Bauhaus-Universität Weimar
Fakultät Medien
Studiengang Medieninformatik

Suchergebnisdiversifizierung durch Anfragesegmentierung

Bachelorarbeit

Felix Lauer

Gutachter: Prof. Dr. Benno Stein
Betreuer: Dr. Matthias Hagen

Datum der Abgabe: 28. Februar 2013

Selbstständigkeitserklärung

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

Weimar, den 28. Februar 2013

.....
Felix Lauer

Inhaltsverzeichnis

1 Einleitung **1**

2 Grundlagen und verwandte Arbeiten **3**

2.1 Anfrageambiguität und Ergebnisdiversifizierung 3

2.1.1 Anfrageambiguität 4

2.1.2 Suchergebnisdiversifizierung 6

2.1.3 TREC Web Track 7

2.1.4 Bemerkungen 8

2.2 Anfragesegmentierung 9

2.2.1 Verfahren zur automatischen Segmentierung von Anfragen . . 9

2.2.2 Bemerkungen 11

2.3 Retrievalmaße 12

3 Uneinigkeit menschlicher Segmentierer **16**

3.1 Segmentierung durch Menschen 16

3.2 Erstellen einer Suchergebnissammlung 18

3.3 Analyse der Suchergebnisse 19

3.3.1 Kategorisierung der Webis-QSeC-10-Anfragen 19

3.3.2 Vergleich der Suchergebnislisten 21

3.4 Schlussfolgerungen 27

4 Pipeline zur Diversifizierung von Anfrageergebnissen **28**

4.1 Längenfilter 30

4.2 Segmentierungsschritt 30

4.3 Kategoriefilter 32

4.4 Retrievalschritt 35

4.5 Ergebnisfilter 35

4.6 Diversifizierungsschritt 37

5 Evaluierung des Verfahrens **39**

5.1 Evaluierungskorpus 39

5.2 Konfiguration der Pipeline 40

5.3 Verwendete Metrik 42

5.4 Evaluierung 43

5.5 Bemerkungen 47

6 Zusammenfassung und Ausblick **48**

Literaturverzeichnis **50**

Kapitel 1

Einleitung

Moderne Suchmaschinen ermöglichen es ihren Nutzern, das immense Informationsangebot des World Wide Webs binnen Sekundenbruchteilen entsprechend ihrer Wünsche zu filtern. Mittels Anfragen drückt der Nutzer seinen Informationsbedarf aus und die Suchmaschine liefert eine Liste von passenden Dokumenten. Häufig sind Anfragen jedoch alles andere als eindeutig und können verschieden interpretiert werden. Man spricht von mehrdeutigen oder ambigen Anfragen. Je nach Interpretation sind dann andere Dokumente relevant, um den Suchenden zufrieden zu stellen. So kann beispielsweise die Anfrage `new york times square dance` als Anfordern von Artikeln der New York Times zum Thema „Square Dance“ gewertet werden, oder aber auch als Suche nach Tanzveranstaltungen auf dem Times Square in New York. Das Erkennen einer solchen ambigen Anfrage ist ein anspruchsvolles Problem. Weiterhin kann eine Suchmaschine nicht „wissen“, welche der verschiedenen Intentionen einer erkannten mehrdeutigen Anfrage dem Informationsbedarf des Nutzers entspricht. Die Suchergebnisse müssen also so angepasst werden, dass sie für jede Interpretation einer Anfrage Dokumente enthalten, die für den Suchenden relevant sein könnten.

Existierende Arbeiten gehen vorwiegend auf die Lösung dieser Probleme für Anfragen geringer Länge ein. Da diese nur ein oder zwei Wörter enthalten, ergibt sich die Mehrdeutigkeit der Anfrage häufig aus der Verwendung ambiger Begriffe. Beispielsweise kann ein Nutzer, der nach `jaguar` sucht, Informationen über die Raubkatze, den Automobilhersteller oder ein Betriebssystem erhalten wollen. Die Ambiguität längerer Anfragen ergibt sich aber, so wie im Beispiel `new york times square dance`, eher daraus, dass sie je nach der logischen Gruppierung der einzelnen enthaltenen Wörter unterschiedlich interpretieren werden können. Mehrdeutige Anfragen dieser Art finden in der bisherigen Forschung kaum Beachtung, es wird vorwiegend auf das Erkennen einzelner ambiger Wörter Wert gelegt.

Diese Arbeit stellt ein neues Verfahren vor, welches lange Anfragen auf Basis der verschiedenen möglichen Gruppierungen ihrer enthaltenen Wörter auf Ambiguität untersucht und für erkannte mehrdeutige Anfragen Ergebnisse liefert, die entsprechend der verschiedenen Intentionen angepasst wurden. Es wird gezeigt, ob dieses Verfahren im Vergleich zu einer herkömmlichen Vorgehensweise bessere Resultate liefert.

Der Rest der Arbeit gliedert sich wie folgt: Kapitel 2 stellt thematisch verwandte Forschungen und Grundbegriffe vor. Insbesondere werden Verfahren zur Suchergebnisdiversifizierung und Anfragesegmentierung sowie verschiedene Metriken zum Bewerten einer Ergebnisliste erläutert. Anschließend zeigt Kapitel 3 anhand eines Experiments, dass Menschen beim Segmentieren einer Anfrage uneinig sind. Diese Uneinigkeit kann ein Zeichen für Ambiguität sein und inspiriert die in dieser Arbeit entwickelte Vorgehensweise, auch bei automatischen Segmentierungsverfahren solche Uneinigkeiten festzustellen und auszunutzen. Kapitel 4 beschreibt dann das entwickelte Verfahren. Dieses vereint verschiedene Filterschritte, um eine lange Anfrage auf Mehrdeutigkeit zu prüfen und schließlich, für den Fall, dass Ambiguität festgestellt wurde, die Suchergebnisse entsprechend anzupassen. Wie gut letztere den Informationsbedarf des Nutzers im Vergleich zu einfachen, d.h. nicht angepassten, Suchergebnissen befriedigen, zeigt Kapitel 5 mittels eines Experiments. Im abschließenden Kapitel 6 werden die Inhalte dieser Arbeit und insbesondere die Erkenntnisse der Evaluierung zusammengefasst und kritisch betrachtet.

Kapitel 2

Grundlagen und verwandte Arbeiten

Dieses Kapitel befasst sich mit Grundbegriffen und -konzepten, die thematisch mit den Inhalten dieser Arbeit zusammenhängen. Insbesondere wird erörtert, was unter Anfrageambiguität, -segmentierung und Suchergebnisdiversifizierung zu verstehen ist. Weiterhin werden Möglichkeiten vorgestellt, Suchergebnisse aufgrund quantitativer Maße zu evaluieren.

2.1 Anfrageambiguität und Ergebnisdiversifizierung

Anfragen besitzen eine bestimmte Intention, d.h., sie sind auf das Erlangen ganz konkreter Informationen ausgerichtet. Die genaue Intention einer Anfrage zu ermitteln ist sehr schwierig, da dies häufig Kenntnis über persönliche Bedürfnisse des Anfragenstellers voraussetzt. Wird zum Beispiel die Anfrage `source of the nile` gestellt, ist nicht klar, ob sie auf die genaue Lage der Quelle des Flusses Nil abzielt, oder Informationen über das Brettspiel names „Source of the Nile“ anfordern soll. Sie hat offensichtlich mehrere mögliche Intentionen, kann von der Suchmaschine dementsprechend verschieden ausgewertet und bearbeitet werden und ist folglich als mehrdeutig oder ambig zu betrachten. Die Herausforderung besteht nun darin, Anfragen mit mehreren möglichen Intentionen automatisch zu erkennen und entsprechende Suchergebnisse zu liefern. Das Ziel ist eine Steigerung der sogenannten *Retrievalperformanz*. Dabei handelt es sich um eine abstrakte Größe zur Beschreibung der Güte einer erhaltenen Ergebnisliste. Insbesondere gilt es, den Nutzen der Ergebnisse für den Anfragesteller einzuschätzen. Je zufriedener dieser mit dem Retrieval (d.h. dem gefilterten Erhalt von Informationen) ist, desto höher ist die Performanz.

2.1.1 Anfrageambiguität

Von einer ambigen Anfrage spricht man, wenn diese mehrere verschiedene Interpretationen bezüglich der zugrunde liegenden Intentionen erlaubt. In der Fachliteratur wird in diesem Zusammenhang genauer zwischen unterspezifierten Anfragen, d.h. solchen, die auf verschiedene Teilaspekte des gleichen Grundthemas abzielen, und den tatsächlich ambigen Anfragen unterschieden [CCS09]. Diese Arbeit fokussiert das Erkennen und Verarbeiten letztgenannter. Im Folgenden werden daher einige bestehende Methoden zum Erkennen solcher Anfragen vorgestellt.

Log-basierte Verfahren

Log-basierte Verfahren beruhen auf der Auswertung der von Suchmaschinen erfassten Anfragedaten (sogenannten Logs). So werden in [CSA+09] die *Klick-Entropie* und die *Anfrage-Neuformulierung* verwendet. Bei erstgenannter wird analysiert, wie oft ein Nutzer nach dem Stellen einer Anfrage verschiedene Einträge im Suchergebnis ausgewählt hat und es können daraus Rückschlüsse auf die Ambiguität der Anfrage gezogen werden. Dies fußt auf der Annahme, dass der Informationsbedarf des Nutzers mit wenigen Einträgen nicht hinreichend oder gar nicht abgedeckt wurde und er deshalb weitere Anfrageergebnisse auswählen musste. Die Anfrage selbst könnte also mehrere Intentionen beinhalten.

Bei der *Anfrage-Neuformulierung* wird in den Log-Daten nach häufigen, kurz hintereinander ausgeführten Änderungen der ursprünglichen Anfrage gesucht. Ist ein Nutzer mit den Ergebnissen einer Anfrage nicht zufrieden, formuliert er oftmals die Anfrage neu. Dieser Umstand kann wiederum ausgenutzt werden, um die Ausgangsanfrage als ambig zu erkennen.

Ein großer Nachteil solcher einfacher Log-basierter Ansätze ist, dass eine Anfrage nicht bearbeitet werden kann, wenn sie in der Vergangenheit noch nicht oder sehr selten gestellt wurde, da nur mit den Log-Daten gearbeitet wird. Weiterhin kann man speziell mit den beiden genannten Methoden nicht zwischen ambigen und unterspezifizierten Anfragen unterscheiden.

