vergleich.

nur dann aus, wenn beide Anfragen aus mindestens drei Buchstaben bestehen oder gleich lang sind. Für zu kurze Anfragen wird der String-Vergleich nicht ausgeführt und der erste Schritt kann keinen Zusammenhang finden.

In Tabelle 4.1 sind die Entscheidungen beider Techniken in Form einer Fehleranalyse-Tabelle aufgeführt. Unser Hauptaugenmerk liegt auf der rechten Seite der Tabelle und hier vor allem auf dem unteren Feld, da die *true-negatives* beschreiben, wie viele Sitzungsfortführungen korrekt erkannt wurden. Wie man sieht, kann die Überprüfung auf Term-Gleichheit 13 Zusammenhänge mehr erkennen als der simple String-Vergleich (1126 gegen 1113), was z. B. daran liegen kann, dass eine Suchanfrage zwar alle Terme der vorausgegangenen Anfrage enthält, aber in einer anderen Reihenfolge (beispielsweise `brazil 2006 world cup team` und `world cup 2006 brazil team`).

Allerdings erscheint der Vorteil, solche Fälle zu erkennen, angesichts der insgesamt erkannten Sitzungsfortführungen und der 12,5-fachen Laufzeit ver-

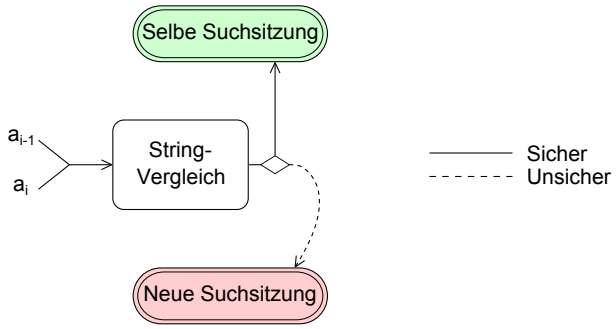


Abbildung 4.3: Die Kaskade nach dem ersten Schritt.

precision	*recall*	*F*-Measure	ACC_{cont}	ACC_{shift}	ACC_{avg}	Laufzeit (ms)
0,5211	1,0000	0,7796	1,0000	0,5211	0,8406	0,0009

		SG im Gold-Standard	
		ja	nein
Für SG gestimmt	ja	555	510
	nein	0	1113

Tabelle 4.2: Ergebnisse der Kaskade nach dem ersten Schritt.

schwindend gering. Da davon auszugehen ist, dass diese 13 Zusammenhänge auch von einem der nächsten Schritte (unter entsprechend größerem Zeitaufwand) erkannt werden können, entscheiden wir uns aufgrund seiner äußerst schnellen Laufzeit, den String- statt des Term-Vergleiches (wie in [HSR11]) als ersten Schritt in unsere Kaskade aufzunehmen. Abbildung 4.3 zeigt den aktuellen Aufbau unserer Kaskade und Tabelle 4.2 die Ergebnisse, die sie erzielt.

Der erste Schritt ist zwar bei sehr niedriger Laufzeit in der Lage, einen großen Teil der Sitzungsfortführungen korrekt zu erkennen, kann allerdings schon bei einem einzelnen Rechtschreibfehler oder Buchstabendreher (aber sonst gleichlautendem Anfragetext) keinen Zusammenhang feststellen. Eine Untersuchung auf Basis von n-Grammen, wie sie im zweiten Schritt – der geometrischen Methode – Anwendung findet, kann für solche Fälle eine hohe Ähnlichkeit und somit die Zusammengehörigkeit beider Anfragen erkennen.

4.4 Geometrische Methode

Die geometrische Methode, die von Gayo-Avello vorgeschlagen wurde, bezieht neben einem lexikalischen Merkmal auch den zeitlichen Abstand beider Anfragen in die Berechnungen zur Ähnlichkeit mit ein [Gay09]. Werden zwei Anfragen nur kurz hintereinander gestellt, ist die Wahrscheinlichkeit von vornherein recht groß, dass beide demselben Informationsbedarf dienen, weshalb sich die Anfragetexte nicht sehr ähnlich sein müssen und trotzdem noch zur selben Sitzung zugeordnet werden. Natürlich verhält sich dies umgekehrt im anderen Fall: Liegen beide Anfragen weit auseinander, müssen sie beinahe identisch sein, um derselben Sitzung zugeordnet zu werden.

Hagen et al. verwenden die geometrische Methode in leicht abgewandelter Form als zweiten Schritt ihrer Kaskade [HSR11]. Zunächst wird die zeitliche Ähnlichkeit zwischen a_{i-1} und a_i mit einem zeitlichen Schwellwert von 24 Stunden folgendermaßen bestimmt:

$$f_{time} = \max\left\{0,\ 1 - \frac{t_i - t_{i-1}}{24\ \text{h}}\right\}$$

Weiter ermitteln Hagen et al. alle 3- bis 5-Gramme der zu untersuchenden Anfrage q_i und der letzten Sitzung s_j, die a_{i-1} enthält. Diese werden in Form von zwei Vektoren dargestellt, zwischen denen die Kosinusähnlichkeit berechnet wird, um so ein Maß für die lexikalische Ähnlichkeit f_{lex} beider Anfragen zu erhalten [HSR11].

Anhand dieser beiden Werte (f_{time} und f_{lex}) tragen sie einen Punkt im zweidimensionalen Raum ab und entscheiden sich für eine Sitzungsfortführung, wenn dieser Punkt außerhalb eines Einheitskreises mit dem Mittelpunkt $(0,\ 0)$ liegt, oder für eine Sitzungsgrenze, falls er innerhalb des Kreises liegt. Formal betrachtet ermitteln sie die Länge f_{geom} des Vektors vom Ursprung bis zu diesem Punkt:

$$f_{geom} \quad = \quad \sqrt{{f_{time}}^2 + {f_{lex}}^2}$$

Ist die Länge größer oder gleich 1, werden beide Anfragen derselben Sitzung zugeordnet, beträgt die Länge weniger als 1, werden sie durch eine Sitzungsgrenze getrennt.

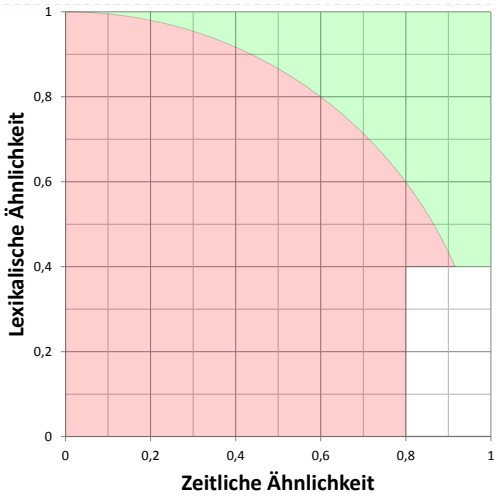
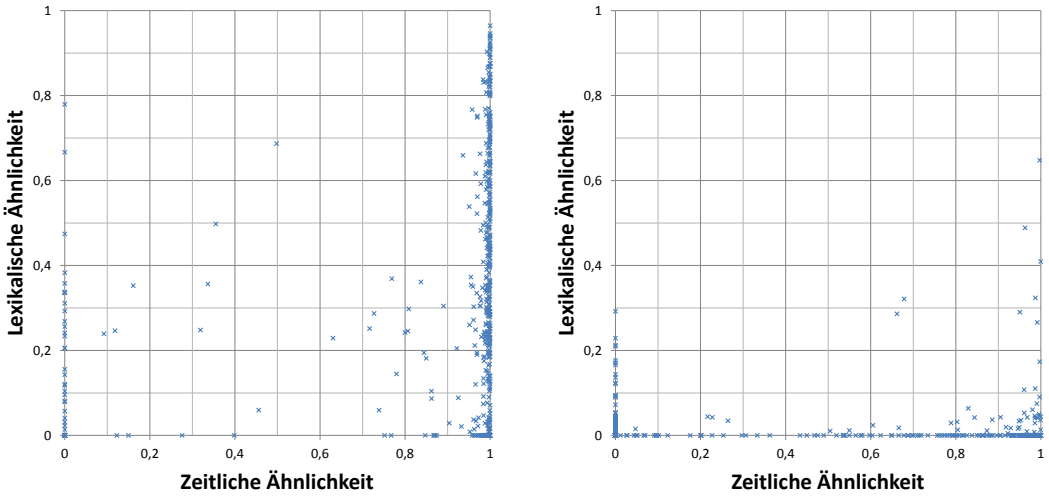


Abbildung 4.4: Grafische Darstellung der geometrischen Methode: Alle Anfragepaare, die innerhalb des roten Bereiches liegen, werden verschiedenen Sitzungen zugeordnet, während alle Anfragepaare im grünen Bereich zu einer Sitzung zusammengefasst werden. Der weiße Bereich zeigt den vorgeschlagenen Unsicherheitsbereich mit $f_{time} \geq 0,8$ und $f_{lex} \leq 0,4$ [HSR11].



(a) Sitzungsfortführung (b) Sitzungsgrenze

Abbildung 4.5: Auswertung der Verteilung nach der geometrischen Methode.

Allerdings kann man der geometrischen Methode in bestimmten Fällen nicht vertrauen. Deswegen definieren Hagen et al. einen Unsicherheitsbereich: Liegt ein Punkt innerhalb dieses Bereiches, kann die geometrische Methode keine klare Aussage über Sitzungsfortführung oder -grenze treffen, weshalb der nächste Schritt der Kaskade auf dem untersuchten Anfragepaar ausgeführt wird [HSR11]. Auch wir werden im Folgenden die Problematik unsicherer Aussagen der geometrischen Methode untersuchen und einen Unsicherheitsbereich definieren. Abbildung 4.4 stellt die modifizierte geometrische Methode von Hagen et al. grafisch dar.

Zunächst bestimmen wir auf dem Trainingsdatensatz nach der beschriebenen Herangehensweise von Hagen et al. für jedes verbleibende Anfragepaar (d. h. der erste Schritt konnte hierfür keinen Zusammenhang erkennen) die Werte f_{time} und f_{lex} und betrachten in Abbildung 4.5 die Verteilung der resultierenden Punkte für die beiden Fälle Sitzungsfortführung bzw. Sitzungsgrenze. Bei Betrachtung der Plots können wir keinen ersichtlichen Grund für die Benutzung des Einheitskreises als Entscheidungsgrundlage für eine Sitzungsfortführung bzw. -grenze erkennen. Auch hat Gayo-Avello in seiner Arbeit keine Begründung hierfür geliefert. Wir experimentieren daher auch mit anderen geometrischen Formen und erhalten für eine lineare Funktion durch die Punkte $(0,\ 1)$ und $(1,\ 0)$ etwas bessere Ergebnisse. Formal ausgedrückt soll unsere Kaskade also für eine Sitzungsfortführung stimmen, wenn $f_{time} + f_{lex}$ größer oder gleich 1 ist, und für eine neue Sitzung, wenn die Summe beider Werte darunter liegt. Da bei Benutzung dieser linearen Funktion sogar noch

drei Rechenoperationen je Anfragepaar entfallen (zwei Quadrierungen, einmal Wurzelziehen), entscheiden wir uns für den Austausch des Einheitskreises durch die lineare Funktion.

Auch der zur Berechnung der zeitlichen Ähnlichkeit verwendete Schwellwert von 24 Stunden, wie er von Gayo-Avello vorgeschlagen [Gay09] und von Hagen et al. übernommen wurde [HSR11], erscheint uns willkürlich gewählt, weshalb wir alle Werte zwischen 0 und 36 Stunden mit einer Schrittweite von einer Stunde testen. Die Optimalkonfiguration ermitteln wir für einen zeitlichen Schwellwert von 18 Stunden.

Zusätzlich experimentieren wir mit verschiedenen Werten für die Ermittlung der n- bis m-Gramme und stellen fest, dass das F-Measure für die Ergebnisse auf dem Trainingsdatensatz unter Verwendung von 3- bis 4-Grammen gegenüber 3- bis 5-Grammen leicht steigt. Ein weiterer Vorteil ist die Verkürzung der Laufzeit, da die Bestimmung der 5-Gramme herausfällt und in der Konsequenz auch die Berechnung der Kosinusähnlichkeit aufgrund der kleineren Vektoren schneller ausgeführt werden kann.

Würden wir die geometrische Methode mit dieser Konfiguration die verbleibenden Anfragepaare untersuchen lassen, ohne einen Unsicherheitsbereich zu definieren, wie er von Hagen et al. vorgeschlagen wird [HSR11], wäre das Ergebnis noch nicht zufriedenstellend, da die geometrische Methode noch zu viele

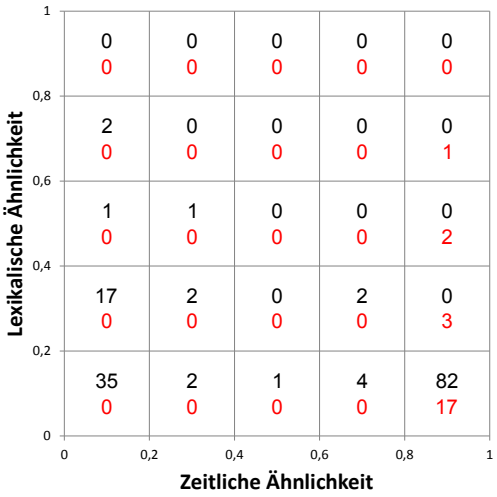


Abbildung 4.6: Fehlentscheidungen der geometrischen Methode auf dem Trainingsdatensatz (ohne Unsicherheitsbereich): Schwarz markiert sind die fehlerhaften Entscheidungen für eine Sitzungsgrenze (false-positive → Typ-A-Fehler), rot markiert sind die fehlerhaften Entscheidungen für eine Sitzungsfortführung (false-negative → Typ-B-Fehler).

Fehlentscheidungen trifft. Der Abbildung 4.6 kann man entnehmen, für welche Kombinationen von f_{time} und f_{lex} wie viele Fehlentscheidungen auftreten.

Die Fehler im linken, unteren Bereich ($f_{time} \leq 0{,}4$ und $f_{lex} \leq 0{,}4$) sind vernachlässigbar, da es sich hier ausschließlich um *false-positives* handelt, d. h. die geometrische Methode hat bei niedriger zeitlicher Ähnlichkeit ($f_{time} = 0{,}4$ entspricht etwa 11 Stunden Zeitunterschied) und niedriger lexikalischer Ähnlichkeit für eine Sitzungsgrenze gestimmt. Da diese Entscheidung aber nicht unbegründet ist und es sich ohnehin nur um vergleichsweise wenige Fehler vom Typ A (d. h. nicht so schwerwiegend) handelt, nehmen wir diese in Kauf. Den 56 Fehlern im angegebenen Bereich stehen 311 korrekte Entscheidungen gegenüber.