Disambiguierungsseiten

Eine weitere Herangehensweise an das Erkennen ambiger Anfragen ist das Verwenden von Disambiguierungsseiten [SPMO10]. Dabei handelt es sich um eine Auflistung mehrdeutiger Terme und ihrer möglichen Interpretationen. Ein sehr bekannter Vertreter ist die Sammlung von Disambiguierungsseiten der Wikipedia-Enzyklopädie[1], welche die verschiedenen Bedeutungen ambiger Wikipedia-Titel enthalten.

[1] `http://en.wikipedia.org/wiki/Wikipedia:Disambiguation` (letzter Zugriff am 26.11.2012)

Damit ist es möglich, einzelne Terme auf Ambiguität zu prüfen und, für den Fall, dass sie ambig sind, Informationen über die möglichen Bedeutungen zu erhalten.

Auch diese Methode bringt Nachteile mit sich. Speziell die Wikipedia-Disambiguierungsseiten decken nur Terme ab, die auch in Wikipedia-Titeln vorkommen. Dementsprechend wird ein mehrdeutiger Term nicht erkannt, wenn in der Wikipedia keine entsprechendenn Artikel existieren. Weiterhin kann natürlich kein Anspruch auf Vollständigkeit erhoben werden, da die Disambiguierungsseiten nicht zwangsläufig alle möglichen Bedeutungen eines Terms abdecken. Nicht zuletzt muss erwähnt werden, dass hierbei nur einzelne Terme disambiguiert werden können. Umfasst eine Anfrage also mehr als ein Wort, verliert das Verwenden von Disambiguierungsseiten an Genauigkeit, da die Ambiguität der Anfrage nicht mehr nur einzelnen Wörtern innewohnt, sondern zusätzlich aus deren Verknüpfung hervorgeht.

Clarity Scores

Einen algorithmisch etwas komplexeren Weg zeigen die Autoren Cronen-Townsend et al. auf [CTC02]. Hier wird anhand von Sprachmodellen, im Speziellen Unigrammen, ein sogenannter *clarity score* errechnet, der als Indikator für die Klarheit einer Anfrage dient. Ist der *clarity score* einer Anfrage klein, ist diese möglicherweise ambig. Das verwendete Unigramm-Sprachmodell ordnet einzelnen Termen in einem bestimmten Kontext eine Wahrscheinlichkeit des Auftretens zu. Für den Aufbau des Modells dienen die in einer Dokumentsammlung auftretenden Themen (in diesem Fall thematisch zusammengehörige Sinneinheiten innerhalb der Sammlung). Zur Berechnung des eigentlichen *clarity scores* wird dabei sowohl ein Sprachmodell über den Themen aller Dokumente als auch über den im Anfrageergebnis enthaltenen aufgebaut. Setzt man nun die Entropien der beiden erhaltenen Modelle ins Verhältnis, erhält man ein Maß für die Klarheit der Anfrage: Sind die Wahrscheinlichkeiten in beiden Modellen sehr ähnlich verteilt, hat die Anfrage einen großen Teil der in der Dokumentensammlung enthaltenen Themen als Ergebnis geliefert und ist somit weniger eindeutig oder „klar“ als bei sehr unterschiedlichen Verteilungen. Somit können Rückschlüsse auf die Ambiguität einer Anfrage gezogen werden. Dieser Ansatz ist aufgrund der aufgebauten Sprachmodelle stark von der zugrunde liegenden Dokumentensammlung abhängig. So kann eine Anfrage im Bezug auf eine Kollektion einen sehr geringen *clarity score* erzielen und somit ambig sein, während sie bezüglich einer anderen eindeutig erscheint.

2.1.2 Suchergebnisdiversifizierung

Nach dem Erkennen einer ambigen Anfrage ist das richtige Auswählen der Ergebnisse und das entsprechende Anpassen der Ergebnisliste an die verschiedenen Intentionen der Anfrage ein wichtiger Schritt, um die Performanz des Retrievals zu erhöhen, d.h. für den Nutzer möglichst relevante Resultate zu liefern. Diese Anpassung bezeichnet man als Diversifizierung der Suchergebnisse entsprechend der Breite an Informationen, die eine ambige Anfrage anfordert. Eine formalere Definition liefern Santos et. al. [SPMO10]:

Finde zu einer gegebenen Anfrage `q` eine sortierte Liste aus Dokumenten `R(q)` mit maximaler Relevanz im Bezug auf `q` und minimaler Redundanz im Bezug auf die Abdeckung der einzelnen in `q` enthaltenen Aspekte.[2]

Dass durch Diversifizierung bessere Suchergebnisse erzielt werden können, wurde durch zahlreiche Experimente belegt und ist der Konsens beteiligter Forscher [SMO11, SPMO10]. Im Folgenden werden wichtige Erkenntnisse und aktuell verwendete Verfahren zur Diversifizierung kurz vorgestellt.

Selektive Diversifizierung

In einer grundlegenden Arbeit [SMO10] ermitteln die Autoren, in welchen Fällen überhaupt diversifiziert werden sollte. Es gilt immer zu beachten, dass eindeutige Anfragen eine reine, nach Relevanz sortierte Ergebnisliste liefern sollten, während ambige Anfragen von einer den beinhalteten Intentionen entsprechend angepassten Liste profitieren können. Es ist daher entscheidend, Diversifizierung nicht auf alle Anfrageergebnisse anzuwenden, sondern eine Anfrage vor dem Stellen auf Ambiguität zu prüfen. Diesem Paradigma folgend stellen die Autoren ein Verfahren vor, welches erkennt, ob und zu welchem Grad Ergebnisse diversifiziert werden sollten. Dabei wird anhand verschiedener Anfrageeigenschaften entsprechend eines beliebigen Diversifizierungsverfahrens ein sogenannter Tradeoff-Parameter maschinell gelernt, der linear zwischen Diversifizierung und einfacher Relevanz wichtet. So könne nach Aussage der Autoren auf Grundlage einiger Beispielanfragen mit von Menschen beurteilter Ambiguität ein stabiles und performantes Verfahren gefunden werden.

[2] Aus dem englischen Original übersetzt: *Given a query `q`, retrieve a ranking of documents `R(q)` with maximum relevance with respect to `q` and minimum redundancy with respect to its coverage of the possible aspects underlying `q`.*

Diversifizierung unter Berücksichtigung der Ausrichtung einer Anfrage

Die Autoren Santos et al. stellen mit der *intent-aware diversification* ein Verfahren vor, welches die verschiedenen Aspekte einer Anfrage zu identifizieren versucht und anhand eines gelernten Modells die Ausrichtung der Anfrage erkennt, um so ein geeignetes Retrievalmodell auszuwählen [SMO11]. Die Ausrichtung einer Anfrage kann auch als „Art“ oder „Zweck“ gesehen werden. Es wird beispielsweise unterschieden, ob sie dem tatsächlichen Erlangen von Informationen zu einem bestimmten Thema (*informational*, beispielsweise `where can i buy books?`), dem Finden (bzw. der Navigation zu) einer bestimmten Seite (*navigational*, z.B. `bookstore.com`) oder dem Ausführen einer bestimmten Aktion (*transactional*, beispielsweise `buy c++ for java programmers`) dienen. Anhand der erkannten Ausrichtung wird ein geeignetes Verfahren zum Erlangen relevanter Dokumente ausgewählt. Die Diversifizierung findet hier also schon vor dem eigentlichen Stellen der Anfrage statt.

Explizite Diversifizierung durch aspektbezogene Anfragen

Eine weitere Möglichkeit zur Ergebnisdiversifizierung wird in [SPMO10] erörtert. Hier werden, ähnlich dem oben vorgestellten Ansatz, zunächst Aspekte einer Anfrage ermittelt. Die Autoren verwenden zur Validierung ihrer Methode eine manuell erstellte Sammlung aus Anfragen mit von Menschen ermittelten Aspekten, schlagen für die Praxis aber das Verwenden von Log-basierten Techniken wie der *Anfrage-Neuformulierung* oder den Gebrauch von *Disambiguierungsseiten* vor. Ausgehend von den in der Anfrage enthaltenen Aspekten werden dann explizit entsprechende Anfragen gestellt, die Dokumente für die finale Ergebnisliste liefern sollen. Die für diese Aufgaben notwendigen Verfahren werden im von den Autoren eingeführten *xQuAD* (*eXplicit Query Aspect Diversification*)-Framework zusammengefasst.

2.1.3 TREC Web Track

Da das Erkennen und Auflösen ambiger Anfragen ein Bereich ist, in dem noch viel geforscht werden kann und sollte, nahmen die Organisatoren der etablierten *Text REtrieval Converence (TREC)* im Jahr 2009 den sogenannten *Diversity Task* in ihren *Web Track* auf. Der *Web Track* ist ein jährlich stattfindender wissenschaftlicher Wettbewerb, der verschiedene Web-Retrievaltechnologien erforscht und evaluiert [CCS09]. Der eingeführte *Diversity Task* umfasst 50 Anfragen mit detaillierten Beschreibungen der jeweiligen Intentionen und mehreren Unterthemen pro Anfrage. Letztere umreißen mehr oder weniger genau, auf welche Informationen die entsprechende Anfrage abzielt, entsprechen also ihren Intentionen. Weiterhin sind sie kategorisch den oben genannten Ausrichtungen *informational*, *navigational* und *transactional* zugeordnet. Die Autoren des *Diversity Tasks* unterscheiden, entsprechend der Erläuterungen zu Beginn des Abschnitts 2.1.1, zwischen unterspezifizierten und

ambigen Anfragen. Erstgenannte erfordern eine Diversifizierung der Ergebnisse, da sie wie erwähnt auf verschiedene Aspekte eines Themas abzielen können. Als ambige Anfragen werden auch hier jene bezeichnet, die durch mehrere verschiedene Interpretationsmöglichkeiten gekennzeichnet sind [CCS09].

Wie für alle anderen Tasks des *Web Tracks* dient auch für den Diversity Task der ClueWeb09-Datensatz als Dokumentsammlung für die Evaluierung eingereichter Verfahren. Dabei handelt es sich um einen Korpus mit zwei verschiedenen Kategorien. Kategorie A bestehet aus ca. einer Milliarde Dokumenten in zehn Sprachen, während Kategorie B lediglich die ersten 50 Millionen Dokumente in englischer Sprache enthält. Die Dokumente wurden von Januar bis Februar 2009 gesammelt[3]. Im Rahmen des *Diversity Tasks* werden neben den eigentlichen Anfragen auch jeweils aktuelle Evaluierungsmaße entwickelt, die helfen sollen, die Anfrageergebnisse zu quantisieren. Auf diese Maße wird im Abschnitt 2.3 näher eingegangen.

2.1.4 Bemerkungen

Die diskutierten Verfahren stellen nur eine Teilmenge der insgesamt existierenden Methoden dar. Alle haben sich in der Evaluierungsphase nach Aussage der Autoren als tragfähig herausgestellt. Als Grundlage für die Evaluierung dienten häufig Anfragen aus dem *Diversity Task* des *Web Tracks*, welche für die Jahre 2009 bis 2011 eine Länge von drei Worten jedoch sehr selten überschreiten. Somit ist für die angeführten Verfahren lediglich nachgewiesen, dass sie für kurze Anfragen funktionieren. Wie bereits erwähnt, kann sich die Ambiguität einer langen Anfrage aber auch daraus ergeben, dass die enthaltenen Worte als unterschiedliche Sinneinheiten aufgefasst werden können. Da die erörterten Verfahren dieser Form der Ambiguität kaum bis gar nicht gerecht werden, ist es das Ziel dieser Arbeit, einen Weg zu finden, lange ambige Anfragen zu erkennen und deren Ergebnisse zu diversifizieren. Da das Gruppieren einzelner Worte zu unterschiedlichen Sinneinheiten innerhalb einer Anfrage vielversprechend erscheint, um die verschiedenen Intentionen auszudrücken, wird im folgenden Kapitel auf sogenannte Segmentierungsverfahren eingegangen.

[3] `http://www.lemurproject.org/clueweb09.php/` (letzter Zugriff am 26.11.2012)

2.2 Anfragesegmentierung

Als Segmentierung von Anfragen bezeichnet man deren Zerlegung in disjunkte Phrasen bzw. sogenannte Segmente. Viele moderne Suchmaschinen unterstützen das Segmentieren von Anfragen als Möglichkeit, einzelne Terme als zusammengehörig zu kennzeichnen und somit eine Art semantische Gruppierung der Terme zu erlauben. Die Anfragesyntax von Google, Bing, Yahoo usw. sieht für die Kennzeichnung eines Segments dabei meist ein voran- und ein nachgestelltes Anführungszeichen vor. So markierte Wortgruppen werden bei der Suche nach relevanten Dokumenten dann in der Regel nicht voneinander getrennt bzw. werden Ergebnisse, die das Segment vollständig enthalten, im oberen Bereich der Ergebnisliste angezeigt. Dies bietet, im Bezug auf lange Anfragen, offensichtlich eine Möglichkeit, die Intention einer Anfrage genauer zu spezifizieren. Man betrachte beispielsweise die von den Autoren Hagen et al. [HPSB11] angeführte und bereits in der Einleitung vorgestellte Anfrage `new york times square dance`: In ihrer unsegmentierten Form ist nicht klar, ob diese Anfrage auf Tanzveranstaltungen auf dem Times Square abzielt, oder aber Artikel der New York Times über Square Dance liefern soll. Wird die Anfrage aber segmentiert gestellt, z.B. als `"new york" "times square dance"`“ bzw. `"new york times" "square dance"`, kann ein klareres Ergebnis erzielt werden, da durch die Wortgruppierung die jeweilige Intention deutlich erkennbar ist.

2.2.1 Verfahren zur automatischen Segmentierung von Anfragen

Leider zeigt die Erfahrung, dass viele Nutzer sich der Option des Segmentierens beim Verwenden einer Suchmaschine gar nicht bewusst sind [HPSB11]. Deshalb wurden Verfahren entwickelt, um die Retrievalperformanz langer Anfragen durch automatische Segmentierung zu verbessern. Der Grundgedanke dabei ist jeweils, allen möglichen Segmentierungen einer Anfrage einen Wert entsprechend verschiedener Kriterien zuzuordnen, um so diejenigen Segmentierungen zu finden, die die voraussichtlich höchste Retrievalperformanz erzielen. Dabei sollen insbesondere möglichst lange Segmente in der gefundenen Segmentierung enthalten sein, da diese die Anfrage stärker spezifizieren.

Wikipedia-basiertes Verfahren

Diesem Ansatz folgend verwendet das in [HPSB11] vorgestellte Wikipedia-basierte Verfahren eine Sammlung von Wikipedia-Titeln mit ihren zugehörigen Disambiguierungsseiten. Weiterhin bezieht die Methode die sogenannten Google-n-Gramme[4] ein. Diese geben Auskunft darüber, wie häufig einzelne Wörter und Wortgruppen in den an Google gestellten Anfragen bis zu einem bestimmten Zeitpunkt enthalten waren. Um ein Segment zu bewerten, wird die folgende Vorschrift angewandt:

$$score(S) = \begin{cases} \sum\limits_{s \in S, |s| \geq 2} |s| \cdot weight(s) & \text{falls } weight(s) > 0 \text{ für alle } s \in S, |s| \geq 2 \\ \\ -1 & \text{sonst.} \end{cases}$$

Die Bewertung $score(S)$ einer Segmentierung S ergibt sich aus der Summe der sogenannten Gewichte $weight(s)$ aller Segmente s der Mindestlänge 2, multipliziert mit der Länge $|s|$ des jeweiligen Segments. Diese Mutiplikation gewährleistet die eingangs geforderte Bevorzugung langer Segmente. Die Berechnung des Segmentgewichts $weight(s)$ ist definiert als

$$weight(s) = \begin{cases} |s| + \max\limits_{\substack{s' \sqsubseteq s \\ |s'|=2}} freq(s') & \text{falls } s \text{ ein Wikipedia-Titel ist} \\ \\ freq(s) & \text{sonst.} \end{cases}$$

Dabei werden die Wikipedia-Titel einbezogen: Ist das Segment ein Wikipedia-Titel, so bestimmt das Verfahren dessen Gewicht als maximale Häufigkeit $\max freq(s')$ unter den im Segment enthaltenen 2-Grammen s'. Andernfalls wird die n-Gramm-Häufigkeit des gesamten Segments als Gewicht verwendet. Sollte das Segment kein Teil der Google-n-Gramme sein, erhält die gesamte Segmentierung die Bewertung -1 und ist somit ungültig.

[4] `http://en.wikipedia.org/wiki/Google_Ngram_Viewer` `http://books.google.com/ngrams` (letzter Zugriff am 26.11.2012)

Wikipedia-Titel-Verfahren

Diese in [HPBS12] als „Baseline“ vorgestellte Methode segmentiert eine Anfrage nur, wenn sie einen oder mehrere Wikipedia-Titel enthält. Weiterhin werden ausschließlich die Titel selbst als Segmente verwendet. Als Berechnungsgrundlage dient auch hier die oben genannte Wikipedia-Titel-Sammlung in Verbindung mit den Google-n-Grammen. Letztere benötigt man, um im Falle sich innerhalb der Anfrage überlappender Wikipedia-Titel für den besseren Segmentkandidaten zu entscheiden. Abermals werden in diesem Fall die im jeweiligen Titel enthaltenen 2-Gramme betrachtet. Als Gewicht dient dann die Länge des Titels multipliziert mit der höchsten enthaltenen 2-Gramm-Häufigkeit. Für alle Segmentierungen innerhalb der Überlappungsregion werden anschließend alle Segmentgewichte (analog zum Wikipedia-basierten Verfahren) aufsummiert und die Segmentierung mit der höchsten Bewertung gewählt. Einer Anfrage ohne enthaltene Wikipedia-Titel wird der Wert -1 zugewiesen.

Wikipedia-Titel-Verfahren mit Strict Noun Phrases (SNP)

Da das Wikipedia-Titel-Verfahren verhältnismäßig konservativ segmentiert (beispielsweise im Vergleich zum Wikipedia-basierten Verfahren), ergänzen die Autoren die Methode um das Zulassen sogenannter Strict Noun Phrases (SNPs) als Segmente. SNPs sind dadurch gekennzeichnet, dass sie ausschließlich Nomen, Zahlen, Adjektive und Artikel enthalten. Somit werden beispielsweise Phrasen, welche Verben oder Konjunktionen wie „und“, „oder“ und „weil“ enthalten, ausgeschlossen. Strict Nount Phrases sind verhältnismäßig einfach automatisch zu erkennen und bieten sich daher als zusätzliche Segmentkandidaten an. Abermals dienen hier die n-Gramm-Frequenzen und die Länge der SNPs als entscheidend für das Gewicht eines Segments, falls dieses kein Wikipedia-Titel ist.

2.2.2 Bemerkungen

Auch die hier diskutierten Verfahren sind nicht die einzigen ihrer Art. Sie sind alle verhältnismäßig einfach zu implementieren und zeigen gute Resultate bei der Evaluierung mit verschiedenen Metriken gegenüber anderen Verfahren [HPSB11, HPBS12]. Im späteren Verlauf dieser Arbeit dienen sie deshalb als Grundlage für das Abschätzen der Ambiguität einer Anfrage auf Grundlage ihrer Segmentierungen (siehe Abschnitt 4.2).

2.3 Retrievalmaße

Um die Güte einer erhaltenen Ergebnisliste zu quantifizieren und somit das dafür verantwortliche Verfahren zu bewerten, ist es notwendig, entsprechende Evaluierungsverfahren zu finden. In den letzten Jahren wurden zahlreiche Maße für verschiedene Zwecke und Aspekte des Informations-Retrievals erarbeitet und etabliert. Mit der Einführung des *Diversity Tasks* in den *TREC Web Track* (siehe Abschnitt 2.1.3) lag das Augenmerk dabei verstärkt auf der Entwicklung neuer Methoden zur Bewertung diversifizierter Anfrageergebnisse bzw. auf der Adaption bereits bestehender Verfahren entsprechend der neuen Anforderungen. Dieser Abschnitt stellt häufig verwendete Bewertungsmaßstäbe für diversifizierte Anfrageergebnisse, d.h. jene, die für den *Diversity Task* entwickelt wurden[5], vor. Die Aufgabenstellung an jede Methode ist dabei die gleiche:

Ermittle zu einer gegebenen, nach Relevanz sortierten Liste, wie gut die enthaltenen Ergebnisse die verschiedenen Aspekte/Intentionen der ursprünglichen Anfrage abdecken.