Die Fehler im rechten, unteren Bereich hingegen beeinflussen das F-Measure sehr stark. Neben den schwerwiegenden Typ-B-Fehlern handelt es sich bei etwa einem Drittel (82 Typ-A-Fehler gegenüber 156 korrekten Entscheidungen) der Entscheidungen für eine Sitzungsgrenze um Fehler, die sich entsprechend negativ auf das F-Measure auswirken. Daher werden auch wir einen Unsicherheitsbereich nach dem Vorschlag von Hagen et al. [HSR11] definieren und Entscheidungen der geometrischen Methode ignorieren, falls sie in diesen Bereich fallen. Indem wir auf dem Trainingsdatensatz jede mögliche Kombination der Werte f_{time} und f_{lex} (Schrittweite jeweils 0,01) durchlaufen und das resultierende F-Measure betrachten, bestimmen wir die Optimalkonfiguration für den Unsicherheitsbereich mit $f_{time} \geq 0{,}93$ (entspricht etwa 75 Minuten) und $f_{lex} \leq 0{,}12$. Durch diese drastische Verkleinerung des Unsicherheitsbereiches erreichen wir bei gleichbleibender Güte, dass unsere Kaskade häufig schon nach

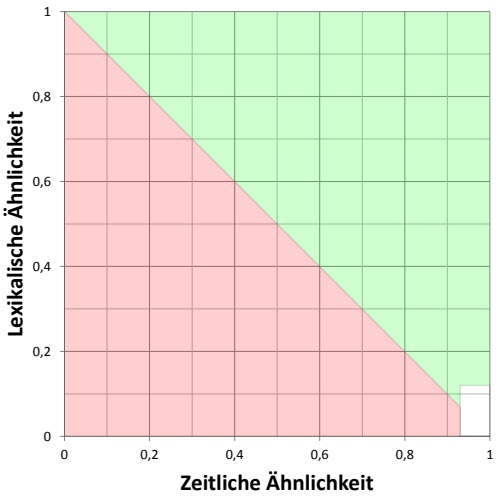


Abbildung 4.7: Unser Vorschlag zur Optimierung der geometrischen Methode.

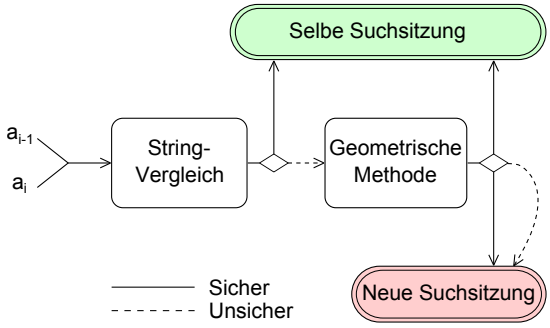


Abbildung 4.8: Die Kaskade nach dem zweiten Schritt. Im Gegensatz zum ersten Schritt kann der zweite eine sichere Entscheidung für eine Sitzungsgrenze fällen.

dem zweiten Schritt eine Entscheidung getroffen hat und keine zeitaufwändigeren Schritte mehr ausgeführt werden müssen. Abbildung 4.7 stellt die optimierte geometrische Methode grafisch dar und Abbildung 4.8 zeigt die Kaskade nach Aufnahme der geometrischen Methode als zweiten Schritt.

Die geometrische Methode als zweiter Schritt der Kaskade wird 1065 mal aufgerufen und stimmt 334 mal für eine Sitzungsfortführung (darunter sind gerade einmal fünf Fehlentscheidungen), d. h. 329 Zusammenhänge werden korrekt erkannt. Weiterhin stimmt sie 482 mal für eine Sitzungsgrenze (407 davon sind korrekt, 75 sind Fehler) und ist sich bei 249 Anfragepaaren unsicher (zur Untersuchung dieser werden also die nächsten Schritte ausgeführt). Die genauen Ergebnisse können der Tabelle 4.3 entnommen werden.

Nach dem Ausführen der ersten beiden Schritte wurde bereits eine sichere Entscheidung für 1929 von 2178 Anfragepaaren getroffen. Da die verbleibenden 249 Anfragepaare nur eine niedrige lexikalische Ähnlichkeit aufweisen ($f_{lex} \leq 0{,}12$), aber zeitlich relativ nah beeinander liegen (höchstens 75 Mi-

precision	*recall*	F-Measure	ACC_{cont}	ACC_{shift}	ACC_{avg}	Laufzeit (ms)
0,7524	0,9910	0,9029	0,9965	0,7524	0,9289	0,24

		SG im Gold-Standard	
		ja	nein
Für SG gestimmt	ja	550	181
	nein	5	1442

Tabelle 4.3: Ergebnisse der Kaskade nach dem zweiten Schritt.

nuten Zeitunterschied), werden wir versuchen, einen eventuellen Zusammenhang auf semantischer Ebene zu erkennen.

4.5 Semantische Ähnlichkeit

Die von Gabrilovich und Markovitch eingeführte *Explizite Semantische Analyse* (*ESA*) kann auf Basis der Wikipedia solche semantischen Zusammenhänge erkennen [GM07]. Hierfür wird zunächst ein Satz an Wikipedia-Artikeln (wir verwenden 100 000 zufällige Artikel) in eine $tf{\cdot}idf$-gewichtete Wortmatrix überführt, sodass für jeden Artikel ein Vektor $\overrightarrow{w_k}$ vorliegt, der das vom Artikel behandelte Thema repräsentiert. Dieser zeitaufwändige Schritt muss nur einmal durchgeführt werden, da sich an der entstandenen Matrix nichts mehr ändert. Anschließend werden die beiden Suchanfragen in denselben Vektorraum überführt und die Ähnlichkeit zu jedem der Wikipedia-Artikel mittels der Kosinusähnlichkeit bestimmt. Dadurch entsteht für jede Anfrage ein Vektor ($\overrightarrow{q_{i-1}}$ bzw. $\overrightarrow{q_i}$), der die Ähnlichkeitswerte zu jedem Wikipedia-Artikel der vorberechneten Matrix beinhaltet. Bestimmt man nun die ESA-Ähnlichkeit f_{esa} der beiden Vektoren $\overrightarrow{q_{i-1}}$ und $\overrightarrow{q_i}$ durch die Kosinusähnlichkeit, so erhält man eine Aussage darüber, wie ähnlich sich beide Anfragen in Bezug auf die Konzepte der Wikipedia-Artikel sind. Die Idee dabei ist, dass themenverwandte Suchanfragen eine hohe Ähnlichkeit zu denselben Wikipedia-Artikeln aufweisen und daher der Winkel zwischen ihren repräsentierenden Vektoren sehr klein und somit die Ähnlichkeit sehr groß ist.

Wir übernehmen die Explizite Semantische Analyse beinahe unverändert als dritten Schritt aus der Kaskade von Hagen et al. [HSR11] und bestimmen auf unserem Trainingsdatensatz einen optimalen Schwellwert von 0,33, d. h. unsere Kaskade stimmt für eine Sitzungsfortführung, wenn die ESA-Ähnlichkeit f_{esa} diesen Schwellwert übersteigt. Ist f_{esa} kleiner als 0,33, kann die

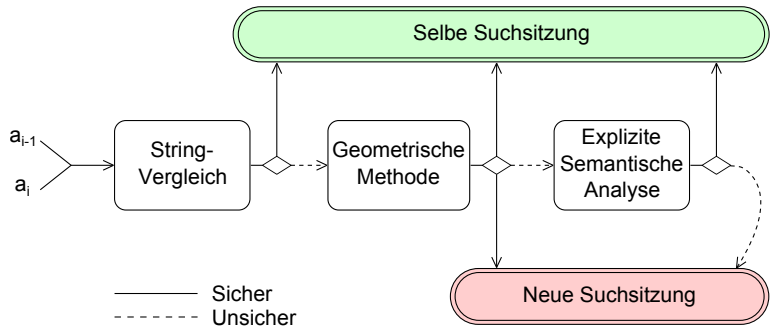


Abbildung 4.9: Die Kaskade nach dem dritten Schritt.

precision	*recall*	F-Measure	ACC_{cont}	ACC_{shift}	ACC_{avg}	Laufzeit (ms)
0,7593	0,9892	0,9049	0,9959	0,7593	0,9306	0,41

		SG im Gold-Standard	
		ja	nein
Für SG gestimmt	ja	549	174
	nein	6	1449

Tabelle 4.4: Ergebnisse der Kaskade nach dem dritten Schritt.

Explizite Semantische Analyse aber keine sichere Entscheidung für eine neue Suchsitzung fällen, da die Ergebnisse der ESA abhängig von den zugrundeliegenden Wikipedia-Artikeln sind. Abbildung 4.9 zeigt unsere Kaskade mit der Expliziten Semantischen Analyse als dritten Schritt und Tabelle 4.4 zeigt die Ergebnisse, die sie für die verbleibenden Anfragepaare auf unserem Trainingsdatensatz erzielt. Es werden lediglich acht Entscheidungen für eine Sitzungsfortführung getroffen, von denen eine falsch ist. Aufgrund seiner vertretbaren Laufzeit von 0,41 ms je Anfragepaar und dem marginal steigenden F-Measure entschließen wir uns aber dennoch, diesen Schritt beizubehalten.

Da die Explizite Semantische Analyse lediglich sieben Zusammenhänge korrekt erkennt und damit leider nur wenig zufriedenstellende Ergebnisse liefert, möchten wir einen weiteren Ansatz zur automatischen Erkennung von inhaltlichen Zusammenhängen bei niedriger lexikalischer Ähnlichkeit untersuchen.

4.6 Linked Open Data

Linked Open Data (LOD) bezeichnet ein Konzept nach der Idee von Tim Berners-Lee,[2] wonach Wissen strukturiert und für Maschinen lesbar aufbereitet wird. Dieses Wissen wird zur freien Verfügung im Internet veröffentlicht und beinhaltet Informationen über Dinge (Personen, Orte, Filme etc.) sowie Verknüpfungen zwischen diesen. Basierend auf der Idee von Hollink et al. [HTV11] werden wir diese Dinge und Verbindungen in einem Graphen organisieren und versuchen, hierüber Zusammenhänge zwischen Suchanfragen zu finden. Als Grundlage für die Erstellung dieses Graphen wird der DBpedia-Datensatz verwendet, der 3,64 Millionen Dinge[3] (sogenannte Entitäten) und Verknüpfungen zwischen diesen beinhaltet.

[2] `http://www.w3.org/DesignIssues/LinkedData.html`, Letzter Zugriff: 27. März 2012
[3] `http://dbpedia.org/About`, Letzter Zugriff: 27. März 2012

Der DBpedia-Datensatz ist in RDF-Tripeln organisiert: Jeweils das erste Element ist der Identifier der zu beschreibenden Entität; es folgt der Typ der Verknüpfung, den es beschreibt (z. B. *spouse* oder *profession*); und das letzte Element ist der Identifier der verknüpften Entität. Zusätzlich enthält jede Entität mindestens einen für Menschen lesbaren Klartext, das sogenannte Label. Der folgende Auszug aus dem DBpedia-Datensatz zeigt die beiden Label der Entität *David Beckham* sowie drei mit ihm verknüpfte Informationen (seine *Mannschaft*, seine *Ehefrau* und seine *Körpergröße*).

```
<http://dbpedia.org/resource/David_Beckham>
<http://xmlns.com/foaf/0.1/name>
"David Beckham"@en .

<http://dbpedia.org/resource/David_Beckham>
<http://xmlns.com/foaf/0.1/name>
"David Robert Joseph Beckham"@en .

<http://dbpedia.org/resource/David_Beckham>
<http://dbpedia.org/ontology/team>
<http://dbpedia.org/resource/Los_Angeles_Galaxy> .

<http://dbpedia.org/resource/David_Beckham>
<http://dbpedia.org/ontology/spouse>
<http://dbpedia.org/resource/Victoria_Beckham> .

<http://dbpedia.org/resource/David_Beckham>
<http://dbpedia.org/ontology/height>
"1.8288"^^<http://www.w3.org/2001/XMLSchema#double> .
```

Zunächst extrahieren wir alle Labels und schreiben diese als Schlüssel-Wert-Paar mit dem Identifier der Entität als Wert in einen Präfixbaum. Dieser Präfixbaum garantiert später einen sehr schnellen Zugriff, da *direkt* auf eine Gruppe von Entitäten zugegriffen werden kann, deren Label mit einem gegebenen Präfix beginnen. Dieser Präfixbaum wird exemplarisch für einige Entitäten in Abbildung 4.10 dargestellt.

Um den Graphen auf das Wesentliche zu beschränken, filtern wir alle Einträge heraus, die Entitäten oder Verküpfungsstypen enthalten, die nicht von DBpedia stammen. Im obigen Auszug wird somit die Körpergröße Beckhams entfernt, die ohnehin nicht von großem Nutzen ist, weil dieselbe Körpergröße im Allgemeinen keine nennenswerte Verbindung zwischen zwei Entitäten darstellt.

Da nur erkannt werden soll, ob zwei Entitäten miteinander verknüpft sind, aber nicht von welchem Typ diese Verknüpfung ist (wie z. B. *spouse*, siehe oben), verwerfen wir jeweils das zweite Element aller RDF-Tripel. Es werden also nur die Entität selbst und die mit ihr verknüpften Entitäten behalten.

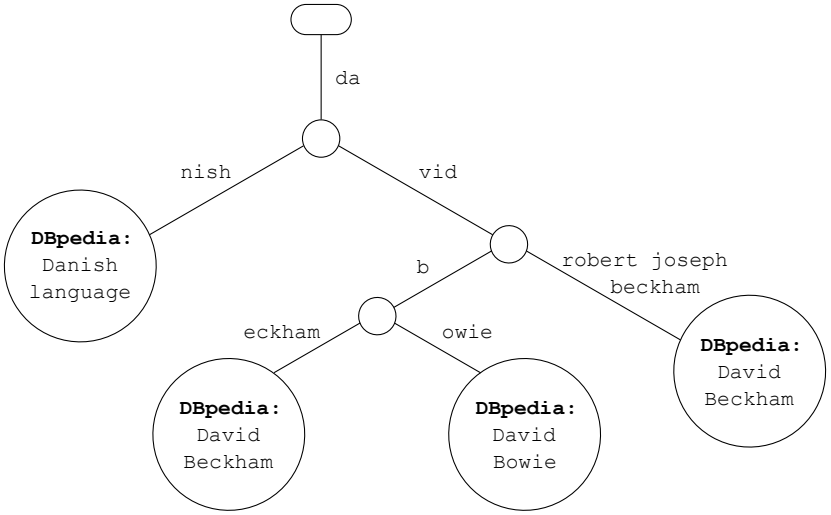


Abbildung 4.10: Exemplarisch dargestellter Präfixbaum: Später kann sehr schnell anhand eines gegebenen Präfix auf eine oder mehrere Entitäten zugegriffen werden.

Zusätzlich errechnen wir für jede Entität ein *idf*-basiertes Gewicht, das beschreibt, wie viel ein Zusammenhang wert ist. Beispielsweise besitzen 192 882 Entitäten eine Verknüpfung zu der Entität *United States*, weshalb dies nicht als starker Zusammenhang angesehen werden kann: Zwischen David Beckham und Thomas Gottschalk, die beide einen Wohnsitz in den USA haben, besteht trotz dieser gemeinsamen Eigenschaft kein ersichtlicher Zusammenhang. Umgekehrt besteht beispielsweise ein starker Zusammenhang zwischen David Beckham und Robbie Keane, da beide mit der Entität *Los Angeles Galaxy* verknüpft sind, zu der deutlich weniger Entitäten einen Zusammenhang aufweisen.