Als Basis dienen in der Regel eine Reihe von Anfragen in Verbindung mit einer Dokumentensammlung, wobei für jedes Dokument von Menschen beurteilt wurde, wie relevant es im Bezug auf die in den Anfragen enthaltenen Aspekte ist. Darauf aufbauend kann mit Hilfe der Evaluierungsmaße ermittelt werden, wie nah das Resultat eines Diversifizierungsverfahrens einer idealen Ergebnisliste gekommen ist. Da nicht jede Anfrage die gleiche Anzahl an Ergebnissen liefert, werden die Evaluierungsmaße in der Regel für feste sogenannte Cutoffs der Größe k (d.h. die ersten k Ergebnisse) berechnet.

Intent Aware Mean Average Precision (MAP-IA)

Die *Mean Average Precision* ist ein einfaches Maß, das für jede gestellte Anfrage den Durchschnittswert der Relevanzen der Dokumente in der jeweiligen Ergebnisliste, gemittelt über alle gestellten Anfragen, enthält. Diese einfache Variante der *Mean Average Precision* lässt sich leicht berechnen. Beispielsweise sei `source of the nile` eine Anfrage q, für die mit Hilfe eines Retrievalverfahrens die folgende Ergebnisliste ermittelt wurde:

Rang	Ergebnis	Relevanz
1	`nile-egypt.info`	1
2	`fly-with-me.com`	0
3	`famouswiki.org/nile`	1
4	`boardgames.com/source-of-the-nile`	1
5	`fun4all.com/games`	0

[5] `http://plg.uwaterloo.ca/~trecweb/2012.html` (letzter Zugriff am 25.11.2013)

In der Spalte Relevanz ist vermerkt, ob ein Ergebnis für die Anfrage relevant ist (1) oder nicht (0). Die *Average Precision (AP)* für diese Ergebnisliste ist der Mittelwert dieser Relevanzbewertungen und beträgt 0,6. Um die *Mean Average Precision* eines Retrievalverfahrens zu errechnen, werden die AP-Werte der Ergebnisse mehrerer Anfragen bestimmt und anschließend über die Anzahl aller gestellten Anfragen gemittelt.

Diese klassische Berechnung wurde modifziert, um Anfragen mit mehreren Intentionen gerecht zu werden. Agrawal et al. schlagen in diesem Zusammenhang ein weiteres Mitteln der MAP-Werte über alle in der Anfrage enthaltenen Intentionen vor [AGHI09]. Besonders im Kontext des *TREC Web Tracks* lässt sich diese Idee leicht umsetzen [VB09]. Als Intentionen können hier die zu den Hauptthemen des *Diversity Tasks* bekannten Unterthemen verwendet werden. So lässt sich zu jeder Ergebnisliste anhand der Relevanzbewertungen im Bezug auf die Teilthemen der MAP-Wert bezüglich jeder Intention berechnen und anschließend mitteln.

Intent Aware Mean Reciprocal Rank (MRR-IA)

Bei dem *Mean Reciprocal Rank* handelt es sich um ein weiteres Maß zur Bewertung eines Retrievalverfahrens. Der reziproke Rang ist das Inverse der Position des ersten relevanten Ergebnisses einer Ergebnisliste in Bezug auf die ursprüngliche Anfrage. Befindet sich kein relevantes Ergebnis in der betrachteten Liste, ist der Wert 0. Je näher das erste relevante Ergebnis an der ersten Position der Liste steht, desto höher wird der reziproke Rang. Der RR-Wert für die obige Ergebnisliste ist beispielsweise 1, da das erste relevante Ergebnis `nile-egypt.info` an erster Stelle der Liste steht. Träte das erste Ergebnis erst an zweiter oder dritte Stelle der Liste auf, wäre der RR-Wert 0,5 bzw. $0{,}\overline{3}$ und so weiter. Für die Berechnung des MRR-Werts werden die reziproken Ränge für alle gestellten Anfragen berechnet und anschließend gemittelt.

Auch für dieses Retrievalmaß wird in [AGHI09] eine Diversifikations-spezifische Modifikation vorgeschlagen. Analog zur $MAP-IA$ werden für den $MRR-IA$-Wert alle MRR-Werte für die bekannten Intentionen einer Anfrage berechnet und anschließend gemittelt.

Normalized Discounted Cumulative Gain, abhänging von α (α-nDCG)

Der *Normalized Discounted Cumulative Gain (nDCG)* vergleicht im Wesentlichen, wie nahe die betrachtete Ergebnisliste einem idealen Suchresultat kommt. Ein ideales Resultat beinhaltet alle Dokumente, für die ein Relevanzwert in Bezug auf die jeweilige Anfrage vorliegt, absteigend sortiert nach diesem Wert. Somit stehen alle relevanten Dokumente im oberen Teil der Liste. Weiterhin werden die Relevanzwerte hinsichtlich ihrer Position in der Liste durch einen sogenannten Discount-Faktor gewichtet. Dadurch ist es möglich, das Auflisten von Dokumenten mit geringer Relevanz an oberer oder das von Dokumenten mit hoher Relevanz an unterer Stelle zu bestrafen und das entsprechende Gegenteil zu belohnen. Formal ist der (nicht normalisierte) *Discounted Cumulative Gain (DCG)* einer Ergebnisliste r für den Cutoff k definiert als

$$DCG(r,k) = \sum_{i=1}^{k} \frac{2^{j(i)} - 1}{log(1+i)}.$$

Dabei ist $j(i)$ die Relevanzbewertung (z.B. $0 = irrelevant$, $1 = relevant$) des Dokuments an Stelle i der Liste r. Das Teilen durch $log(1+i)$ entspricht der Wichtung entsprechend der Position in der Liste (Discount-Faktor). Zur Normalisierung des Werts, d.h. zur Berechnung des $nDCG$, wird folgende Formel angewandt:

$$nDCG(r,k) = \frac{DCG(r,k)}{DCG(r_{ideal},k)}.$$

In diesem Kontext ist r die Ergebnisliste, deren Güte ermittelt werden soll und r_{ideal} das ideale Resultat. Je näher r an r_{ideal} liegt, desto höher ist der $nDCG$-Wert.

Um dieses Maß an das Beurteilen diversifizierender Verfahren anzupassen, schlagen Clarke et al. eine modifizierte Ermittlung der Relevanz-Werte vor [CKC+08]. Dabei ermitteln sie im Grunde für jedes Dokument der Ergebnisliste einen Wert, der ausdrückt, mit welcher Wahrscheinlichkeit ein Nutzer unter Berücksichtigung bereits vorangestellter Ergebnisse am aktuellen Dokument interessiert ist. Weiterhin wird in die Berechnung einbezogen, dass bei der Beurteilung der Dokumentrelevanzen durch Menschen immer gewisse Unstimmigkeiten und somit auch ein Potential für Fehlbewertungen liegen. Die Fehlerwahrscheinlichkeit für die Relevanzbewertung wird daher mit einem konstanten Wert α abgeschätzt (daher der Name $\alpha - nDCG$).

Diese Evaluierungsmetrik berücksichtigt das tatsächliche Interesse des Nutzers stärker, als es beispielsweise bei MAP-IA oder MRR-IA der Fall ist. Zentrale Aspekte in der Wichtung der Relevanzwerte sind die Bewertung des Vermeidens von Redundanz in der Ergebnisliste sowie die Belohnung der Einführung von Ergebnissen, die bisher nicht abgedeckte Teilthemen einer Anfrage betreffen.

Die hier vorgestellten Metriken sind Standardmaße für die Evaluierung von Ergebnislisten im Kontext des *TREC Web Tracks*. Alle wurden modifiziert, um der Bewertung diversifizierter Resultate gerecht zu werden.

Nachdem dieses Kapitel Grundbegriffe und -konzepte vorgestellt hat, auf die in dieser Arbeit Bezug genommen wird, betrachtet das folgende Kapitel, wie Anfragen von Menschen segmentiert werden.

Kapitel 3

Uneinigkeit menschlicher Segmentierer

Dieses Kapitel beschreibt die Motivation zur Entwicklung eines Verfahrens, welches die Suchergebnisse mehrdeutiger Anfragen verbessern soll. Es wird mit Hilfe eines Experiments betrachtet, wie Anfragen von Menschen segmentiert werden. Die Ergebnisse dieses Versuchs bilden die Inspiration für das in Kapitel 4 vorgestellte Verfahren zum Erkennen ambiger Anfragen auf Grundlage ihrer Segmentierungen.

3.1 Segmentierung durch Menschen

Auf der Suche nach einem Verfahren zur Verbesserung der Suchergebnisse ambiger Anfragen ergibt sich zunächst der naheliegende Gedanke, dass es nach wie vor die Nutzer einer Suchmaschine sind, die die Intention ihrer Anfrage am besten kennen. Wie in Abschnitt 2.2 beschrieben, bietet das Segmentieren einer Anfrage die Möglichkeit, semantische Gruppierungen vorzunehmen. Dementsprechend könnte man Rückschlüsse auf Ambiguität ziehen, wenn dieselbe Anfrage durch verschiedene Nutzer unterschiedlich segmentiert wird. Dies gilt insbesondere, wenn eine ähnliche Anzahl von Anfragestellern jeweils eine bestimmte Segmentierung präferiert.

Diesem Gedanken folgend wurde der in [HPSB11] beschriebene *Webis-QSeC-10* für entsprechende Experimente verwendet. Es handelt sich hierbei um eine im Jahr 2010 erstellte Sammlung aus insgesamt 53 437 Anfragen mit ihren entsprechenden Segmentierungen. Der Korpus wurde durch zufällige Auswahl und anschließendes Filtern nach verschiedenen Kriterien aus dem *AOL-Query-Log* (eine 36 389 567 Anfragen umfassende Sammlung) aufgebaut. Der Intention dieser Arbeit, die Ambiguität langer Anfragen zu analysieren, kommt zugute, dass keine der im entstandenen Korpus enthaltenen Anfragen weniger als drei Worte umfasst.