Zur Bestimmung des Gewichts einer Entität e berechnen wir zunächst in Analogie zur inversen Dokumentfrequenz den Wert idf_e:

$$idf_e = \log\left(\frac{|\mathcal{E}|}{pl_e}\right),$$

wobei $\mathcal{E}$ hierbei die Gesamtheit aller Entitäten im DBpedia-Datensatz[4] beschreibt und pl_e die Anzahl jener Entitäten, die eine Verknüpfung zu e aufweisen. Während der Berechnung merken wir uns den maximalen *idf*-Score idf_{max} und normalisieren im Folgeschritt alle *idf*-Scores bezüglich dieses Wertes, um

[4]Wir zählen nur die Entitäten, die in mindestens einer anderen Entität als Verknüpfung enthalten sind.

so ein Gewicht für eine Entität e zwischen 0 und 1 zu bestimmen:

$$w_e = \frac{idf_e}{idf_{max}}.$$

Wir speichern alle Entitäten zusammen mit ihrem jeweiligen Gewicht und den mit ihr verküpften Entitäten in einem invertierten Index, sodass wir auch hierauf schnellen Zugriff haben. Die Entität *David Beckham* aus dem obigen Beispiel würde dementsprechend folgendermaßen abgespeichert:

```
<http://dbpedia.org/resource/David_Beckham> - 0,9207
   [  <http://dbpedia.org/resource/Los_Angeles_Galaxy>,
      <http://dbpedia.org/resource/Victoria_Beckham>  ]
```

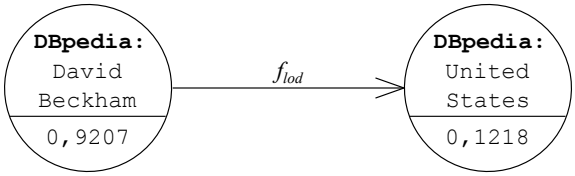
Der gesamte bisher beschriebene Prozess dient lediglich zur Generierung des Präfixbaumes und des invertierten Index, und muss daher nur einmal ausgeführt werden.

Um zu erkennen, ob zwei Anfragen inhaltlich zusammengehören, führen wir folgende Schritte aus:

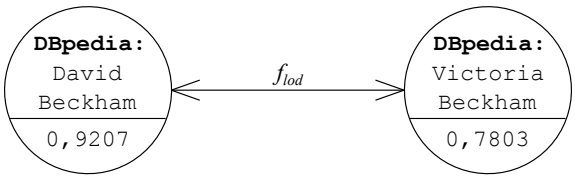
Schritt 1. Der Anfragetext beider Anfragen wird jeweils mittels des Präfixbaumes auf eine oder mehrere Entitäten abgebildet (wir bezeichnen die Menge dieser als $\mathcal{E}_{i-1} = \{e_1, \ldots, e_n\}$ bzw. $\mathcal{E}_i = \{e_1, \ldots, e_m\}$). Existiert für eine der beiden Anfragen kein einziges passendes Label, d. h. keine passende Entität, kann auch kein Zusammenhang gefunden werden.

Schritt 2. Die zugeordneten Entitäten $\mathcal{E}_{i-1}$ und $\mathcal{E}_i$ und die jeweils mit ihnen verknüpften Entitäten in Form von *Postlisten* (wir bezeichnen die Menge dieser als $\mathcal{P}_{i-1}$ und $\mathcal{P}_i$) werden zur weiteren Verarbeitung aus dem invertierten Index in den Speicher geladen.

Schritt 3. Es wird überprüft, ob eine der Entitäten $e_j \in \mathcal{E}_{i-1}$ *direkt* in einer der Postlisten $p_k \in \mathcal{P}_i$ (bzw. im umgekehrten Fall $e_j \in \mathcal{E}_i$ in $p_k \in \mathcal{P}_{i-1}$) enthalten ist. In diesem Fall bestimmt das Gewicht w_j der entsprechenden Entität e_j die Stärke des Zusammenhangs (besteht der Zusammenhang auch in umgekehrter Richtung, wird das größere Gewicht gewählt). Die so ermittelte Stärke des Zusammenhanges bezeichnen wir als f_{lod}.
Die folgenden zwei Beispiele verdeutlichen die Ermittlung der Stärke f_{lod} des Zusammenhangs. Die Richtung der Pfeile gibt jeweils an, in welche Richtung der Zusammenhang besteht.



$$f_{lod} = 0{,}1218$$

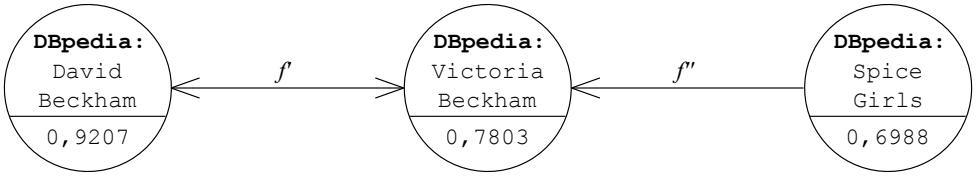


$$f_{lod} = max(0{,}9207,\ 0{,}7803) = 0{,}9207$$

Schritt 4. Ist keine der Entitäten direkt in einer der Postlisten enthalten, werden die Schritte 2 und 3 rekursiv für jede Entität jeder Postlist bis zu einer vorher festgelegten Rekursionstiefe (hierdurch wird die maximale Anzahl von Teilpfaden festgelegt) ausgeführt. Wird hierüber ein Zusammenhang gefunden, werden die Gewichte entlang des Pfades (die nach obiger Beschreibung ermittelt werden) aufsummiert und durch die Anzahl der Teilpfade geteilt. Alle so ermittelten Gewichte entlang *verschiedener* Pfade sollen so aufaddiert werden, dass das Gesamtgewicht f_{lod} nicht größer als 1 sein kann. Der Durchschnitt aller Gewichte ist hierfür aber nicht geeignet, da hohe Gewichte durch niedrige entwertet würden, d. h. eine starke Verbindung, für die man eigentlich annehmen würde, dass beide Entitäten eng miteinander verknüpft sind, wird durch eine schwache Verbindung so entkräftet, dass man keinen starken Zusammenhang mehr vermutet. Da die ermittelten Gewichte einige Ähnlichkeiten zu Wahrscheinlichkeiten aufweisen (jedes Gewicht liegt zwischen 0 und 1 und auch das Gesamtgewicht darf nicht größer als 1 sein), orientieren wir uns an einem altbekannten Mittel der Statistik: dem Additionssatz für die Wahrscheinlichkeit des Auftretens nicht disjunkter Ereignisse. Mit der Siebformel von Poincaré und Sylvester kann die Gesamtwahrscheinlichkeit für *alle* Ereignisse bestimmt werden (dies ist der Kehrwert der Wahrscheinlichkeit für das Eintreten *keines* der Ereignisse). Wir orientieren uns an dieser Formel und bestimmen die Summe aller ermittelten Stärken der einzelnen Pfade $f_1, f_2, \ldots, f_n$, indem wir ihre Komplemente miteinander multiplizieren und das Ergebnis von 1 abziehen:

$$f_{lod} = 1 - [(1 - f_1) \cdot (1 - f_2) \cdot \ldots \cdot (1 - f_n)] \ . \tag{4.1}$$

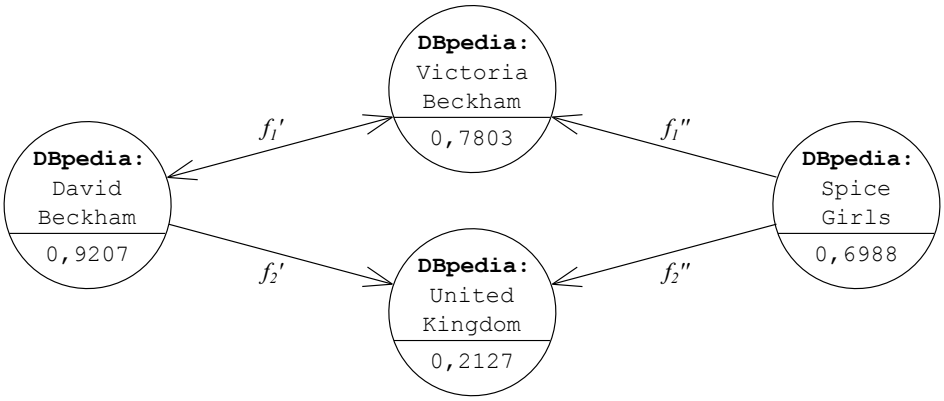
Zur Veranschaulichung dienen die beiden folgenden Beispiele:



$$
\begin{aligned}
f' &= max(0{,}9207,\ 0{,}7803) = 0{,}9207 \\
f'' &= 0{,}7803
\end{aligned}
$$

$$
f_{lod} = \frac{f' + f''}{2} = \frac{0{,}9207 + 0{,}7803}{2} = 0{,}8505
$$

Die Stärke f' des Zusammenhangs zwischen David und Victoria Beckham haben wir oben bereits berechnet und f'' ergibt sich aus dem Gewicht der Entität *Victoria Beckham*, da diese in der Postlist der Entität *Spice Girls* enthalten ist. Normiert auf die Anzahl der Teilpfade ergeben diese beiden Werte ein Gesamtgewicht von 0,8505.



$$
f_1 = \frac{f_1{}' + f_1{}''}{2} = \frac{0{,}9207 + 0{,}7803}{2} = 0{,}8505
$$

$$
f_2 = \frac{f_2{}' + f_2{}''}{2} = \frac{0{,}2127 + 0{,}2127}{2} = 0{,}2127
$$

$$
\begin{aligned}
f_{lod} &= 1 - [(1 - 0{,}8505) \cdot (1 - 0{,}2127)] \\
&= 1 - 0{,}1177 \\
&= 0{,}8823
\end{aligned}
$$

Die Gesamtstärke f_1 für den oberen Pfad haben wir bereits im letzten Beispiel errechnet. Die beiden Teilpfade $f_2{}'$ und $f_2{}''$ bekommen jeweils das Gewicht der Entität *United Kingdom* zugewiesen, da diese als Verknüpfung in den Postlisten von *David Beckham* und den *Spice Girls* enthalten ist. Die Stärke des

gesamten Pfades f_2, die sich aus dem Durchschnitt beider Pfade errechnet, ist somit 0,2127. Die Gesamtstärke f_{lod} der beiden Pfade f_1 und f_2 wird nach Formel 4.1 berechnet und ergibt 0,8823. Aus dieser Rechnung schließen wir also, dass zwischen den beiden Entitäten *David Beckham* und *Spice Girls* eine Verbindung mit der Stärke von 0,8823 besteht.

Um geeignete Parameter für den LOD-Schritt zu bestimmen, führen wir diesen für die nach dem dritten Schritt verbleibenden 241 Anfragepaare des Trainingsdatensatzes mit verschiedenen Konfigurationen aus. Wir experimentieren zunächst mit verschiedenen Mapping-Verfahren, um die Anfragetexte q_{i-1} und q_i auf Entitäten des LOD-Graphen abzubilden:

1. **Exakte Übereinstimmung**
   Eine Anfrage kann nur auf eine Entität abgebildet werden, wenn das Label der Entität *exakt* mit dem Anfragetext übereinstimmt. Für die untersuchten 241 Anfragepaare werden aber nur zwei Fälle gefunden, in denen dies überhaupt für beide Anfragen zutrifft.

2. **Ungefähre Übereinstimmung**
   Eine Anfrage wird auf alle Entitäten abgebildet, deren Label mit dem Anfragetext *beginnt*. Hierbei werden natürlich jede Menge Entitäten erfasst, nach denen eigentlich nicht gesucht wurde. Beispielsweise existieren für den Suchbegriff `new york` 37 weitere Labels, die sich aber nicht auf die Entität *New York* an sich beziehen, sondern z. B. auf ein Buch, eine Band oder einen Fußball-Verein.

3. **Anfragesegmente**
   Wir segmentieren den Anfragetext mit einem Verfahren von Hagen et al. [HPSB11] und suchen nach Labeln, die *exakt* mit den einzelnen Segmenten übereinstimmen. Da hierbei für jedes der Segmente ein passendes Label gefunden werden kann, werden auch hier häufig Entitäten auf die Anfrage abgebildet, die nicht den Kern der Anfrage treffen. Beispielsweise wird die Anfrage `david beckham current team` in die Segmente `david beckham`, `current` und `team` unterteilt. Neben der Entität *David Beckham*, die über das erste Segment gefunden wird und den Kern der Anfrage trifft, werden auch die Entitäten *Current (film)* und *TEAM (Slovak band)* gefunden und somit auf die Anfrage abgebildet. Dies kann zu Fehlern führen, da hierüber gefundene Verbindungen (nach denen eigentlich gar nicht gesucht wurde) natürlich auch in die Berechnung des Gesamtgewichts eingehen und dieses somit verfälschen.

Da das erste Mapping-Verfahren dazu führen würde, dass wir in gerade einmal zwei Fällen versuchen könnten, einen Zusammenhang über den LOD-Graphen zu finden, entscheiden wir uns gegen dieses Verfahren. Das zweite Ver-

fahren liefert schlichtweg zu viele unnötige Entitäten, was zum einen in einer längeren Laufzeit resultiert und zum anderen die Ergebnisse massiv verfälscht. Daher führen wir unsere weiteren Betrachtungen unter Anwendung des dritten Mapping-Verfahrens durch.

Wir experimentieren mit sämtlichen möglichen Kombinationen aus Schwellwerten für f_{lod} (von 0,5 bis 1,0, Schrittweite 0,01) und maximaler Tiefe für Verbindungen im Graphen (wir testen 1 bis 3). Wir erhalten das bestmögliche F-Measure für die Konfiguration mit einer Tiefe von 1 und einem Schwellwert für f_{lod} von 1,0, was bedeutet, dass bessere Ergebnisse erzielt werden, wenn dieser Schritt keine Entscheidungen trifft. Für alle anderen Konfigurationen bei gleichbleibender Tiefe werden gerade einmal zwei Entscheidungen für eine Sitzungsfortführung getroffen, von denen auch noch eine falsch ist. Erhöhen wir die maximale Tiefe für Verbindungen auf 2, entscheidet sich der LOD-Schritt bei einem Schwellwert für f_{lod} von 0,95 insgesamt 33 mal für einen Zusammenhang, wovon allerdings nur 13 korrekt sind.