Die Autoren Hagen et al. nutzten den sogenannten *Amazon Mechanical Turk*[1], um alle Anfragen des Webis-QSeC-10 von Menschen entsprechend ihrer Präferenzen segmentieren zu lassen. Für jeden Eintrag im Korpus liegen die favorisierten Segmentierungen und die Anzahl der Mechanical-Turk-Arbeiter vor, die diese gewählt haben. Mindestens zehn Menschen beurteilten auf diese Weise, welche Segmentierung einer Anfrage sie für die beste hielten. Am Beispiel der Anfrage `new york times square dance` könnte ein möglicher Eintrag im Webis-QSeC-10 wie folgt aussehen:

```
(4) "new york times square" "dance"
(4) "new york times" "square dance"
(2) "new york" "times square" dance
```

Vier von zehn Arbeitern haben sich demnach für die erste Segmentierung, weitere vier für die zweite und zwei für die dritte entschieden. Daraus ergibt sich eine interessante Beobachtung: Menschen, von denen unter Berücksichtigung einer gewissen Fehlerquote angenommen werden kann, dass sie die Intention einer Anfrage relativ zuverlässig beurteilen können, sind nicht immer einer Meinung, wenn es um die Auswahl einer Segmentierung geht. Insbesondere Verteilungen wie die obige, bei der keine klare Mehrheit der Arbeiter eine bestimmte Zerlegung der Anfrage favorisiert, könnten demnach aufschlussreiche Informationen liefern. Möglicherweise ist die offensichtliche Uneinigkeit der menschlichen Segmentierer ein Indikator dafür, dass der Anfrage mehrere Intentionen innewohnen und sie folglich als ambig eingestuft werden kann. Für den Fall, dass die verschiedenen Segmentierungen unterschiedliche Intentionen abbilden, sollten beim Stellen dieser Segmentierungen als Anfragen an eine Suchmaschine entsprechend verschiedene Ergebnisse erzielt werden. Um diesem Gedanken nachzugehen und ihn auf Richtigkeit zu prüfen, bietet es sich an, die von den einzelnen Segmentierungen erzielten Suchergebnisse untereinander zu vergleichen.

[1] `https://www.mturk.com/mturk/welcome` (letzter Zugriff am 12.12.2012)

3.2 Erstellen einer Suchergebnissammlung

Als Vorbereitung für den Vergleich der Anfrageergebnisse wurden diese im Juli 2012 für alle möglichen Segmentierungen der im Webis-QSeC-10 enthaltenen Anfragen gesammelt.

Ergebnisse mit Bing

Als Suchmaschine für die Anfragen des gesamten Webis-QSeC-10 diente Bing[2] bzw. die Bing-API (zu diesem Zeitpunkt eine der letzten noch frei verfügbaren Programmierschnittstellen für die Suche im gesamten Web). Für den späteren Vergleich ist ein reines Betrachten der Ergebnislisten vollkommen ausreichend, weshalb auch nur diese (und nicht die entsprechenden Dokumente) gesammelt wurden. Die 53 437 Anfragen des Webis-QSeC-10 haben insgesamt 953 212 verschiedene Segmentierungen. Für jede dieser Segmentierungen wurde eine entsprechende Anfrage an Bing gestellt und die ersten 500 (sofern vorhandenen) Einträge der Ergebnisliste in einer separaten Datei gespeichert. Zur späteren Zuordnung dienten die von Hagen et al. in ihrem Korpus verwendeten *Query IDs* (eine innerhalb des Korpus' eindeutige Zahlenkombination), ergänzt um einen segmentierungsspezifischen Teil, als Namensschema. Leider konnten bis Ende Juli 2012 nur für 897 474 der 953 212 Anfragen Ergebnisse gesammelt werden. Seit Anfang August 2012 ist die Bing-API nicht mehr im vollen Umfang frei zugänglich.

Ergebnisse mit Indri

Da die Dokumente des World Wide Web einem ständigen Wandel unterliegen und Suchergebnisse daher binnen sehr kurzer Zeit veraltet sein können, verwenden Forscher häufig den in Abschnitt 2.1.3 vorgestellten ClueWeb09-Datensatz. Eine Möglichkeit, diesen zu durchsuchen, bietet die Suchmaschine Indri[3], welche neben einer umfassenden Anfragesyntax auch eine eigene Programmierschnittstelle bereitstellt. Leider ist die Geschwindigkeit, in der Ergebnisse geliefert werden, zu gering, um diese für alle Segmentierungen der im Webis-QSeC-10 enthaltenen Anfragen in einem angemessenen Zeitraum zu sammeln. Deshalb wurde nur eine Teilmenge des Webis-QSeC-10, das sogenannte Webis-QSeC-10-Training-Set, verwendet. Dieses umfasst 4 850 der insgesamt 53 437 Anfragen, deren Wortlängenverteilung der des gesamten Korpus entspricht. Für diese ca. 5 000 Anfragen existieren 86 736 mögliche Segmentierungen, welche als Anfragen an Indri gestellt wurden. Das Speichern der Ergebnislisten erfolgte analog der Ergebnissammlung mit Bing.

[2] `http://www.bing.com/` (letzter Zugriff am 12.12.2012)

[3] `http://www.lemurproject.org/clueweb09.php/index.php#Services` (letzter Zugriff am 05.01.2013)

3.3 Analyse der Suchergebnisse

Die wie in Abschnitt 3.2 beschrieben gesammelten Suchergebnisse werden im Folgenden einer Analyse unterzogen. Das Ziel ist es dabei, Rückschlüsse auf die Verwendbarkeit menschlicher Segmentierungsfavorisierungen als Indikator für die Ambiguität einer Anfrage zu ziehen. Zu diesem Zweck werden die Anfragen des Webis-QSeC-10 zunächst in verschiedene Kategorien eingeteilt.

3.3.1 Kategorisierung der Webis-QSeC-10-Anfragen

Um ein differenziertes Betrachten der Anfragen und ihrer Ergebnisse zu gewährleisten, erscheint es sinnvoll, für diese zunächst genauere Unterscheidungen zu treffen. Zum einen dient die Länge der Anfrage als Hauptdifferenzierungsmerkmal. Zum anderen sind für die Analyse insbesondere jene Anfragen interessant, bei denen sich die menschlichen Segmentierer „uneinig“ waren, d.h., bei denen mehrere verschiedene Segmentierungen mit gleicher oder ähnlicher Häufigkeit favorisiert werden (vgl. Beispiel in Abschnitt 3.1). Deshalb werden für diese Art der Unterscheidung, die man auch als „Sicherheit in der Anfragesegmentierung“ bezeichnen könnte, drei verschiedene Kategorien eingeführt. Anfragen, bei denen die Segmentierer „uneinig“ waren, fallen in die Kategorie *uncertain*, da keine Segmentierung besonders bevorzugt wurde. Weiterhin existieren die Kategorien *semi*, welche Anfragen enthält, deren Segmentierungen klarer, aber noch nicht eindeutig zu Gunsten einer speziellen priorisiert wurden, und *certain*. Letztgenannte beinhaltet Anfragen, für die die menschlichen Segmentierer eine bestimmte Segmentierung im Konsens favorisierten. Die Einteilung der Anfragen erfolgt auf Grundlage der im Webis-QSeC-10 gegebenen Daten, davon ausgehend, dass zehn Segmentierer ihren Favoriten gewählt haben.

Tabelle 1 stellt schematisch die verschiedenen Stimmverteilungen ihren zugeordneten Kategorien gegenüber. Die Notation ist folgendermaßen zu lesen: Eine Verteilung von `8 : 1 : 1` bedeutet, dass acht von zehn Segmentierern sich für dieselbe Segmentierung einer Anfrage entschieden haben, während die zwei übrigen jeweils eine andere Segmentierung vorzogen. Die Anzahl der Stimmen ist dabei, wie auch im Webis-QSeC-10, in absteigender Reihenfolge angegeben. Daraus ergibt sich automatisch eine Art Bewertung der Segmentierungen.

Kategorie	Stimmverteilung	Vorkommen im Korpus
uncertain	5 : 5	1 150 (2,307 %)
	5 : 4 : 1	2 983 (5,986 %)
	4 : 4 : 2	987 (1,981 %)
	4 : 4 : 1 : 1	878 (1,762 %)
	4 : 3 : 2 : 1	2 488 (4,993 %)
	4 : 3 : 1 : 1 : 1	783 (1,571 %)
	4 : 2 : 2 : 2	367 (0,737 %)
	4 : 2 : 2 : 1 : 1	799 (1,603 %)
	4 : 2 : 1 : 1 : 1 : 1	501 (1,005 %)
	3 : 3 : 3 : 1	345 (0,692 %)
	3 : 3 : 2 : 1 : 1	712 (1,429 %)
	3 : 3 : 1 : 1 : 1 : 1	228 (0,482 %)
	3 : 2 : 2 : 2 : 1	347 (0,458 %)
	3 : 2 : 2 : 1 : 1 : 1	677 (1,359 %)
	3 : 2 : 1 : 1 : 1 : 1 : 1	343 (0,688 %)
	3 : 1 : 1 : 1 : 1 : 1 : 1 : 1	53 (0,106 %)
	2 : 2 : 2 : 2 : 2	16 (0,032 %)
	2 : 2 : 2 : 2 : 1 : 1	129 (0,259 %)
	2 : 2 : 2 : 1 : 1 : 1 : 1	195 (0,391 %)
	2 : 2 : 1 : 1 : 1 : 1 : 1 : 1	107 (0,215 %)
	2 : 1 : 1 : 1 : 1 : 1 : 1 : 1 : 1	41 (0,082 %)
	1 : 1 : 1 : 1 : 1 : 1 : 1 : 1 : 1 : 1	6 (0,012 %)
		Σ 14 135 (28,366 %)
semi	6 : 4	2 386 (4,788 %)
	5 : 3 : 2	2 120 (4,254 %)
	5 : 3 : 1 : 1	1 832 (3,677 %)
	5 : 2 : 2 : 1	1 425 (2,860 %)
	5 : 2 : 1 : 1 : 1	845 (1,700 %)
	4 : 1 : 1 : 1 : 1 : 1 : 1	78 (0,157 %)
		Σ 8 686 (17,432 %)
certain	10 : 0	2 736 (5,491 %)
	9 : 1	4 451 (8,932 %)
	8 : 2	3 110 (6,241 %)
	8 : 1 : 1	2 923 (5,866 %)
	7 : 3	2 723 (5,465 %)
	7 : 2 : 1	3 878 (7,782 %)
	7 : 1 : 1 : 1	1 105 (2,218 %)
	6 : 3 : 1	3 388 (6,799 %)
	6 : 2 : 1 : 1	2 158 (4,331 %)
	6 : 1 : 1 : 1 : 1	385 (0,773 %)
	5 : 1 : 1 : 1 : 1 : 1	152 (0,305 %)
		Σ 27 009 (54,202 %)

Tabelle 1: Anfragekategorien und entsprechende Stimmverteilungen mit ihren Auftrittshäufigkeiten im Webis-QSeC-10

Ausschlaggebend für die Kategorisierung nach obigem Schema ist die Forderung, dass die Differenz der Anzahl der Stimmen der beiden ersten Segmentierungen gering sein soll, um eine Anfrage als *uncertain* einzustufen. Dem gegenüber steht die Prämisse, dass als *certain* kategorisierte Anfragen sich durch eine große Differenz der Stimmzahl der beiden ersten Segmentierungen auszeichnen.