Bei genauerer Betrachtung der Typ-B-Fehler fallen uns folgende Schwachstellen auf:

1. **Ermittlung des Gesamtgewichts**
   Bei Anwendung der Siebformel von Poincaré und Sylvester (siehe Formel 4.1) kann zwar niemals ein Gesamtgewicht für alle Pfade größer als 1 entstehen, jedoch nähert sich die Siebformel recht schnell diesem Wert je mehr Verbindungen gefunden werden. Bei der Betrachtung der gefundenen Pfade und deren Gewichten für die beiden Suchanfragen `volcanoes in france` und `french desert recipe` stellen wir fest, dass bereits für fünf mittelstarke Verbindungen von 0,39, 0,49, 0,44, 0,32 und 0,41 schon ein Gesamtgewicht von 0,93 resultiert. Allerdings findet der LOD-Schritt zwischen zwei Entitäten durchschnittlich 6,3 Verbindungen (vorausgesetzt es existieren überhaupt welche) und somit erhalten wir fast immer ein Gesamtgewicht von mehr als 0,95, unabhängig davon, ob die Gewichte der einzelnen Pfade sehr niedrig sind oder nicht. Eine Entscheidung über Sitzungsfortführung oder -grenze gleicht damit einem Münzwurf (13 korrekte Entscheidungen, 20 Fehler).

2. **Mapping der Anfragen auf Entitäten**
   Ein Großteil der Anfragen wird mit dem von uns verwendeten Mapping-Verfahren auf unnötige und/oder falsche Entitäten abgebildet. Unter unnötigen Entitäten verstehen wir solche, für die ein Zusammenhang zur Anfrage klar erkennbar ist, jedoch eher eine Zusatzinformation darstellen (z. B. werden bei dem Suchbegriff `victoria beckham` neben der Entität *Victoria Beckham* auch noch ihr gleichnamiges Solo-Album und die TV-Sendung *Victoria Beckham: Coming to America* auf die Anfrage

abgebildet). Falsche Entitäten können auf Segmente zurückgeführt werden, die nicht den eigentlichen Anfragekern widerspiegeln. Beispielsweise würde man bei der Anfrage `lockheed f 104 hunter` 14 Entitäten allein aufgrund des Segments `Hunter` auf die Anfrage abbilden, aber nur eine einzige auf das eigentlich gesuchte Jagdflugzeug. Bei den Entitäten für *hunter* handelt es sich unter anderem um eine Stadt in den USA, einen Fluss in Wales, einen Film aus dem Jahr 1973, zwei Songs von zwei verschiedenen Künstlern, sowie eine Band aus Polen und ein gleichnamiges Album der Band *The Residents*. Für eine folgende Anfrage ist somit das Risiko sehr hoch, dass die ihr zugeordneten Entitäten eine (indirekte) Verbindung zu einer der genannten aufweisen, da diese bereits ein sehr breites Spektrum abdecken. Wir beobachten sogar viele Fälle, in denen ein Zusammenhang über zwei Musikalben oder Liedtitel derselben Musikrichtung gefunden wird, obwohl keine der Anfragen dem Thema Musik zuzuordnen ist. Das Problem der falsch gemappten Anfragen wirkt sich also auch sehr stark auf das erste Problem (Ermittlung des Gesamtgewichts) aus, da viele zusätzliche Pfade in die Berechnung mit eingehen und so das Ergebnis verfälschen.

Um dem erstgenannten Problem beizukommen, beziehen wir in die Berechnung auch die Anzahl der gefundenen Pfade mit ein, indem wir diese mit dem Produkt der Komplemente multiplizieren:

$$f_{lod} = 1 - [(1 - f_1) \cdot (1 - f_2) \cdot \ldots \cdot (1 - f_n) \cdot n]$$

Hiermit erhalten wir für das in Punkt 1 genannte Beispiel ein Gesamtgewicht von 0,65 statt 0,93.

Auf die 241 Anfragen aus unserem Testdatensatz hat diese Anpassung jedoch nur marginale Auswirkungen, da einfach zu viele Pfade über die falsch gemappten Entitäten gefunden werden und somit trotzdem für beinahe jedes Anfragepaar ein Zusammenhang von über 0,95 berechnet wird (vorausgesetzt, es wird überhaupt ein Zusammenhang gefunden). Wir sehen also das Mapping von Anfragetexten auf Entitäten als Hauptproblem an und experimentieren daher mit einigen Lösungsansätzen, die aber auch nicht den gewünschten Erfolg liefern. Beispielsweise versuchen wir, auf Basis des größten Gewichtes oder des längsten Labels jeweils nur diejenige Entität auszuwählen, die am ehesten mit der Suchanfrage übereinstimmt. Allerdings erscheinen uns die gewählten Merkmale zu vage und eine Entscheidung auf deren Grundlage zu willkürlich. Man müsste Algorithmen finden, die nach erfolgter Segmentierung den Kern einer Anfrage bestimmen können, sodass wir gezielt nur nach Entitäten für diesen suchen müssen.

Um das Potenzial des LOD-Schrittes zur Suchsitzungserkennung aufzuzeigen, wenn denn ein geeigneter Anfragetext gegeben ist, entwickeln wir eine Li-

		ESA	LOD
Zusammenhang vorhanden	✓	14	52
	✗	86	48
kein Zusammenhang vorhanden	✓	47	45
	✗	3	5

Tabelle 4.5: Die Anzahl jeweils korrekter (✓) und falscher (✗) Entscheidungen beider Techniken für die Anfragen aus unserer eigens erstellten Liste.

ste, die Anfragen für 100 zusammenhängende Dinge (z. B. `david beckham` und `spice girls`, `mars` und `jupiter`, `linus torvalds` und `linux`) und 50 eher nicht zusammenhängende Dinge (z. B. `the simpsons` und `matt stone`) enthält. Wir lassen für jedes dieser Anfragepaare die Explizite Semantische Analyse (*ESA*) und den von uns entwickelten Linked-Open-Data-Schritt (*LOD*) entscheiden, ob ein Zusammenhang besteht oder nicht und vergleichen die Ergebnisse in Tabelle 4.5. Die komplette Liste der Anfragen und die jeweilige Entscheidung der beiden Techniken kann den Anhängen B.1 und B.2 entnommen werden.

Wie man erkennen kann, findet unser LOD-Ansatz etwas mehr als die Hälfte aller Verbindungen und würde diese Anfragen somit zu einer Suchsitzung zusammenfassen, während die Explizite Semantische Analyse gerade einmal 14 Zusammenhänge erkennen kann. Unser Ansatz entscheidet sich zwar etwas leichtfertiger für eine Sitzungsfortführung als die ESA (5 gegen 3 Fehler), aber angesichts seiner vielen richtigen Entscheidungen bei tatsächlich vorhandenem Zusammenhang zwischen zwei Suchanfragen befinden wir unseren Ansatz als vorteilhafter gegenüber der ESA. Allerdings benötigt unser Verfahren mit durchschnittlich 172 ms Laufzeit je Anfragepaar deutlich länger als die Explizite Semantische Analyse, weshalb wir den LOD-Schritt in unsere Kaskade – wenn überhaupt – nur als vierten Schritt aufnehmen würden. Da jedoch die Suchanfragen aus unserem Gold-Standard mit den uns verfügbaren Algorithmen nicht so eindeutig auf Entitäten abgebildet werden können, wie die Anfragen aus unserer eigens erstellten Liste, sehen wir davon ab, den LOD-Schritt zum jetzigen Zeitpunkt in die Kaskade aufzunehmen. Für die Zukunft können wir uns aber eine Anwendung des LOD-Schrittes vorstellen, wenn denn Lösungen für die beiden genannten Probleme (Bestimmung des Gesamtgewichts bei vielen Pfaden und das Mapping von Anfragen auf Entitäten) gefunden werden.

Um trotzdem noch einige Zusammenhänge zwischen den nach wie vor verbleibenden 241 Anfragepaaren auf unserem Trainingsdatensatz zu erkennen, werden wir im nächsten Schritt versuchen, einen Zusammenhang zwischen je zwei Anfragen über ihre Suchergebnisse zu finden.

4.7 Anfrage-Erweiterung

Die Idee hinter der Anfrage-Erweiterung ist sehr einfach: Werden für zwei Suchanfragen, egal wie verschieden sie sind, sehr ähnliche Ergebnisse gefunden, gehören sie mit hoher Wahrscheinlichkeit zusammen.

Hagen et al. weisen darauf hin, dass die Anfrage-Erweiterung in ihrer Kaskade zwar noch eine marginale Verbesserung des F-Measures bewirkt, dem aber die verhältnismäßig hohe Laufzeit gegenübersteht, und schlagen deshalb vor, auf diesen Schritt komplett zu verzichten [HSR11]. Dies mag für den Online-Anwendungsfall durchaus sinnvoll sein, aber da wir in dieser Arbeit den Anfrage-Log offline in Sitzungen einteilen, möchten wir die maximal mögliche Güte erreichen und experimentieren daher auch mit Verfahren zur Anfrage-Erweiterung.

Hagen et al. prüfen, ob unter den jeweils ersten zehn Suchergebnissen beider Anfragen wenigstens eine Domain übereinstimmt und entscheiden sich in diesem Fall für eine Sitzungsfortführung [HSR11]. Bei unseren Untersuchungen stellen wir aber eine Reihe an Fehlentscheidungen für „Meta-Webseiten“[5] fest, für die wir auf Gleichheit der gesamten URL prüfen. Wir erkennen bei dieser Herangehensweise 25 Zusammenhänge zwischen den 241 verbleibenden Anfragepaaren, fassen aber fälschlicherweise 11 Anfragepaare zusammen, die laut Gold-Standard nicht zusammengehören (schwerwiegende Typ-B-Fehler).

Außerdem experimentieren wir mit der Ähnlichkeit zwischen den Kurzbeschreibungen (*Snippets*) der Suchergebnisse. Hierfür bestimmen wir die n- bis m-Gramme aller Snippets für die jeweilige Suchanfrage und berechnen, wie auch bei der geometrischen Methode, die Kosinusähnlichkeit zwischen den resultierenden Vektoren. Wir ermitteln die Optimalkonfiguration des Schwellwertes für die Ähnlichkeit der Suchergebnisse f_{sr} von 0,44 bei Ermittlung der 3- bis 4-Gramme der ersten 10 Suchergebnisse. In dieser Konfiguration erfasst diese Technik 17 Zusammenhänge, also 8 weniger als der Domainvergleich, trifft dafür aber keine einzige Fehlentscheidung. Bei der hohen Zeichenanzahl in 10 Snippets dauert die Berechnung der Kosinusähnlichkeit auf Basis von n-Grammen natürlich entsprechend lange, was bei dieser Technik aber kaum ins Gewicht fällt: Die durchschnittliche Laufzeit je Anfragepaar beläuft sich nämlich auf 432,78 ms, wovon allerdings 427 ms auf das Stellen der beiden Anfragen an eine Suchmaschine entfallen. Für den Online-Anwendungsfall würde diese Zeit natürlich erheblich niedriger ausfallen, da der Suchmaschinenbetreiber direkten Zugriff auf seinen invertierten Index besitzt.

Aufgrund der höheren Güte würden wir das zweitgenannte Verfahren als vierten Schritt in die Kaskade aufnehmen. Wir raten nicht generell von der

[5] Für unseren Trainingsdatensatz erkennen wir folgende „Meta-Webseiten“: Amazon, Facebook, IMDB, Myspace, Twitter, Wikipedia und YouTube.

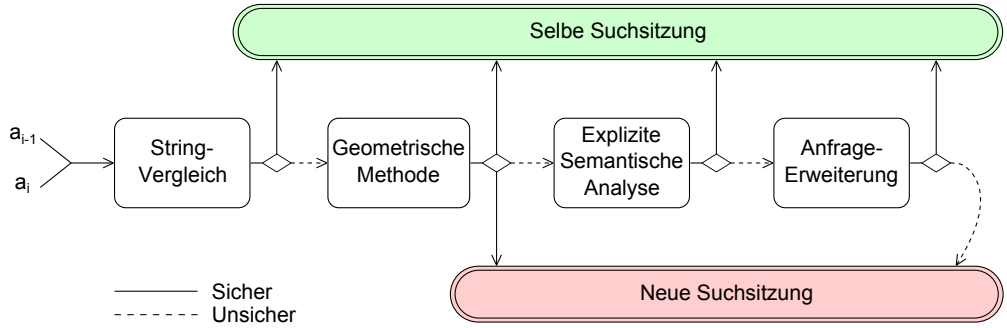


Abbildung 4.11: Die Kaskade nach dem vierten Schritt.

precision	*recall*	F-Measure	ACC_{cont}	ACC_{shift}	ACC_{avg}	Laufzeit (ms)
0,7776	0,9892	0,9128	0,9959	0,7776	0,9362	432,78

		SG im Gold-Standard	
		ja	nein
Für SG gestimmt	ja	549	157
	nein	6	1466

Tabelle 4.6: Ergebnisse der Kaskade nach dem vierten Schritt.

Ausführung dieses Schrittes ab, weisen aber ebenso wie Hagen et al. darauf hin, dass die Anfrage-Erweiterung im Vergleich zu den anderen Techniken sehr viel Zeit benötigt. Abbildung 4.11 zeigt den Aufbau der Kaskade und Tabelle 4.6 die Ergebnisse, die unter Verwendung der Anfrage-Erweiterung als vierten Schritt erzielt werden.

4.8 Ergebnisse auf dem Trainingsdatensatz

Unsere Kaskade nach dem Aufbau aus Abbildung 4.11 segmentiert den Trainingsdatensatz, bestehend aus 2210 Anfragen von 32 Nutzern, innerhalb von 363 ms (ohne den vierten Schritt) bzw. 104 660 ms (mit der Anfrage-Erweiterung als vierten Schritt) bei einer Güte von 90,49 % bzw. 91,11 %, gemessen am F-Measure mit $\beta = 1{,}5$.

Unser Verfahren erkennt 1466 Zusammenhänge korrekt und entscheidet sich in gerade einmal 6 Fällen für eine Sitzungsfortführung zwischen zwei Anfragen, obwohl laut Gold-Standard kein Zusammenhang besteht. In 241 Fällen kann unser Verfahren weder eine sichere Entscheidung für eine Sitzungsfortführung

noch für eine Sitzungsgrenze treffen und trennt daher an diesen Stellen. 142 dieser Trennungen sind korrekt und 83 mal wurde ein bestehender Zusammenhang nicht erkannt. In der Summe fügt unser Verfahren gerade einmal 157 falsche Sitzungsgrenzen ein (Typ-A-Fehler) und ist somit in der Lage, 90,3 % aller Zusammenhänge zwischen Suchanfragen zu erkennen.

Die Tabelle 4.7 fasst die wichtigsten Zwischenergebnisse jedes Schrittes unserer Kaskade in einer einzigen Übersicht zusammen. Hierbei handelt es sich lediglich um die Ergebnisse auf dem Trainingsdatensatz; die erzielten Ergebnisse unserer Kaskade auf dem Testdatensatz sind dem Kapitel 6 zu entnehmen.

				Laufzeit (ms)	
Technik	# Aufrufe	F-Measure	ACC_{avg}	je Paar	gesamt
String-Vergleich	2178	0,7796	0,8407	0,0009	2
Geometrische Methode	1065	0,9029	0,9289	0,24	261
ESA	249	0,9049	0,9306	0,41	363
Anfrage-Erweiterung	241	0,9128	0,9362	432,78	104 660

Tabelle 4.7: Die erzielten Ergebnisse der Kaskade auf dem Trainingsdatensatz.

Kapitel 5

Missionserkennung

Über die Suchsitzungserkennung hinaus stellt die Missionserkennung ein sehr interessantes, aber bisher nur wenig untersuchtes, Forschungsgebiet dar. Spink et al. erkennen, dass Nutzer häufig Suchanfragen zu verschiedenen Themengebieten innerhalb eines kurzen Zeitraumes stellen, und bezeichnen dieses Verhaltensmuster als *Multitasking* [SÖÖ02]. Da dieses Multitasking auch Fälle beinhaltet, in denen ein Zusammenhang zwischen zwei nicht aufeinanderfolgenden Suchanfragen besteht (d. h., es wurden zwischendurch eine oder mehrere Anfragen zu einem anderen Informationsbedarf gestellt), kann man dieses Verhalten auch als Bearbeitung verschiedener Missionen betrachten. Jones und Klinkner untersuchen etwas genauer die Struktur eines Suchanfrage-Logs und stellen fest, dass 17 % der Aufgaben eines Nutzers ineinander verschachtelt und 20 % hierarchisch organisiert sind [JK08].

Wir möchten die von Spink et al. sowie von Jones und Klinkner beschriebenen Strukturen automatisiert erfassen und werden daher versuchen, für jede Sitzung zu erkennen, ob und zu welcher früheren Sitzung ein inhaltlicher Zusammenhang besteht. Auch hierbei soll der Grundgedanke unserer Kaskade zur Suchsitzungserkennung verfolgt und schnelle Techniken vor solchen ausgeführt werden, die viel Zeit für eine Entscheidung benötigen.

Unsere Missionserkennung soll zusammengehörende Sitzungen durch die Vergabe derselben Missions-ID kennzeichnen, wie dies auch schon bei der Entstehung des neuen Goldstandards in Abschnitt 3.3 getan wurde. Längerfristig möchte man hier aber nicht einfach nur wissen, welche Sitzungen einem gemeinsamen Zweck dienen, sondern um welchen (ausformulierten) Zweck es sich dabei handelt, wie etwa eine *Reiseplanung*, die *Recherche für einen Vortrag* oder das *Erstellen eines Familienstammbaumes* (einige ausformulierte Missionsziele können dem Anhang A.1 entnommen werden).

Bisher haben sich nur zwei Forschungsgruppen mit einer automatischen Missionserkennung befasst, nämlich Jones und Klinkner [JK08] sowie Lucche-

		Gold-Standard	
		Neu	Alt
Missionserkennung	Neu	TN	FN
	Alt (korrekt)	–	TP
	Alt (falsch)	FP_1	FP_2

Tabelle 5.1: Fehleranalyse-Tabelle für die Missionserkennung.

se et al. [LOP+11]. Sie stellen in ihren jeweiligen Arbeiten zwar die verwendeten Algorithmen vor, gehen aber nur spärlich auf ihre erzielten Ergebnisse ein, was sicher auch einem fehlenden Evaluationsmodell geschuldet ist. Wir befassen uns daher vor der eigentlichen Missionserkennung mit der Entwicklung eines geeigneten Evaluationsmaßes.

5.1 Evaluation

Wir erstellen zunächst eine Fehleranalyse-Tabelle (ähnlich zu der Tabelle in Abschnitt 3.4), die alle möglichen Übereinstimmungen und Diskrepanzen zwischen den Entscheidungen unserer Missionserkennung und dem Gold-Standard beinhaltet: Im Gold-Standard ist eine Sitzung entweder einer neuen oder einer früheren Mission zugeordnet, wohingegen die Entscheidungen unserer Missionserkennung in drei Kategorien eingeteilt werden:

1. *Neu* - Es wurde keine frühere Sitzung gefunden, die demselben Informationsbedarf dient.

2. *Alt (korrekt)* - Der untersuchten Sitzung wird eine frühere Missions-ID zugewiesen und im Gold-Standard sind diese beiden Sitzungen ebenfalls derselben Mission zugeordnet.

3. *Alt (falsch)* - Der untersuchten Sitzung wird eine frühere Missions-ID zugewiesen, aber im Gold-Standard sind die beiden Sitzungen verschiedenen Missionen zugeordnet.

Tabelle 5.1 zeigt für alle auftretenden Fälle, welchem der vier Maße *true-positive* (TP), *true-negative* (TN), *false-positive* (FP) und *false-negative* (FN) der jeweilige Fall entspricht. Wir entscheiden uns, die Güte anhand korrekt erkannter Missionszuordnungen zu bestimmen, weshalb eine Entscheidung für die Wiederaufnahme einer früheren Mission als *positive* und eine Entscheidung für eine neue Mission als *negative* angesehen wird. Je nachdem, ob die Entscheidung mit dem Gold-Standard übereinstimmt oder nicht, ist diese Entscheidung

richtig (*true*) oder falsch (*false*). Im Gegensatz zur Fehleranalyse-Tabelle für die Sitzungserkennung können hier aber drei statt zwei Fehlentscheidungen auftreten: Entweder ist unser Verfahren nicht in der Lage, einen existierenden Zusammenhang zu einer früheren Sitzung zu erkennen (FN), oder aber das Verfahren entscheidet sich für die Zuordnung zu einer früheren Sitzung, obwohl es sich im Gold-Standard um eine neue Mission handelt (FP_1) bzw. die Sitzung im Gold-Standard einer *anderen* Mission zugeordnet wurde (FP_2). Die beiden letztgenannten Fehler erachten wir als schwerwiegender und weichen damit von unserer Betrachtungsweise aus den Kapiteln 3 und 4 ab, sodass nun die *false-positives*, d. h. die Typ-A-Fehler, als schwerwiegend und die *false-negatives*, also die Fehler vom Typ B, als nicht so schwerwiegend angesehen werden. Einen Typ-B-Fehler, also eine nicht erkannte Wiederaufnahme einer früheren Mission, sehen wir nicht als schwerwiegend an, weil ein Nutzer in diesem Fall mit den Suchergebnissen versorgt würde, die er auch ohne Missionserkennung erhalten hätte; wohingegen ein Typ-A-Fehler die Suchergebnisse u. U. stark verfälschen kann (nämlich dann, wenn die Suchmaschine die Anfrage in einem falschen Kontext bearbeitet). In der Tabelle 5.1 existiert für einen Fall keine Bezeichnung, weil dieser Fall nicht auftreten kann: Ist im Gold-Standard keine Missionsfortführung vermerkt, so kann die Zuordnung zu einer früheren Mission *niemals* korrekt sein.

Wir können nun die beiden Gütemaße *precision* und *recall* folgendermaßen berechnen:

$$\begin{aligned} precision &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}_1 + \mathrm{FP}_2} \\ recall &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}_2} \end{aligned}$$

Die *precision* beschreibt das Verhältnis der korrekten Entscheidungen für eine Wiederaufnahme zu allen vom Verfahren getroffenen Entscheidungen für eine Wiederaufnahme einer früheren Mission. Der *recall* sagt aus, wie viele der laut Gold-Standard zu erkennenden Missionsfortführungen von unserem Verfahren als solche erkannt werden.

Aus diesen beiden Werten kann das F-Measure berechnt werden. Allerdings sollen hier die *false-positives* stärker gewichtet werden, weshalb für β ein Wert kleiner als 1 gewählt werden muss; wir entscheiden uns für $\beta = 0{,}5$.

$$F - Measure_{\beta=0,5} = \frac{(1+\beta^2) \cdot precision \cdot recall}{(\beta^2) \cdot precision + recall}$$

5.2 Einzelschritte der Missionserkennung

Um die Ergebnisse des Verfahrens zur Missionserkennung zu evaluieren, testen wir erneut auf dem aus Kapitel 4 bekannten Trainingsdatensatz. Allerdings betrachten wir nicht die Sitzungen, die unsere Kaskade zur Suchsitzungserkennung bestimmt hat, da diese jeweils fehlerhaft sein können und so einen direkten Vergleich der Ergebnisse der Missionserkennung mit dem Gold-Standard unmöglich machen. Stattdessen wird als Grundlage für die Missionserkennung die Sitzungseinteilung aus dem Gold-Standard verwendet. So kann die optimale Parameterkonfiguration unter der Annahme bestimmt werden, dass die vorausgegangene Suchsitzungserkennung perfekte Arbeit geleistet und die Sitzungen mit 100 %-iger Genauigkeit bestimmt hätte.

Der Trainingsdatensatz beinhaltet 555 Missionen von 32 Nutzern, es gilt also 523 Entscheidungen zu treffen, ob die untersuchte Sitzung s_j einer früheren Mission zuzuordnen oder eine neue Missions-ID zu vergeben ist. In 215 dieser 523 Entscheidungen handelt es sich um die Wiederaufnahme einer früheren Mission und in den verbleibenden 308 Fällen um eine neue Mission.

Im Folgenden werden wir jeweils kurz auf die einzelnen Schritte der Kaskade zur Missionserkennung eingehen, die im Wesentlichen die gleichen sind wie bei der Sitzungserkennung, jedoch mit etwas angepassten Parametern. Auch bei der Missionserkennung hat sich die aus der Sitzungserkennung bekannte Reihenfolge der einzelnen Schritte bewährt und bleibt daher unverändert.

String-Vergleich

Während bei der Suchsitzungserkennung nur zwei einzelne Anfragetexte q_{i-1} und q_i miteinander verglichen wurden (siehe Abschnitt 4.3), wird bei der Missionserkennung geprüft, ob unter *allen* Anfragen der zu untersuchenden Sitzung s_j eine Anfrage vorhanden ist, die eine Wiederholung, eine Generalisierung oder eine Spezialisierung einer Anfrage aus den vorausgegangen Sitzungen darstellt. In diesem Fall wird der Sitzung s_j die Missions-ID jener Sitzung zugeteilt, die diese vorausgegangene Anfrage enthält.

Mit der für die Sitzungserkennung bestimmten Optimalkonfiguration (Anwendung des String-Vergleiches, Anfragen müssen aus mindestens drei Zeichen bestehen) kann die Missionserkennung bereits 75 Fortführungen erfassen, trifft dabei allerdings 30 Fehlentscheidungen vom Typ A. Wird die Mindestlänge der Suchanfragen auf 5 Zeichen erhöht, werden bei gleichbleibender Anzahl korrekter Entscheidungen immerhin 6 Fehlentscheidungen weniger getroffen. Tabelle 5.2 zeigt die Entscheidungen des ersten Schrittes und die daraus resultierenden Gütemaße *precision*, *recall* und F-Measure.

		Gold-Standard	
		Neu	Alt
Missionserkennung	Neu	306	118
	Alt (korrekt)	–	75
	Alt (falsch)	2	22

precision	*recall*	*F*-Measure	Laufzeit je Entscheidung (ms)
0,7576	0,3488	0,6137	0,0852

Tabelle 5.2: Ergebnisse der Missionserkennung nach dem ersten Schritt.

Geometrische Methode

Auch bei der geometrischen Methode können wir von den bereits vollständig erkannten Sitzungen profitieren und verwenden daher die n- bis m-Gramme *aller* Anfragen der zu untersuchenden Sitzung s_j, um die Kosinusähnlichkeit zu den vorausgegangenen Sitzungen zu ermitteln. Allerdings entsteht hierfür eine gewisse Störanfälligkeit für kleine n und m, weshalb wir für die Missionserkennung erneut mit verschiedenen Kombinationen dieser beiden Werte experimentieren. Das beste Gesamtergebnis erhalten wir für $n = 4$ und $m = 5$. Da sich Missionen über mehrere Tage oder auch Wochen hinweg erstrecken können, muss ein größerer zeitlicher Schwellwert für die maximale zeitliche Distanz zwischen zwei Anfragen gewählt werden. Wir testen verschiedene Werte und erzielen auf dem Trainingsdatensatz das höchste F-Measure für $t_{max} = 45$ Tage.[1]

Weiterhin vergrößern wir den Unsicherheitsbereich (siehe Abschnitt 4.4) in beide Richtungen und ignorieren bei der Missionserkennung alle Entscheidungen der geometrischen Methode, wenn diese im Bereich $f_{time} \geq 0{,}8$ (das entspricht 9 Tagen zeitlicher Differenz zwischen zwei Sitzungen) und $f_{lex} \leq 0{,}4$ liegen. Die geometrische Methode trifft aber nach wie vor eine sichere Entscheidung für alle Wertepaare außerhalb des Unsicherheitsbereiches – eine Entscheidung für eine neue Mission ist dann also verbindlich und die weiteren Schritte der Kaskade werden nicht mehr aufgerufen.

In dieser Konfiguration erkennt die Kaskade 54 weitere Zusammenhänge und trifft dabei gerade einmal 6 schwerwiegende Fehlentscheidungen (Typ-A-Fehler), wirkt sich aber auch auf die Entscheidungen des ersten Schrittes aus, sodass dieser 6 Fehlentscheidungen weniger trifft. Die genauen Ergebnisse nach Ausführung des zweiten Schrittes können der Tabelle 5.3 entnommen werden.

[1] Der zeitliche Abstand zweier Sitzungen wird durch den Abstand der letzten Anfrage der vorausgegangenen und der ersten Anfrage der zu untersuchenden Sitzung bestimmt.

		Gold-Standard	
		Neu	Alt
Missionserkennung	Neu	300	70
	Alt (korrekt)	–	129
	Alt (falsch)	8	16

precision	*recall*	*F*-Measure	Laufzeit je Entscheidung (ms)
0,8431	0,6000	0,7799	3,0774

Tabelle 5.3: Ergebnisse der Missionserkennung nach dem zweiten Schritt.

Explizite Semantische Analyse

Auch bei der Expliziten Semantischen Analyse prüfen wir, ob es sinnvoll ist, die ESA-Ähnlichkeit aller Anfragen der zu untersuchenden Sitzung mit jeder Anfrage der vorausgegangenen Sitzungen zu berechnen. Bei dieser Herangehensweise (und einem als optimal ermittelten Schwellwert $f_{esa} = 0{,}7$) kann die Kaskade gerade einmal zwei Missionsfortführungen korrekt erkennen, benötigt aber relativ viel Zeit, da *jede* Anfrage einer Sitzung auch dann untersucht werden *muss*, wenn die beiden Sitzungen nicht zusammengehören. Wir untersuchen daher auch, ob die Bestimmung der Ähnlichkeit zwischen der ersten Anfrage der zu untersuchenden Sitzung und der jeweils letzten Anfrage aller vorausgegangenen Sitzungen ein hinreichendes Kriterium darstellt, und können bei dieser Herangehensweise tatsächlich einen der beiden Zusammenhänge finden, benötigen aber weniger als ein Achtel der Zeit.

Obwohl die ESA mit 0,5069 ms benötigter Laufzeit je Entscheidung vergleichsweise schneller ist als die geometrische Methode nehmen wir sie als

		Gold-Standard	
		Neu	Alt
Missionserkennung	Neu	300	69
	Alt (korrekt)	–	130
	Alt (falsch)	8	16

precision	*recall*	*F*-Measure	Laufzeit je Entscheidung (ms)
0,8442	0,6047	0,7822	0,5069

Tabelle 5.4: Ergebnisse der Missionserkennung nach dem dritten Schritt.