Die auf diese Weise kategorisierten Anfragen und ihre zugehörigen Suchergebnisse bilden die Grundlage für die weitere Analyse.

3.3.2 Vergleich der Suchergebnislisten

Die Kategorisierung der Anfragen des Webis-QSeC-10 und ihrer entsprechenden Ergebnisse erlaubt es, diese differenziert zu analysieren. Es stellt sich insbesondere die Frage: Liefern die Segmentierungen einer Anfrage, die als *uncertain* kategorisiert wurde, häufig verschiedene Ergebnisse? Der Fakt, dass die menschlichen Segmentierer verschiedene Segmentierungen mit gleicher Häufigkeit favorisiert haben, zeigt offenbar, dass man die Anfrage auf verschiedene Weisen interpretieren kann, sie also eventuell mehrere Intentionen hat.

Zur Überprüfung dieser Vermutung werden die gesammelten Suchergebnisse herangezogen. Da eine ambige Anfrage unterschiedliche Intentionen beinhaltet, sollten je nach betrachteter Intention andere Suchergebnisse relevant sein. Will man mit der Anfrage `new york times square dance` Informationen über Tanzveranstaltungen auf dem New York Times Square erlangen, könnten in der Ergebnisliste die Internetauftritte von Tanzvereinen der Stadt New York o.ä. angeführt sein. Hat man aber mit derselben Anfrage die Absicht, Artikel der New York Times zum Thema „Square Dance“ zu finden, erwartet man eher Archiveinträge der Zeitung unter den Suchergebnissen. Wenn sich die Intentionen also durch verschiedene Segmentierungen der Anfrage abbilden lassen, dann ist zu erwarten, dass sich die zu den Segmentierungen erhaltenen Ergebnislisten deutlich unterscheiden. Insbesondere kann sich die „Unsicherheit“ der menschlichen Segmentierer als guter Indikator für ambige Anfragen erweisen, wenn sich die Ergebnislisten der Segmentierungen der als *uncertain* kategorisierten Anfragen häufig besonders stark unterscheiden. Demnach bietet sich als nächster Schritt an, die Ergebnislisten der Segmentierungen der kategorisierten Anfragen zu vergleichen.

Auswahl bestimmter Segmentierungen

Zum Vergleich der Ergebnisse werden pro Anfrage zwei Segmentierungen ausgewählt. Natürlich können ambige Anfragen mehrere verschiedene Bedeutungen haben, dementsprechend müssten auch mehrere verschiedene Segmentierungen bestimmt werden. Im Sinne der Übersichtlichkeit, der grundlegenden Art dieser Arbeit und des Fakts, dass ambige Anfragen zwar beliebig viele Intentionen haben können, jedoch immer mindestens zwei besitzen müssen, beruhen alle folgenden Experimente dieser Art auf den zwei „besten“ Segmentierungen einer Anfrage. Zur Bestimmung der Güte bestimmter Segmentierungen dienen neben den Favorisierungen der menschlichen Segmentierer die in Abschnitt 2.2 erörterten Google-n-Gramme als Entscheidungshilfe. Für alle im Webis-QSeC-10 vorkommenden n-Gramme ist die Häufigkeit des Auftretens im Google Index von 2006 bekannt. Davon ausgehend ist der Vergleich bzw. die Ordnung zweier Segmentierungen einer Anfrage wie folgt definiert:

1. Beim Vergleich zweier Segmentierungen mit *verschiedener* Anzahl der Stimmen gewinnt jene, welche von den meisten Segmentierern gewählt wurde.
2. Beim Vergleich zweier Segmentierungen mit *gleicher* Anzahl der Stimmen gewinnt jene, bei der die Summe der um die jeweilige Segmentlänge potenzierten Google-n-Gramm-Häufigkeiten aller enthaltenen Segmente größer ist.

Diese einfachen Regeln lassen sich rekursiv auf die im Webis-QSeC-10 vorkommenden Anfragen anwenden, bis zu jeder von ihnen zwei bevorzugte Segmentierungen bestimmt sind. Das Potenzieren der Google-n-Gramm-Häufigkeit um die Länge des Segments in Regel 2 dient als Wichtung. Längere Segmente, die durchschnittlich deutlich weniger in den Anfragen der Nutzer auftauchen und somit geringere n-Gramm-Häufigkeiten besitzen, haben auf diese Weise die Chance, besser bewertet zu werden als kurze Segmente mit großer Häufigkeit. Betrachtet man nun beispielsweise die hypothetische Anfrage `good bowling center weimar`, wäre die unten stehende Auswahl der besten Segmentierungen denkbar.

Die Häufigkeiten der in der Anfrage enthaltenen n-Gramme seien:

`good`	50 000
`good bowling`	300
`bowling`	700 000
`bowling center`	40 000
`center`	50 000
`weimar`	60 000
`center weimar`	2 000

Es seien weiterhin die Favorisierungen der menschlichen Segmentierer wie folgt verteilt:

```
(8) "good bowling center" "weimar"
(1) "good bowling" "center weimar"
(1) "good" "bowling center" "weimar"
```

Vergleicht man nun die beiden ersten Segmentierungen anhand obiger Regeln, steht `"good bowling center" "weimar"` als erste der beiden bevorzugten Segmentierungen fest, da sie mit acht von zehn Stimmen mehr Favorisierungen hat. Beim Vergleich der zweiten Segmentierung mit der dritten tritt die zweite Regel in Kraft, da die Favorisierungshäufigkeit bei beiden 1 beträgt. Das Summieren der um die Segmentlänge potenzierten n-Gramm-Häufigkeiten der enthaltenen Segmente ergibt für die zweite Segmentierung 4 090 000, für die dritte 1 600 110 000. Die beiden besten Segmentierungen sind demnach die erste und die dritte.

Die beiden so gewählten Segmentierungen beinhalten potentiell zwei verschiedene Intentionen der Anfrage. Demnach können ihre Suchergebnisse im Folgenden verglichen und auf Ähnlichkeit geprüft werden.

Einschränkung der Ergebnislisten

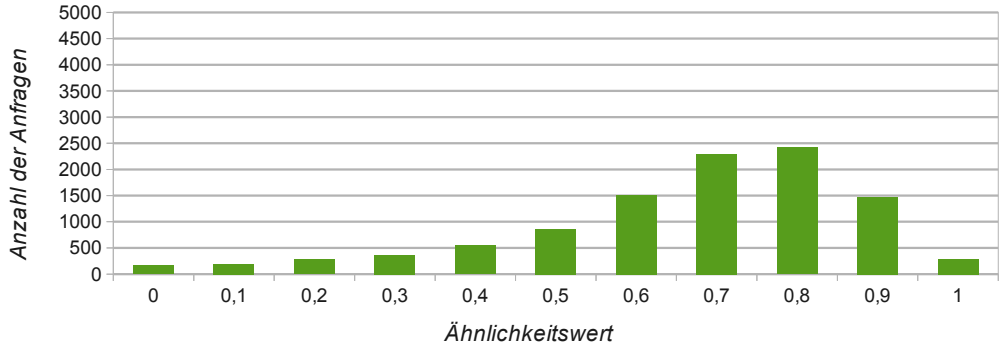
Die Ergebnislisten der nach obigem Verfahren ausgewählten Segmentierungen einer Anfrage werden zur besseren Vergleichbarkeit auf eine gemeinsame Länge beschränkt. In einer alltäglichen Situation, in der ein Nutzer eine Anfrage an eine Suchmaschine stellt, betrachtet dieser erfahrungsgemäß lediglich die ersten fünf bis zehn Einträge der erhaltenen Ergebnisliste. Folglich ist es auch beim Vergleichen der Ergebnisse absolut ausreichend, die Betrachtung auf diese Anzahl einzugrenzen. An dieser Stelle sei angemerkt, dass beim Stellen der Segmentierungen als Anfragen an Bing und Indri nicht immer zehn Ergebnisse geliefert wurden. Da ein fairer Vergleich auf dieser Grundlage nicht möglich ist, entfallen diese Segmentierungen aus den nachfolgenden Betrachtungen.

Verfahren zum Vergleich der Ergebnislisten

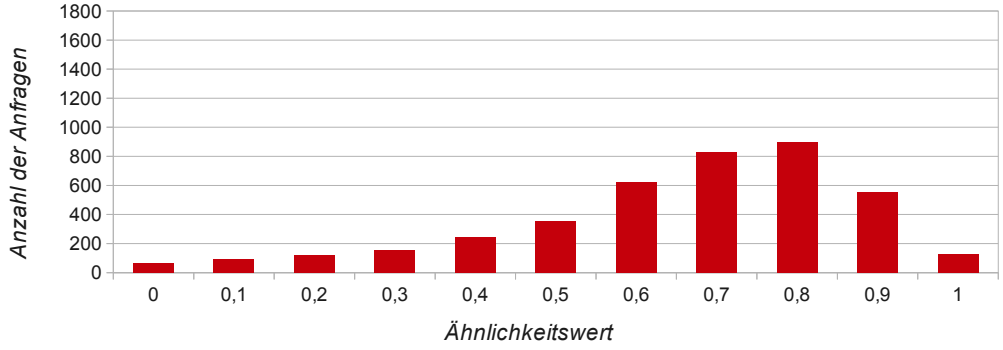
Nach der Auswahl der beiden besten Segmentierungen einer Anfrage wird ein Ähnlichkeitswert S für die ersten fünf bzw. zehn Einträge der Ergebnislisten berechnet:

$$S(s_1,\ s_2,\ k) = \frac{\mid r(s_1, k)\ \cap\ r(s_2, k) \mid}{k}.$$

Dabei stehen s_1 und s_2 für die beiden besten Segmentierungen und die Funktion $r(s_i, k)$ liefert die ersten k Einträge der Ergebnisliste der jeweiligen Segmentierung.