		Gold-Standard	
		Neu	Alt
Missionserkennung	Neu	300	66
	Alt (korrekt)	–	133
	Alt (falsch)	8	16

precision	*recall*	F-Measure	Laufzeit je Entscheidung (ms)
0,8471	0,6186	0,7888	277,42

Tabelle 5.5: Ergebnisse der Missionserkennung nach dem vierten Schritt.

dritten Schritt in die Kaskade auf, weil sie nur so wenige positive Entscheidungen trifft (würde sie als zweiter Schritt ausgeführt, träfe sie drei weitere positive Entscheidungen). Der Tabelle 5.4 kann man die durch die Explizite Semantische Analyse geringfügig verbesserten Ergebnisse entnehmen.

Anfrage-Erweiterung

Als vierten und letzten Schritt werden wir die Anfrage-Erweiterung in die Kaskade zur Missionserkennung aufnehmen und untersuchen zunächst, wie auch schon bei den letzten beiden Schritten, ob nur die erste Anfrage der zu untersuchenden Sitzung und die jeweils letzte Anfrage jeder vorausgegangenen Sitzung oder (unter entsprechender Zunahme der Laufzeit) alle Anfragen in die Entscheidung mit einfließen sollten. Unter Verwendung aller Anfragen werden allerdings nur vier Zusammenhänge mehr gefunden, denen aber 14 schwerwiegende Typ-A-Fehler gegenüberstehen, weshalb wir uns, wie auch schon bei der ESA, für die erstgenannte Variante entscheiden. Das höchste F-Measure erzielen wir bei Betrachtung der 4-Gramme und einer Mindestähnlichkeit f_{sr} der Snippets von 0,45. Tabelle 5.5 zeigt die erzielten Ergebnisse der Anfrage-Erweiterung auf dem Trainingsdatensatz.

Allerdings weisen wir auch an dieser Stelle darauf hin, dass dieser Schritt nur dann ausgeführt werden sollte, wenn die benötigte Zeit keine Rolle spielt: Die Laufzeit der Anfrage-Erweiterung setzt sich nämlich für eine einzige Entscheidung aus durchschnittlich 1,2 Online-Zugriffen[2] auf eine Suchmaschine (mit einer Anfragedauer von jeweils 213,48 ms) und 21,24 ms für die Berechnung der Kosinusunsähnlichkeit zusammen und benötigt somit in der Summe 277,42 ms. Auf dem Trainingsdatensatz wird der vierte Schritt 363 mal aufge-

[2]Wir cachen die bereits gestellten Suchanfragen und da diese sich des Öfteren wiederholen, müssen wir nicht für jedes untersuchte Sitzungspaar zwei Anfragen stellen.

rufen, d. h. allein auf die Anfrage-Erweiterung entfallen 100,7 Sekunden, was angesichts der drei erkannten Zusammenhänge absolut unverhältnismäßig erscheint (auf die Ausführung der ersten drei Schritte entfallen insgesamt 2,14 Sekunden).

5.3 Ergebnisse auf dem Trainingsdatensatz

Die vorgestellte Kaskade zur Missionserkennung beruht im Wesentlichen auf den bereits implementierten Techniken der Sitzungserkennung, für die lediglich einige Anpassungen der Parameter, nicht aber der Funktionsweise, vorgenommen werden müssen. Mit diesen simplen Mitteln erzielen wir auf unserem Trainingsdatensatz sehr zufriedenstellende Ergebnisse bei akzeptabler Laufzeit: Ohne den vierten Schritt, der Anfrage-Erweiterung, benötigt die Kaskade 2,14 Sekunden, wobei sie ein F-Measure von 0,7822 erzielt. Bei Anwendung des zeitaufwändigen vierten Schrittes erhöht sich die Laufzeit auf etwas über eineinhalb Minuten (bzw. 9,45 Sekunden bei simuliertem Direkt-Zugriff auf den invertierten Index einer Suchmaschine) und kann das F-Measure nur marginal auf 0,7888 erhöhen. Spielt die Laufzeit eine wichtige Rolle, so schlagen wir vor, neben der Anfrage-Erweiterung zusätzlich auch auf die Explizite Semantische Analyse zu verzichten, da bereits nach dem Ausführen der ersten beiden Schritte fast alle der überhaupt durch unsere gesamte Kaskade erkennbaren Zusammenhänge gefunden wurden.

Von 215 im Gold-Standard verzeichneten Missionsfortführungen kann unsere Kaskade 133 korrekt erkennen. Demgegenüber stehen gerade einmal 24 Fehlentscheidungen, die wir als schwerwiegend erachten (falsch erkannte Missionsfortführung; Typ-A-Fehler). Die Kaskade entscheidet sich 366 mal für eine neue Missionen, wovon 300 Entscheidungen korrekt sind; 66 Zusammenhänge zu früheren Sitzungen wurden nicht erkannt (Typ-B-Fehler).

Tabelle 5.6 zeigt die Ergebnisse nach jedem einzelnen Schritt in einer einzigen Übersicht.

				Laufzeit	
Technik	*precision*	*recall*	F-Measure	je	gesamt
String-Vergleich	0,7576	0,3488	0,6137	0,0852	44
Geometrische Methode	0,8431	0,6000	0,7799	3,0774	1555
ESA	0,8442	0,6047	0,7822	0,5069	1739
Anfrage-Erweiterung	0,8471	0,6186	0,7888	277,42	102 442

Tabelle 5.6: Die Ergebnisse der Missionserkennung auf dem Trainingsdatensatz.

Kapitel 6

Experimentelle Evaluation

Alle in den Kapiteln 4 und 5 präsentierten Ergebnisse bezogen sich ausschließlich auf den Trainingsdatensatz. Wir haben alle Parameter so bestimmt, dass wir bei Bearbeitung dieses Trainingsdatensatzes das bestmögliche F-Measure erzielen konnten. Die beiden Kaskaden zur Suchsitzungs- sowie Missionserkennung mit den so ermittelten Parameter werden nun auf den Testdatensatz angewandt, um die Güte aufzuzeigen, die sie auf einem Datensatz erzielen können, auf den die Parameter nicht gezielt trainiert wurden.

6.1 Verfahren zur Suchsitzungserkennung

Der Testdatensatz umfasst die verbleibenden 6630 Anfragen von 95 Nutzern aus unserem Gold-Standard und enthält neben 1724 Sitzungsgrenzen 4811 Anfragepaare, zwischen denen ein Zusammenhang besteht.

Der erste Schritt der Kaskade benötigt für die Ausführung aller 6535 String-Vergleiche insgesamt gerade einmal 7 ms (0,0011 ms je Anfragepaar) und kann schon 3413 der 4811 Sitzungsfortführungen korrekt erkennen.

Die geometrische Methode als zweiter Schritt kann weitere 751 Zusammenhänge korrekt erkennen, fasst aber fälschlicherweise auch 20 Anfragen zu einer Sitzung zusammen, zwischen denen laut Gold-Standard kein Zusammenhang besteht (schwerwiegende Typ-B-Fehler). Darüber hinaus fällt die geometrische Methode 1332 sichere Entscheidungen für eine Sitzungsgrenze (1188 davon sind korrekt, 144 bleiben als Typ-A-Fehler erhalten) und überlässt somit den beiden folgenden Schritten nur noch 1019 Entscheidungen, von denen es sich in 503 Fällen um eine Sitzungsfortführung handelt. Die geometrische Methode wird 3122 mal aufgerufen und benötigt in der Summe 498 ms (0,16 ms je Anfragepaar) für ihre Berechnungen.

Die Explizite Semantische Analyse kann, wie bereits bei der Anwendung auf dem Trainingsdatensatz, nur verhältnismäßig wenige Zusammenhänge er-

Technik	# Aufrufe	F-Measure	ACC_{avg}	Laufzeit (ms) je Paar	Laufzeit (ms) gesamt
String-Vergleich	6535	0,8003	0,8947	0,0011	7
Geometrische Methode	3122	0,8889	0,9163	0,16	505
ESA	1019	0,8905	0,9177	0,23	736
Anfrage-Erweiterung	997	0,8940	0,9218	316,41	316 196

Tabelle 6.1: Die erzielten Ergebnisse der Kaskade zur Suchsitzungserkennung auf dem Testdatensatz.

kennen: Auf dem Testdatensatz trifft sie gerade einmal 22 Entscheidungen für eine Sitzungsfortführung, von denen aber 3 falsch sind (Typ-B-Fehler). Für ihre Berechnungen benötigt sie 0,23 ms je Anfragepaar bzw. insgesamt 231 ms.

Würde man die Kaskade nach dem dritten Schritt abbrechen, hätte diese insgesamt 736 ms benötigt, um die 6630 Suchanfragen in Sitzungen einzuteilen. Bis zu diesem Schritt wurden bereits 4183 Zusammenhänge korrekt erkannt und 23 Anfragen wurden einer falschen Sitzung zugeordnet. Es verbleiben also 2329 Anfragepaare, für die eine Sitzungsgrenze angenommen wird: Unsere Kaskade erzielt somit ein F-Measure von $F_{\beta=1,5} = 0{,}8905$.

Da in dieser Arbeit ein bereits abgespeicherter Suchanfrage-Log in Sitzungen eingeteilt werden soll und für uns die Laufzeit eine eher untergeordnete Rolle spielt, brechen wir nicht bereits nach dem dritten Schritt ab, sondern führen zusätzlich noch die Anfrage-Erweiterung aus. Durch sie werden 64 weitere Zusammenhänge erkannt, aber aufgrund der 15 getroffenen Fehlentscheidungen steigt das F-Measure nur minimal. Die bisher sehr schnelle Laufzeit von 736 ms wird durch Ausführung des vierten Schrittes (316,43 ms je Anfragepaar) unverhältnismäßig erhöht, sodass die Kaskade über 5 Minuten (316 196 ms) zur Bearbeitung des gesamten Testdatensatzes benötigt.

Tabelle 6.1 zeigt die wichtigsten Ergebnisse der einzelnen Schritte zusammen mit den Laufzeiten (je Anfragepaar und Gesamtzeit für die Ausführung der Kaskade). Den Tabellen in C.1 bis C.4 können die Teilergebnisse nach Ausführung jedes einzelnen Schrittes entnommen werden.

Wir haben in Kapitel 4 alle Parameter explizit auf den Trainingsdatensatz trainiert und konnten so ein bestmögliches F-Measure von $F_{\beta=1,5} = 0{,}9128$ erreichen. Mit den Ergebnissen auf dem Testdatensatz und einem F-Measure von $F_{\beta=1,5} = 0{,}8940$ sind wir daher äußerst zufrieden.

6.2 Verfahren zur Missionserkennung

Die Ergebnisse unserer Kaskade zur Missionserkennung hingegen fallen deutlich durchwachsener aus. Für den Testdatensatz müssen 1635 Sitzungen hinsichtlich ihrer Missionszugehörigkeit untersucht werden, von denen 794 im Gold-Standard als Missionsfortführung und die verbleibenden 841 Sitzungen als neue Mission gekennzeichnet sind.

Der erste Schritt ist bereits in der Lage, knapp die Hälfte, nämlich 365, der zu findenden Zusammenhänge zwischen Sitzungen zu erkennen, trifft dabei aber auch 148 Fehlentscheidungen. Dieser Schritt benötigt insgesamt 133 ms für die Ausführung der String-Vergleiche zwischen den Sitzungen (0,08 ms je Entscheidung) und erzielt ein F-Measure von $F_{\beta=0,5} = 0{,}6413$.

Die geometrische Methode benötigt mit 5895 ms Gesamtlaufzeit (4,00 ms je Entscheidung) erheblich länger und trifft lediglich 68 richtige Entscheidungen über eine Missionsfortführung, denen aber 22 Typ-A-Fehler gegenüberstehen. Der zweite Schritt kann das F-Measure auf $F_{\beta=0,5} = 0{,}6753$ erhöhen.

Enttäuschende Ergebnisse liefert der dritte Schritt, die Explizite Semantische Analyse. Diese trifft nach 591 ms Gesamtlaufzeit (0,49 ms je Entscheidung) nur drei Entscheidungen für eine Missionsfortführung, von denen jedoch keine korrekt ist. Das F-Measure gegenüber der alleinigen Ausführung der ersten beiden Schritte sinkt somit sogar. Über die Ursachen des schlechten Abschneidens der ESA bei der Missionserkennung können wir nur spekulieren: Zum einen dezimiert unsere Herangehensweise, nur die erste Anfrage der zu untersuchenden und die jeweils letzte Anfrage der vorausgegangenen Sitzungen zu verwenden, natürlich die Anzahl der ESA-Aufrufe und somit u. U. gefundener Zusammenhänge. Allerdings wird sich beim nächsten Schritt zeigen, dass unter dieser Herangehensweise trotzdem noch einige Zusammenhänge gefunden werden können, weshalb wir darin nicht das schwerwiegendste Problem sehen. Zum anderen gehen wir davon aus, dass die ESA generell nur von bedingtem Nutzen für den Vergleich von Suchanfragen ist, da sie schon bei der Sitzungserkennung keine bemerkenswerten Ergebnisse erzielen konnte. Die größte Hürde für die Explizite Semantische Analyse stellt in unseren Augen die Kürze der Anfragen dar, weil häufig schon ein Wort, welches in beiden Suchanfragen vorkommt, die Richtung der beiden Vektoren maßgeblich bestimmt und dadurch der Winkel zwischen diesen sehr klein respektive die Ähnlichkeit sehr groß ist (wie z. B. bei den beiden Suchbegriffen `taurus mountains` und `ford taurus`, siehe Anhang B.2).

Nach den ersten drei Schritten, deren Ausführung insgesamt 6618 ms dauert, werden 433 Missionsfortführungen korrekt erkannt und 173 Sitzungen fälschlicherweise derselben Mission zugeordnet. Das F-Measure hierfür beträgt $F_{\beta=0,5} = 0{,}6728$, was im Vergleich zur Missionserkennung auf dem Trainings-

Technik	# Aufrufe	*precision*	*recall*	F-Measure	Laufzeit gesamt
String-Vergleich	1635	0,7115	0,4597	0,6413	133
Geometrische Methode	1472	0,7181	0,5453	0,6753	6028
ESA	1197	0,7145	0,5453	0,6728	6618
Anfrage-Erweiterung	1196	0,7267	0,5793	0,6915	362 558

Tabelle 6.2: Die erzielten Ergebnisse der Kaskade zur Missionserkennung auf dem Testdatensatz.

datensatz ($F_{\beta=0,5} = 0,7822$) deutlich niedriger ausfällt. Auch erachten wir die Laufzeit in Anbetracht der Ergebnisse für unangemessen hoch, weshalb wir vorschlagen, im Online-Anwendungsfall nur den ersten Schritt, den simplen String-Vergleich, auszuführen.

Da für uns die Laufzeit aber keine wichtige Rolle spielt, führen wir auch den vierten Schritt, die Anfrage-Erweiterung, aus und können bei gleichbleibender Anzahl von Typ-A-Fehlern immerhin 27 Zusammenhänge erkennen. Natürlich benötigt die Ausführung auch hier sehr viel Zeit, da allein 1155 verschiedene Suchanfragen gestellt werden müssen und sich somit die benötigte Zeit für jede Entscheidung auf durchschnittlich auf 297,61 ms beläuft. Allerdings kann das F-Measure durch die Anfrage-Erweiterung immerhin noch auf $F_{\beta=0,5} = 0,6915$ gesteigert werden, weshalb wir nicht auf diesen Schritt verzichten.