(a) Anfragen der Kategorie *certain*, insgesamt 10 409



(b) Anfragen der Kategorie *uncertain*, insgesamt 4 056

Abbildung 1: Verteilung der Ähnlichkeitswerte der ersten zehn Bing-Ergebnisse der besten Segmentierungen für Drei-Wort-Anfragen

Die Mächtigkeit der Schnittmenge, d.h. die Anzahl der Einträge, die beide Listen gemeinsam haben, wird auf die Anzahl der betrachteten Ergebnisse normalisiert. Wenn also für zwei betrachtete Segmentierungen fünf von zehn Einträgen übereinstimmen, beträgt der Ähnlichkeitswert 0,5, bei zwei von zehn Einträgen 0,2 usw. Anders ausgedrückt sind die Ergebnislisten der Segmentierungen zu 50% bzw. zu 20% gleich.

Verteilung der Ergebnisüberlappung der besten Segmentierungen

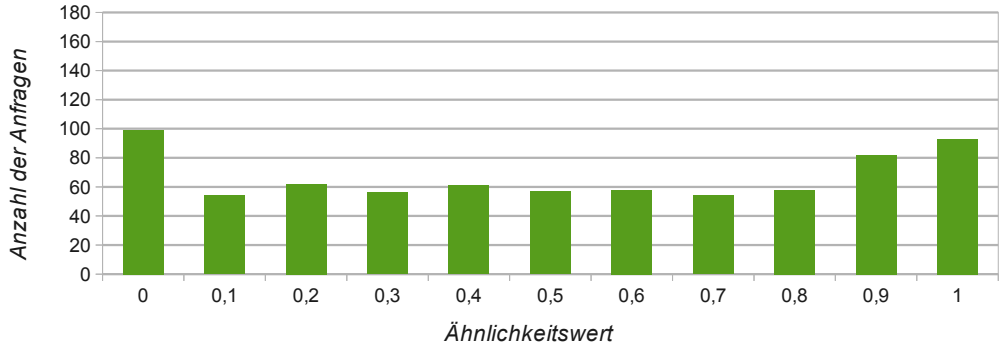
Berechnet man nun die Ähnlichkeitswerte der Ergebnislisten der beiden besten Segmentierungen für die, wie in Abschnitt 3.3.1 beschrieben, kategorisierten Anfragen des Webis-QSeC-10, lassen sich interessante Tendenzen beobachten.

Abbildung 1 zeigt exemplarisch, wie die Ähnlichkeitswerte der ersten zehn Bing-Suchergebnisse der besten Segmentierungen von Drei-Wort-Anfragen verteilt sind. Dabei wird die Kategorisierung als *certain* bzw. *uncertain* einbezogen. Je geringer der Ähnlichkeitswert S ist, desto verschiedener sind die Ergebnislisten. Aus der Graphik lässt sich erkennen, dass ein nicht unerheblicher Anteil der Ergebnisvergleiche einen Ähnlichkeitswert von weniger als 0,5 erzielt hat. Das bedeutet in diesem Fall, dass weniger als fünf der ersten zehn Ergebnisse übereinstimmen. Offenbar können mit verschiedenen Segmentierungen derselben Anfrage deutlich unterschiedliche Resultate erzielt werden. Dies erlaubt den wichtigen Schluss, dass es prinzipiell möglich ist, die Intentionen einiger mehrdeutiger Anfragen mit Hilfe verschiedener Segmentierungen auszudrücken.

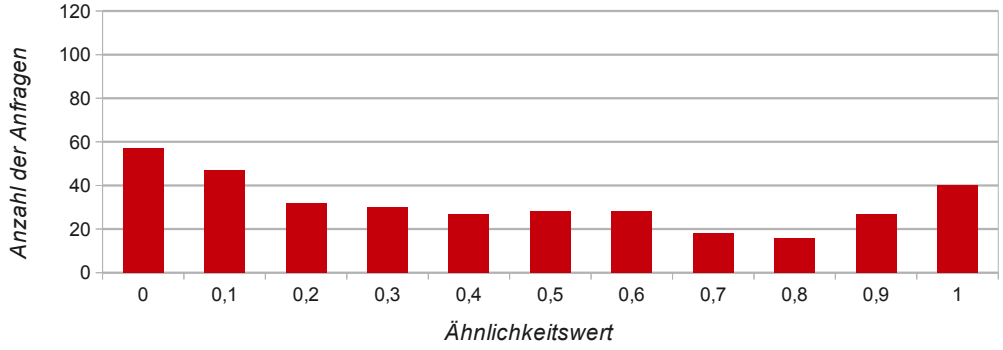
Weiterhin lässt sich in den Diagrammen eine Häufung der Ähnlichkeitswerte im Bereich 0,6 bis 0,9 erkennen. Von der Vermutung ausgehend, dass menschliche Segmentierer die Intentionen einer Anfrage besser erkennen können als automatische Verfahren und demnach Anfragen der Kategorie *uncertain* möglicherweise ambig sind, wäre eine andere Verteilung in diesem Bereich wünschenswert: Die Häufung der Ähnlichkeitswerte sollte für unsichere Anfragen idealerweise in einem Bereich unter 0,5 liegen. Dies würde anzeigen, dass die Uneinigkeit der Menschen bei den Segmentierungen der Anfragen häufig auch verschiedene Ergebnisse liefert. Somit wäre ein guter Indikator für das Erkennen mehrdeutiger Anfragen gefunden.

Die für diese Betrachtungen verwendeten Ergebnisse stammen von der properitären Suchmaschine Bing. Diese verwendet möglicherweise Techniken zur Disambiguisierung o.ä. und kann demnach die obige Verteilung beeinflussen. Um falsche Schlüsse auf dieser Grundlage zu vermeiden, ist ein Betrachten der Ähnlichkeitswertverteilungen für die mit Indri erzielten Ergebnisse (vgl. Abschnitt 3.2) hilfreich. Sie ist (abermals exemplarisch für Drei-Wort-Anfragen) in Abbildung 2 dargestellt. Da nur eine Teilmenge des gesamten Webis-QSeC-10 (etwa 5 000 Anfragen und ihre entsprechenden Segmentierungen) an Indri gestellt wurde, ist die Magnitude der Häufigkeiten der Ähnlichkeitswerte deutlich kleiner als in den Betrachtungen für die mit Bing gestellten Anfragen. Trotzdem können sie auf Grund der ähnlichen Verteilung der Wortlängen unter den Anfragen als Vergleichswert herangezogen werden.

Bei der Analyse der Abbildung 2 fällt auf, dass eine Häufung der Werte wie beim Stellen der Anfragen mit Bing nicht mehr im Intervall 0,6 bis 0,9, sondern in den Randbereichen (Abbildungen 2(a) und 2(b)) auftritt. Die unterschiedliche Verteilung der Häufigkeiten im Vergleich zu den Bing-Resultaten spricht dafür, dass Bing die gestellten Anfragen auf verschiedene Weisen modifiziert und somit die Ergebnisse verfälscht werden. Dennoch lässt sich auch auf Basis der mit Indri gewonnenen Ergebnisse kein nennenswerter Unterschied der Verteilungen zwischen den Kategorien *certain* und *uncertain* feststellen.



(a) Anfragen der Kategorie *certain*, insgesamt 734



(b) Anfragen der Kategorie *uncertain*, insgesamt 350

Abbildung 2: Verteilung der Ähnlichkeitswerte der ersten zehn Indri-Ergebnisse der besten Segmentierungen für Drei-Wort-Anfragen

Ähnliche Tendenzen ließen sich innerhalb der ausführlichen Experimentierphase dieser Arbeit auch für Anfrage größerer Länge beobachten. Da die Verteilung der Werte somit weder von der Anzahl der Wörter noch von der Kategorisierung abhängig zu sein scheint, muss die eingangs gestellte zentrale Frage, ob die Segmentierungen einer Anfrage, die als *uncertain* kategorisiert wurde, häufig verschiedene Ergebnisse liefern im Bezug auf die vorliegenden Daten verneint werden.

3.4 Schlussfolgerungen

Es ist nicht erkennbar, dass eine Uneinigkeit der Mechanical-Turk-Arbeiter im Bezug auf das Segmentieren einer Anfrage häufig große Unterschiede beim folgenden Vergleich der Ergebnisse der beiden besten Segmentierunen bewirkt. Im Gegenteil: Es scheint in Bezug auf die Verteilung der Ähnlichkeitswerte keine Rolle zu spielen, ob sich die menschlichen Segmentierer uneinig sind oder nicht.

Dennoch ergeben sich aus den Betrachtungen dieses Kapitels verschiedene Forschungsparameter für den weiteren Verlauf der Arbeit. Zum einen unterscheiden sich die Ergebnisse für einen nicht unerheblichen Anteil der Segmentierungspaare. Demnach ist es wünschenswert, solche Paare für das Auflösen ambiger Anfragen besonders in Betracht zu ziehen. Da durch Menschen getroffene Bewertungen der Segmentierungen einer Anfrage im alltäglichen Umgang mit Suchmaschinen natürlich nicht vorliegen, sollten für das Finden der beiden besten Segmentierungen die in 2.2 vorgestellten Verfahren einbezogen werden. Des Weiteren erscheint die Kategorisierung der Anfragen in *certain*, *semi* und *uncertain* nach wie vor als durchaus sinnvoll. Auf diese Weise kann ein Filtern der Anfragen und die nachfolgende ausschließliche Betrachtung unsicherer Kandidaten später überflüssige Retrievalschritte vermeiden. Natürlich können auch hier nicht die Bewertungen menschlicher Segmentierer einbezogen werden. Deshalb ist es wichtig, die Kategorisierung auf Grundlage der genannten automatischen Segmentierungsverfahren vorzunehmen.