Jedoch fällt auch nach Ausführen des vierten Schrittes das Gesamtergebnis nicht vollends zufriedenstellend aus. Verglichen mit dem zuvor erzielten F-Measure von $F_{\beta=0,5} = 0,7888$ müssen wir erkennen, dass die auf dem Trainingsdatensatz bestimmten Parameter wohl nur begrenzte Anwendung auf einem realen Suchanfrage-Log finden können, auf den die Parameter nicht gezielt abgestimmt wurden. Tabelle 6.2 zeigt die Zusammenfassung der Ergebnisse, die unsere Kaskade zur Missionserkennung auf dem Testdatensatz erzielt. Den Tabellen C.5 bis C.8 können die genauen Ergebnisse der einzelnen Schritte und die daraus resultierenden Gütemaße für die gesamte Kaskade entnommen werden.

Da bisher noch keine Ergebnisse anderer Verfahren zur Missionserkennung vorliegen, können wir die Qualität unserer Ergebnisse nicht recht einordnen. Wir vergleichen daher die Güte der Entscheidungen unserer Kaskade mit denen einer Baseline, die auf Basis eines statistischen Merkmals eine zufällige Entscheidung trifft. Hierfür ermitteln wir zunächst auf dem Trainingsdatensatz, wie häufig eine neue Mission vorliegt bzw. eine frühere Sitzung wieder aufgegriffen wurde (wir zählen 308 neue Missionen und 215 Missionsfortführungen, siehe Abschnitt 5.2). Im Fall einer Missionsfortführung zählen wir, wie viele andere

Baseline		Gold-Standard	
		Neu	Alt
Missionserkennung	Neu	558	475
	Alt (korrekt)	–	45
	Alt (falsch)	283	274

Kaskade		Gold-Standard	
		Neu	Alt
Missionserkennung	Neu	806	223
	Alt (korrekt)	–	433
	Alt (falsch)	35	138

Tabelle 6.3: Gegenüberstellung der Baseline und der Kaskade zur Missionserkennung.

Sitzungen zwischen der untersuchten und der vorausgegangenen Sitzung mit derselben Missions-ID liegen (in 81 Fällen liegt nur eine Sitzung dazwischen, in 30 Fällen zwei Sitzungen usw.). Auf dem Testdatensatz entscheidet sich unsere Baseline anhand der vorher bestimmten relativen Häufigkeiten zufällig für die Vergabe einer neuen oder einer alten Missions-ID, wobei im letzteren Fall die Missions-ID jener Sitzung gewählt wird, für die die Anzahl der zufällig bestimmten Sitzungen zwischen ihr und der untersuchten Sitzung liegen.

In Tabelle 6.3 sind die Ergebnisse der Baseline und unserer Kaskade gegenübergestellt. Wie man erkennen kann, trifft unser Verfahren deutlich mehr richtige Entscheidungen (Kombinationen *Neu-Neu* und *Alt-Alt (korrekt)* in der Fehleranalyse-Tabelle) als die Baseline, für die sich ein F-Measure von gerade einmal $F_{\beta=0,5} = 0{,}0703$[1] ergibt.

[1]Es handelt sich hierbei um das beste Ergebnis, das die Baseline in 1000 Durchläufen erzielen konnte.

Kapitel 7

Fazit und Ausblick

Wir haben uns in dieser Arbeit mit Methoden zur automatischen Sitzungs- sowie Missionserkennung in Suchanfrage-Logs, d. h. der Gruppierung von Suchanfragen, die demselben Informationsbedarf dienen, befasst. Wir haben insbesondere drei Schwerpunkte behandelt:

1. Wir haben einen neuen, unserer Auffassung nach qualitativ besseren, Gold-Standard erstellt, der zur freien Verfügung online gestellt wurde.[1]

2. Anschließend haben wir die existierende Kaskade zur Suchsitzungserkennung von Hagen et al. [HSR11] untersucht und Verbesserungen an ihr vorgenommen. Zusätzlich zu den bereits existierenden Schritten haben wir einen neuen möglichen Schritt zur Suchsitzungserkennung vorgestellt, der Zusammenhänge zwischen Anfragen über einen Graphen auf Basis von miteinander verknüpften Daten im Web (*Linked Open Data*) finden sollte.

3. Abschließend haben wir ein Modell zur Evaluation für die Ergebnisse einer automatischen Missionserkennung vorgestellt und untersucht, ob und wie gut unsere verbesserte Kaskade zur Suchsitzungserkennung auch diese Aufgabe erfüllen kann.

Ein neuer Gold-Standard

Auf Basis des Gold-Standards von Daniel Gayo-Avello [Gay09] haben wir einen neuen Gold-Standard erstellt, bei dessen Bearbeitung wir großen Wert auf Gründlichkeit gelegt haben. Zusätzlich haben wir jeder Anfrage eine Missions-ID zugeordnet, sodass der neue Gold-Standard zur Evaluation von Verfahren zur Missionserkennung genutzt werden kann. Unserem Kenntnisstand nach

[1] http://www.webis.de/research/corpora

existiert kein anderer Gold-Standard in diesem Umfang (8840 Anfragen von 127 Nutzern), der für diesen Zweck geeignet ist.

Suchsitzungserkennung

Wir haben die bestehende Kaskade, die von Hagen et al. vorgeschlagen wurde [HSR11], gründlich untersucht und einige Anpassungen vorgenommen. Diese resultieren gleichermaßen in einer besseren Laufzeit gegenüber der Nachimplementierung der originalen Kaskade, als auch in etwas besserer Güte auf unserem neuen Gold-Standard (die Kaskade von Hagen et al. erreicht ein F-Measure von 0,8812 gegenüber unserer Kaskade, die ein F-Measure von 0,8940 erzielt).

Der von uns entwickelte LOD-Ansatz erscheint zunächst vielversprechend, liefert jedoch keine verlässlichen Entscheidungen, sodass wir in unseren beiden Kaskaden zur Suchsitzungs- sowie Missionserkennung auf diesen neuen Schritt vorerst verzichten. Wir erkennen aber sehr wohl das Potenzial dieser Technik und haben anhand einer eigens für diesen Zweck erstellten Liste gezeigt, dass der LOD-Ansatz deutlich mehr Zusammenhänge zwischen, zugegebenermaßen sehr klar formulierten, Suchanfragen erkennen kann als die Explizite Semantische Analyse.

Die Ursache für die nicht zufriedenstellenden Ergebnisse des LOD-Schrittes sehen wir in erster Linie darin, dass reale Suchanfragen nicht so klar formuliert sind, wie die aus unserer Liste, und somit eine exakte Abbildung einer Anfrage auf eine Entität im DBpedia-Datensatz schwerfällt. Mit Verfahren zur Anfragesegmentierung, die neben der alleinigen Ermittlung der Segmente auch in der Lage sind, den Anfragekern zu erkennen und zu markieren, könnte man diesem Problem aber vermutlich beikommen.

Desweiteren ist nicht klar, ob die durch uns festgelegte Bestimmung der Gewichte zwischen den Entitäten die geeignetste Lösung darstellt. Grundsätzlich erachten wir eine Bewertung auf Grundlage der inversen Dokumentfrequenz (idf) als sinnvoll, jedoch ist zu überlegen, ob vielleicht auch der Typ der Beziehung zwischen zwei Entitäten in die Berechnung mit eingehen sollte. So könnte man beispielsweise dem Beziehungstypen *spouse* grundsätzlich eine starke Gewichtung geben (man würde für zwei hierdurch verbundenen Entitäten einen sehr starken Zusammenhang völlig unabhängig vom Gewicht der verknüpften Entität zuweisen) und dem Beziehungstypen *country* eine sehr niedrige, weil man einen Zusammenhang über das gleiche Herkunftsland als nicht ausreichend erachtet.

Nicht zuletzt ist auch die Verwendung des Siebformel-Ansatzes für die Summierung der Gewichte aller gefundenen Pfade fraglich. Für zukünftige Arbeiten schlagen wir vor, nach geeigneteren Lösungen für dieses Problem zu suchen.

Zusammenfassend können wir sagen, dass wir mit unserer verbesserten Kaskade zur Suchsitzungserkennung sehr gute Ergebnisse erzielen konnten, auch wenn letztlich nur einige Verbesserungen an der bereits bestehenden Kaskade durchgeführt wurden und der neu entwickelte LOD-Schritt leider keine Vorteile bringt.

Missionserkennung

Wir haben in dieser Arbeit untersucht, ob die einzelnen Schritte der Kaskade zur Suchsitzungserkennung ebenso auf die Missionserkennung angewendet werden können. Und tatsächlich konnten wir nach Anpassung einiger Parameter auf unserem Trainingsdatensatz ein F-Measure von 0,7888 erzielen. Unter Anwendung derselben Kaskade auf dem Testdatensatz jedoch zeigte sich, dass die Missionserkennung ein äußerst komplexes Problem darstellt, das durch Anwendung des Verfahrens zur Suchsitzungserkennung in ausreichender, aber nicht in bedeutend hoher, Güte gelöst werden kann (wir erreichen auf dem Testdatensatz ein F-Measure von 0,6915). Zwar haben wir gezeigt, dass unser Verfahren weitaus bessere Entscheidungen als die Baseline (zufällige Entscheidung auf Basis eines zuvor ermittelten statistischen Merkmals) trifft, aber aufgrund der hohen Anzahl an schwerwiegenden Fehlentscheidungen (auf dem Testdatensatz sind es insgesamt 173) sind wir mit den erzielten Ergebnissen nur bedingt zufrieden.

Für zukünftige Arbeiten raten wir, eine initiale Untersuchung der Charakteristika von Missionen in Suchanfrage-Logs durchzuführen, d. h. das Suchverhalten von Nutzern in Bezug auf Missionen ausführlich zu studieren, und die gewonnenen Erkenntnisse in die Entwicklung von Algorithmen zur Missionserkennung einfließen zu lassen.

Anhang A

Überarbeiteter Gold-Standard

A.1 Missionen für Nutzer #1936169

MissionID	Mission
1	Inform me about my favorite soaps.
2	Find information on garden and plants.
3	Find out something about cats.
4	Give me porn!
5	Read the news.
6	Find information on Star Wars.
7	Find online games.
8	Find out something about Old Pendleton County.
9	List me all prison inmates in South Carolina.
10	I want to find out all about chickens. :)
11	Where can I buy playing cards?
12	Read something about 80's music.
13	Find information on the oncoming geography contest.
14	Watch music television online.
15	Direct me to the kmart webpage.
16	Give me information on drug use.
17	Find a magistrate in Oconee County.
18	Search for alternative search engines.
19	Find a new house to live in.
20	Inform me about mushrooms.
21	Find information on pandas.
22	Find a tax assessor in Pickens County.
23	Inform me about millipedes.
24	Find out something about Titanic.
25	Help me creating a genealogical tree.
26	Find online medicines.
27	Navigate me to Ebay.
28	Find general information on music.
29	Direct me to Wikipedia.

A.2 Sonderfälle

ID	Anfrage	Zeit	Rang	URL
6392781	`internet`	2006-03-03 22:51:08	1	
6392781	`map of north carolina`	2006-03-03 22:51:32	10	www.geology.enr.state.nc.us
6392781	`internet`	2006-03-04 23:49:18		
6392781	`georgia lottery`	2006-03-04 23:49:39	1	www.galottery.com
6392781	`internet`	2006-03-26 22:11:13		
6392781	`south carolina education lottery`	2006-03-26 22:12:01	1	www.sceducationlottery.com
6392781	`internet`	2006-03-30 22:29:17		
6392781	`south carolina education lottery`	2006-03-30 22:31:44	1	www.sceducationlottery.com

Wahrscheinlich handelte es sich hier um eine ältere Person, die dachte, dass sie immer erst `internet` in das Suchfeld eingeben müsse, um sich damit zu verbinden. Daher haben wir die Anfrage in diesem Zusammenhang einfach ignoriert und sie der Sitzung zugeordnet, zu der es entsprechend seines Zeitstempels am ehesten gehört.

ID	Anfrage	Zeit	Rang	URL
8818991	`autobiographer of speake memory`	2006-03-12 13:54:07		
8818991	`autobiographer of speak memory 1951`	2006-03-12 13:55:09		
8818991	`author rushdie`	2006-03-12 13:56:22		
8818991	`phedre playwright`	2006-03-12 13:57:03		
8818991	`a dolls house wife`	2006-03-12 13:58:00		
8818991	`jeffersons vice prseident`	2006-03-12 13:59:06	3	vice-president-of-the-...
8818991	`us vice presidents`	2006-03-12 14:01:39	1	www.presidentsusa.net
8818991	`creighton university`	2006-03-12 14:02:38	1	www.creighton.edu
8818991	`charlotte--- virgin islands`	2006-03-12 14:04:10		
8818991	`royal lakes`	2006-03-12 20:31:08		

Diese Anfragen scheinen auf den ersten Blick nichts miteinander zu tun zu haben, jedoch ergab sich bei genauerer Recherche, dass es sich hierbei um Fragen aus dem Kreuzworträtsel der New York Times vom 12. März 2006 handelt (vgl. beispielsweise `http://crosswordtracker.com/clue/author-rushdie/` oder `http://crosswordtracker.com/clue/creighton-university-site/`, Letzter Zugriff: 27. März 2012). Bei der letzten Anfrage `royal lakes` haben wir keine Information darüber gefunden, ob es sich auch um eine Frage aus einem Kreuzworträtsel handelt. Gemäß unseres Grundsatzes „Im Zweifelsfall lieber einer neuen Sitzung zuordnen“ haben wir uns daher für eine neue Suchsitzung entschieden.

Anhang B

Vergleich von ESA und LOD

B.1 Zusammenhängende Suchanfragen

Wir prüfen auf den folgenden 100 Anfragepaaren, wie viele Zusammenhänge die ESA bzw. der LOD-Schritt finden können. Konnte eine der beiden Techniken den Zusammenhang erkennen, ist dies jeweils mit einem ✓ markiert.

Anfrage 1	Anfrage 2	ESA	LOD
buzz aldrin	neil armstrong		✓
pyongyang	kim yong il		✓
dirk nowitzki	jason kidd		
chaine des puys	puy de dome		✓
eiffel tower	paris		✓
angelina jolie	brad pitt		
tokio hotel	magdeburg		✓
thuringia	erfurt		
walmart	tesco		
volkswagen	martin winterkorn		✓
david beckham	spice girls		✓
caribbean	haiti		
dell	texas		
washington	barack obama		✓
michael schumacher	formula 1		
berlin	frankfurt	✓	✓
berlin	munich		✓
jodie foster	panic room		
snatch	jason statham		✓
snatch	crank		✓
stuttgart	black forest		

Anfrage 1	Anfrage 2	ESA	LOD
michael jackson	earth song		
snickers	mars		
mars	jupiter		✓
ottawa	toronto		✓
the offspring	americana		✓
james bond	aston martin		
james bond	martini		
lake erie	lake huron	✓	✓
madrid	barcelona		✓
mercedes	amg		✓
albania	montenegro		✓
golden globe	clint eastwood		
asterix	obelix		✓
empire state building	world trade center	✓	
empire state building	chrysler building		
how i met your mother	cbs		
cbs	nbc	✓	✓
linus torvalds	linux		✓
bill gates	steve ballmer		✓
pyrenees	andorra		✓
monaco	monte carlo		✓
provence	marseille		
cuba	gulf of mexico		
caribbean	cuba		✓
stephen king	richard bachmann		
franz kafka	the metamorphosis		✓
olympic games	greece		
ea games	crysis		
julius caesar	brutus		
prince	queen		✓
robbie williams	take that		✓
bauhaus	walter gropius		
north pole	amundsen scott		
arnold schwarzenegger	terminator		
boeing	chicago		
bruce willis	die hard		
texas	austin		✓
mount everest	nepal		✓
mount everest	edmund hillary		
mariana trench	pacific ocean		
bank of scotland	edinburgh	✓	

Anfrage 1	Anfrage 2	ESA	LOD
lima	peru		✓
alien	predator		✓
john travolta	uma thurman	✓	
christoph waltz	oscar		
super mario	nintendo	✓	
kurt cobain	cocaine		✓
johny cash	elvis presley	✓	✓
den haag	amsterdam		✓
quechua	peru	✓	✓
quechua	lima		✓
ural	russia		✓
great wall of china	taj mahal		
steven spielberg	jurassic park		
tom cruise	katie holmes		
grand canyon	las vegas		
nile	egypt		✓
intel	amd	✓	
challenger	dodge charger		✓
mount rushmore	theodore roosevelt		
australian open	wimbledon	✓	
alaska	university peak		✓
yale	princeton		✓
snowy mountains	mount jagungal		✓
astrid lindgren	pippi longstocking		✓
everglades	florida		
the simpsons	matt groening	✓	✓
phil collins	genesis		✓
vatican city	rome		✓
alps	austria		✓
club mate	chaos computer club	✓	
johnny depp	tim burton		
edgar allen poe	howard phillips lovecraft		
lord of the rings	hobbit		
sugarloaf mountain	rio de janeiro		✓
berlin	brandenburg gate		
mark zuckerberg	facebook		
galicia	santiago de compostela	✓	✓
titanic	leonardo dicaprio		✓

Während die ESA nur 14 Zusammenhänge erkennen kann, würde unser LOD-Schritt 52 Anfragepaare zusammenfassen.

B.2 Nicht zusammenhängende Suchanfragen

Wir prüfen auf den folgenden 50 Anfragepaaren, ob die ESA bzw. der LOD-Schritt korrekterweise keinen Zusammenhang zwischen den Anfragen herstellen können. Sollte eine der beiden Techniken fälschlicherweise einen Zusammenhang erkennen, ist dies jeweils mit einem ✗ markiert.

Anfrage 1	Anfrage 2	ESA	LOD
dirk nowitzki	mel gibson		
thomas gottschalk	david beckham		
eiffel tower	germany		
barack obama	paris hilton		
austria	berlin		
lake erie	hudson river		
back street boys	robbie williams		
big ben	paris		✗
berlin wall	china		
tom hanks	snatch		
beatles	linus torvalds		
canada	wyoming		✗
albania	mount moehau		
katie holmes	david beckham		
empire state building	paris		
united kingdom	united airlines	✗	
sleepless in seattle	guy ritchie		
paul kalkbrenner	pink floyd		
france	canberra		✗
john travolta	lebron james		
mel gibson	melanie chisholm		
audi	amg		
boeing	toulouse		
jimmy carter	bruce willis		
thomas edison	justin timberlake		
taurus mountains	egypt		
taurus mountains	ford taurus	✗	
pyramid	south africa		
edinburgh	cardiff		
eminem	albert einstein		
jason statham	spice girls		
lake ontario	aral sea		
kill bill	bruce willis		
oscar	canne		
great barrier reef	sweden		

Anfrage 1	Anfrage 2	ESA	LOD
david beckham	michelle obama		
vatican city	chicago		
statue of liberty	las vegas		
montenegro	mount rushmore		
california	tallahassee		✗
ural	spain		
steve ballmer	joe cocker		
the simpsons	matt stone		
czech republic	san miguel		✗
mariah carey	uma thurman	✗	
dreamworks	finding nemo		
bruce willis	france		
michael jackson	johnny depp		
hong kong	charles de gaulle		
alps	cuba		

Während sich die ESA dreimal fälschlicherweise für einen Zusammenhang zwischen den Anfragen entscheidet, trifft der LOD-Schritt fünf Fehlentscheidungen. Wir stellen fest, dass dieser häufig Verbindungen zwischen geografischen „Orten“ (in den obigen Beispielen Länder, Städte und Sehenswürdigkeiten) findet und hierfür die meisten Fehler entstehen. Die ESA hingegen trifft vor allem dann Fehlentscheidungen, wenn ein Wort in beiden Anfragen enthalten ist (wie z. B. `taurus` in `taurus mountains` und `ford taurus`), wodurch die Richtung des Vektors maßgeblich bestimmt wird.

Anhang C

Ergebnisse der Einzelschritte

C.1 Suchsitzungserkennung

precision	*recall*	F-Measure	ACC_{cont}	ACC_{shift}	ACC_{avg}	Laufzeit
0,5522	1,0000	0,8003	1,0000	0,5522	0,8947	0,0011

		SG im Gold-Standard	
		ja	nein
Für SG gestimmt	ja	1724	1398
	nein	0	3413

Tabelle C.1: Ergebnisse auf dem Testdatensatz nach dem ersten Schritt.

precision	*recall*	*F*-Measure	ACC_{cont}	ACC_{shift}	ACC_{avg}	Laufzeit
0,7248	0,9884	0,8889	0,9952	0,7248	0,9163	0,16

		SG im Gold-Standard	
		ja	nein
Für SG gestimmt	ja	1704	647
	nein	20	4164

Tabelle C.2: Ergebnisse auf dem Testdatensatz nach dem zweiten Schritt.

precision	*recall*	*F*-Measure	ACC_{cont}	ACC_{shift}	ACC_{avg}	Laufzeit
0,7304	0,9867	0,8905	0,9945	0,7304	0,9177	0,23

		SG im Gold-Standard	
		ja	nein
Für SG gestimmt	ja	1701	628
	nein	23	4183

Tabelle C.3: Ergebnisse auf dem Testdatensatz nach dem dritten Schritt.

precision	*recall*	*F*-Measure	ACC_{cont}	ACC_{shift}	ACC_{avg}	Laufzeit
0,7493	0,9780	0,8940	0,9911	0,7493	0,9218	316,41

		SG im Gold-Standard	
		ja	nein
Für SG gestimmt	ja	1686	564
	nein	38	4247

Tabelle C.4: Ergebnisse auf dem Testdatensatz nach dem vierten Schritt.

C.2 Missionsserkennung

		Gold-Standard	
		Neu	Alt
Missionserkennung	Neu	833	289
	Alt (korrekt)	–	365
	Alt (falsch)	8	140

precision	*recall*	*F*-Measure	Laufzeit je Entscheidung (ms)
0,7115	0,4597	0,6413	0,0816

Tabelle C.5: Ergebnisse auf dem Testdatensatz nach dem ersten Schritt.

		Gold-Standard	
		Neu	Alt
Missionserkennung	Neu	808	224
	Alt (korrekt)	–	433
	Alt (falsch)	33	137

precision	*recall*	*F*-Measure	Laufzeit je Entscheidung (ms)
0,7181	0,5453	0,6753	4,0048

Tabelle C.6: Ergebnisse auf dem Testdatensatz nach dem zweiten Schritt.

		Gold-Standard	
		Neu	Alt
Missionserkennung	Neu	806	223
	Alt (korrekt)	–	433
	Alt (falsch)	35	138

precision	*recall*	*F*-Measure	Laufzeit je Entscheidung (ms)
0,7145	0,5453	0,6728	0,4937

Tabelle C.7: Ergebnisse auf dem Testdatensatz nach dem dritten Schritt.

		Gold-Standard	
		Neu	Alt
Missionserkennung	Neu	802	200
	Alt (korrekt)	–	460
	Alt (falsch)	39	134

precision	*recall*	*F*-Measure	Laufzeit je Entscheidung (ms)
0,7267	0,5793	0,6915	297,61

Tabelle C.8: Ergebnisse auf dem Testdatensatz nach dem vierten Schritt.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[BBC+08] Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, Aristides Gionis und Sebastiano Vigna. The Query-flow Graph: Model and Applications. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, Seiten 609–618, 2008.

[Gay09] Daniel Gayo-Avello. A Survey on Session Detection Methods in Query Logs and a Proposal for Future Evaluation. *Information Sciences*, 179:1822 – 1843, 2009.

[GM07] Evgeniy Gabrilovich und Shaul Markovitch. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th International Joint Conference for Artificial Intelligence*, Seiten 1606–1611, 2007.

[HCO03] Chien-Kang Huang, Lee-Feng Chien und Yen-Jen Oyang. Relevant Term Suggestion in Interactive Web Search Based on Contextual Information in Query Session Logs. *Journal of the American Society for Information Science and Technology*, 54:638–649, 2003.

[HG00] Daqing He und Ayse Göker. Detecting Session Boundaries from Web User Logs. In *Proceedings of the BCS-IRSG 22nd Annual Colloquium on Information Retrieval Research*, Seiten 57–66, 2000.

[HGH02] Daqing He, Ayse Göker und David J. Harper. Combining Evidence for Automatic Web Session Identification. *Information Processing and Management*, 38:727–742, 2002.

[HPSB11] Matthias Hagen, Martin Potthast, Benno Stein und Christof Bräutigam. Query Segmentation Revisited. In *Proceedings of the 20th International Conference on World Wide Web*, Seiten 97–106, 2011.

[HSR11] Matthias Hagen, Benno Stein und Tino Rüb. Query Session Detection as a Cascade. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, Seiten 147–152, 2011.

[HTV11] Vera Hollink, Theodora Tsikrika und Arjen P. de Vries. Semantic Search Log Analysis: A Method and a Study on Professional Image Search. *Journal of the American Society for Information Science and Technology*, 62:691–713, 2011.

[JK08] Rosie Jones und Kristina Lisa Klinkner. Beyond the Session Timeout: Automatic Hierarchical Segmentation of Search Topics in Query Logs. In *Proceedings of the 17th ACM International Conference on Information and Knowledge Management*, 2008.

[JS06] Bernard J. Jansen und Amanda Spink. How are we Searching the World Wide Web? A Comparison of Nine Search Engine Transaction Logs. *Information Processing and Management*, 42:248–263, 2006.

[JSBK07] Bernard J. Jansen, Amanda Spink, Chris Blakely und Sherry Koshman. Defining a Session on Web Search Engines. *Journal of the American Society for Information Science and Technology*, 58:862–871, 2007.

[JSBS98] Bernard J. Jansen, Amanda Spink, Judy Bateman und Tefko Saracevic. Real Life Information Retrieval: A Study of User Queries on the Web. *SIGIR Forum*, 32:5–17, 1998.

[JSS00] Bernard J. Jansen, Amanda Spink und Tefko Saracevic. Real Life, Real Users, and Real Needs: A Study and Analysis of User Queries on the Web. *Information Processing and Management*, 36:207–227, 2000.

[LH99] Tessa Lau und Eric Horvitz. Patterns of Search: Analyzing and Modeling Web Query Refinement. In *Proceedings of the 7th International Conference on User Modeling*, Seiten 119–128, 1999.

[LOP+11] Claudio Lucchese, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri und Gabriele Tolomei. Identifying Task-based Sessions in Search Engine Query Logs. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, Seiten 277–286, 2011.

[MLC06] Craig G. Murray, Jimmy Lin und Abdur Chowdhury. Identification of User Sessions with Hierarchical Agglomerative Clustering. *Proceedings of the American Society for Information Science and Technology*, 43:1–5, 2006.

[MRS08] Christopher D. Manning, Prabhakar Raghavan und Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 1. Auflage, 2008.

[ÖÇ05a] H. Cenk Özmutlu und Fatih Çavdur. Application of Automatic Topic Identification on Excite Web Search Engine Data Logs. *Information Processing and Management*, 41:1243–1262, 2005.

[ÖÇ05b] Seda Özmutlu und Fatih Çavdur. Neural Network Applications for Automatic New Topic Identification. *Online Information Review*, 29:34–53, 2005.

[ÖÖB07] Seda Özmutlu, H. Cenk Özmutlu und Buket Buyuk. Using Monte-Carlo Simulation for Automatic New Topic Identification of Search Engine Transaction Logs. In *Proceedings of the 39th Conference on Winter Simulation*, Seiten 2306–2314, 2007.

[ÖÖS08] Seda Özmutlu, H. Cenk Özmutlu und Amanda Spink. Automatic New Topic Identification in Search Engine Transaction Logs Using Multiple Linear Regression. In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, 2008.

[PCT06] Greg Pass, Abdur Chowdhury und Cayley Torgeson. A Picture of Search. In *Proceedings of the 1st International Conference on Scalable Information Systems*, 2006.

[Rüb11] Tino Rüb. Ein Kaskadierendes Verfahren zur Detektierung von Suchsitzungen in Anfrage-Log-Dateien. Bachelorarbeit, Bauhaus-Universiät Weimar, Fakulät Medien, Mediensysteme, Februar 2011.

[RJ05] Filip Radlinski und Thorsten Joachims. Query Chains: Learning to Rank from Implicit Feedback. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, Seiten 239–248, 2005.

[SC06] Nuno Seco und Nuno Cardoso. Detecting User Sessions in the Tumba! Query Log, 2006.

[SH06] Mehran Sahami und Timothy D. Heilman. A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets. In *Proceedings of the 15th International Conference on World Wide Web*, Seiten 377–386, 2006.

[SLY+10] Shuqi Sun, Sheng Li, Muyun Yang, Haoliang Qi und Tiejun Zhao. Utilizing Variability of Time and Term Content, within and across Users in Session Detection. In *Proceedings of the 23rd International Conference on Computational Linguistics*, Seiten 1203–1210, 2010.

[SMHM99] Craig Silverstein, Hannes Marais, Monika Henzinger und Michael Moricz. Analysis of a Very Large Web Search Engine Query Log. *SIGIR Forum*, 33:6–12, 1999.

[SÖÖ02] Amanda Spink, H. Cenk Özmutlu und Seda Özmutlu. Multitasking Information Seeking and Searching Processes. *Journal of the American Society for Information Science and Technology*, 53:639–652, 2002.

[Spi98] Amanda Spink. Modeling Users' Successive Searches in Digital Environments. Technischer Bericht, D-Lib Magazine, 1998. http://www.dlib.org/dlib/april98/04spink.html, Letzter Zugriff: 27. März 2012.

[STZ05] Xuehua Shen, Bin Tan und ChengXiang Zhai. Implicit User Modeling for Personalized Search. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, Seiten 824–831, 2005.

[ZM06] Yuye Zhang und Alistair Moffat. Some Observations on User Search Behavior, 2006.