Die Quintessenz dieses Kapitels und gleichzeitige Motivation für die Suche nach einer Verbesserung der Ergebnisse ambiger Anfragen auf Grundlage ihrer Segmentierungen ist die folgende: Es gibt Anfragen, bei denen auch Menschen nicht eindeutig eine bestimmte Segmentierung favorisieren. Auch wenn dies keinen Unterschied im Bezug auf die Ähnlichkeit der Resultate verglichen mit sicheren Anfragekandidaten macht, so ist es dennoch denkbar, dass sich automatische Segmentierungsverfahren im Hinblick auf die Suchergebnisse als zuverlässigere Grundlage erweisen. Dementsprechend beschäftigt sich das folgende Kapitel mit der Frage, ob diese Verfahren zum Finden unsicherer Anfragekandidaten (und somit potentiell ambiger Anfragen) geeignet sind und inwiefern das Gesamtergebnis auf dieser Basis verbessert werden kann.

Kapitel 4

Pipeline zur Diversifizierung von Anfrageergebnissen

Dieses Kapitel beschreibt das im Rahmen der Arbeit entwickelte Verfahren zur Verbesserung der Ergebnisse ambiger Anfragen auf Grundlage ihrer Segmentierungen. Es werden die in Abschnitt 2.2 vorgestellten Segmentierungsverfahren zum Finden der beiden „besten“ Segmentierungen verwendet und eine einfache Diversifizierungsmethode auf die Ergebnisse angewandt.

Der Prozess zur Ergebnisdiversifizierung von Anfragen unter Berücksichtigung ihrer Segmentierungen umfasst eine feste lineare Verkettung einzelner Teilschritte. Deshalb bietet sich die Beschreibung als „Pipeline“ an. Das vorgeschlagene Konzept vereint eine Reihe von Prä- und Post-Retrievalschritten zur Filterung und Verarbeitung von Anfragen, für die sich eine Ergebnisdiversifizierung lohnen könnte.

Abbildung 3 stellt diesen Prozess schematisch dar. Eine Anfrage q wird in sechs konsekutiven Schritten verarbeitet. Zunächst prüft ein *Längenfilter*, ob die Anfrage eine Mindestzahl an Wörtern beinhaltet, um zu kurze Anfragen schon frühzeitig auszuschließen. Anschließend werden auf Grundlage eines ausgewählten Segmentierungsverfahrens die beiden „besten“ Segmentierungen von q bestimmt. Sie sind für das spätere Retrieval unerlässlich. Im folgenden Schritt, dem *Kategoriefilter*, wird die Anfrage einer der in Abschnitt 3.3.1 vorgestellten Kategorien zugeordnet und nur weiter verarbeitet, wenn sie als *uncertain* erkannt wurde. Der *Retrieval*-Schritt ermittelt dann Ergebnislisten zu den beiden Segmentierungen der Anfrage. Der *Ergebnisfilter* überprüft, ob ein Mischen beider Listen Änderungen gegenüber den Ergebnissen der besten Segmentierung liefert. Wenn dies der Fall ist, mischt der finale Teilschritt *Diversifizierung* die Ergebnisse der gefilterten und verarbeiteten Anfrage.

Die folgenden Abschnitte dieses Kapitels befassen sich mit der ausführlichen Beschreibung der einzelnen Teilschritte.

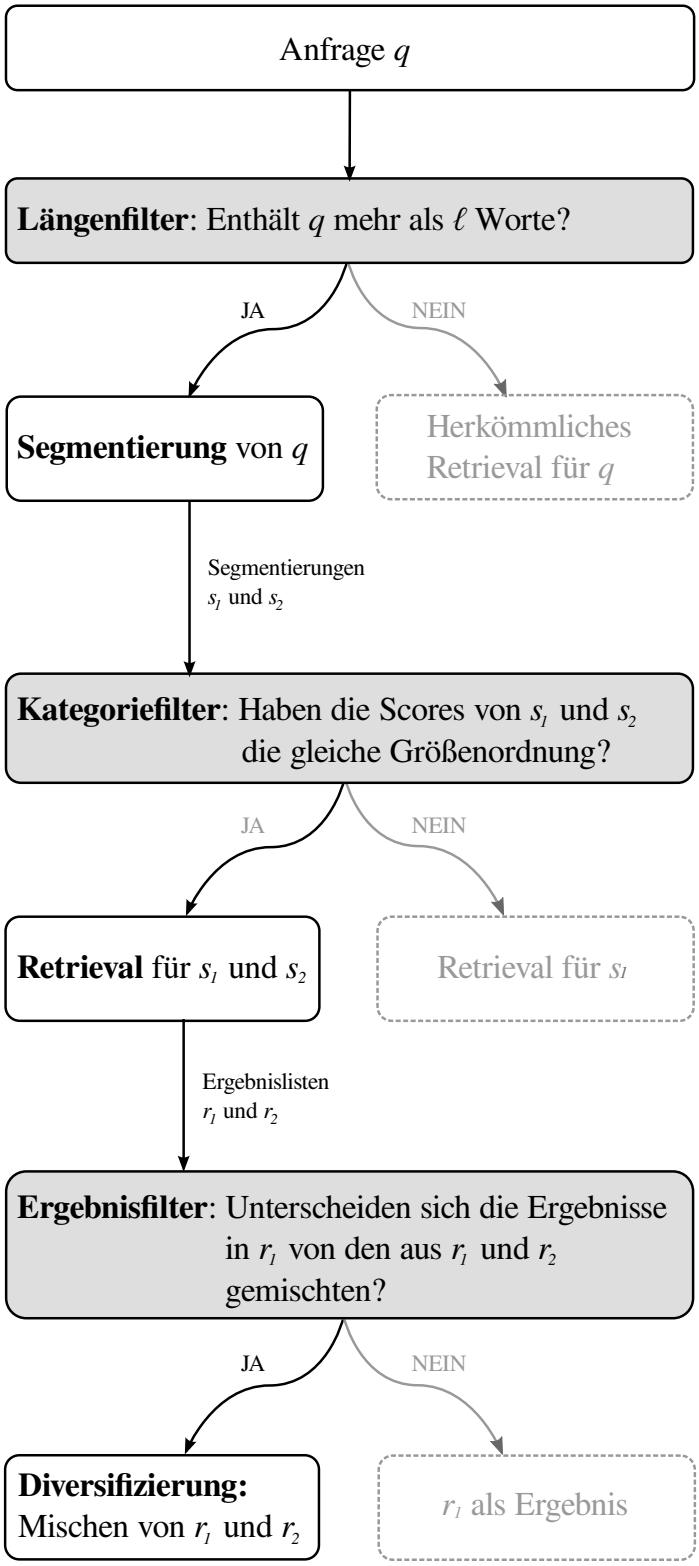


Abbildung 3: Diversifizierungspipeline

4.1 Längenfilter

Der erste Schritt im Verarbeitungsprozess der Pipeline dient dem Ausschluss von Anfragen, die „zu kurz“ sind. Eine Anfrage q wird nur an den nächsten Schritt übergeben, wenn sie mehr als ℓ Wörter beinhaltet. Andernfalls scheidet q aus dem Prozess aus und wird in herkömmlichen Retrievalverfahren weiter verarbeitet.

4.2 Segmentierungsschritt

Nachdem eine Anfrage den Längenfilter passiert hat, werden ihre Segmentierungen ermittelt. Da für eine Anfrage im Alltag einer Suchmaschine keine von Nutzern präferierten Segmentierungen entsprechend denen in Webis-QSeC-10 vorliegen, muss auf automatische Segmentierungsverfahren zurückgegriffen werden. Hierbei finden die in Abschnitt 2.2 vorgestellten Methoden Verwendung (Wikipedia-basiertes, Wikipedia-Titel- und das Wikipedia-Titel-SNP-Verfahren). Analog zur Analyse der von Menschen segmentierten Anfragen (vgl. Abschnitt 3.3.2) gilt es an dieser Stelle, die beiden „besten“ Segmentierungen jeder Anfrage zu finden.

Alle drei Verfahren liefern Listen der Segmentierungen einer Anfrage, absteigend sortiert nach den ermittelten *Scores* (Bewertungen der Segmentierungen). Die bestbewertete Segmentierung steht demnach an erster Stelle, die zweitbeste an zweiter und so weiter. Das Anwenden eines dieser Verfahren auf die Anfrage `new family sports center weimar` könnte folgendes Resultat haben:

"new family sports" "center weimar"	22 000
"new family sports" "center" "weimar"	14 000
"new" "family sports center" "weimar"	13 500
"new" "family sports" "center weimar"	11 000
"new family" "sports" "center weimar"	11 000
"new" "family" "sports" "center weimar"	8 000
"new family" "sports center" "weimar"	8 000
"new family" "sports center weimar"	5 000
"new" "family""sports center" "weimar"	5 000
"new" "family sports" "center" "weimar"	3 000
"new family" "sports" "center" "weimar"	3 000
"new" "family" "sports center weimar"	2 000
"new" "family sports center weimar"	100
"new family sports center" "weimar"	80
"new" "family" "sports" "center" "weimar"	0
"new family sports center weimar"	-1

Jeder Segmentierung wurde in dieser Auflistung eine Bewertung zugeordnet und entsprechend sortiert. Das Beispiel veranschaulicht ein Phänomen, das unter den Resultaten der Verfahren häufig eintritt: Die beiden ersten Segmentierungen sind sich sehr ähnlich, da sie im Wesentlichen aus dem Segment `"new family sports"` bestehen. Deshalb ist bei der späteren Suche nicht zu erwarten, dass deutlich verschiedene Ergebnisse erzielt werden können. Zur Auswahl der „besten“ Segmentierungen der Anfrage reicht es demnach nicht, einfach die ersten beiden Einträge der Resultatliste zu verwenden. Um dieses Problem zu umgehen, wird im Rahmen dieser Arbeit das folgende Auswahlverfahren angewandt.

Eine Anfrage q beinhaltet $|q|$ Worte und hat folglich $|q| - 1$ mögliche Stellen, an denen sie in einzelne Segmente getrennt werden kann. Diese Trennstellen werden von links nach rechts innerhalb der Anfrage nummeriert. Für die Beispielanfrage ergibt sich also

$$\texttt{new}\ |_0\ \texttt{family}\ |_1\ \texttt{sports}\ |_2\ \texttt{center}\ |_3\ \texttt{weimar}\ ,$$

wobei ein senkrechter Strich eine mögliche Trennstelle markiert. Die Menge der Trennstellen T, die q tatsächlich besitzt, ist wie folgt definiert:

$$T := \{i \mid 0 \ \leq \ i < |q| - 1\}$$

Zur Bestimmung der beiden „besten“ Segmentierungen wird die vom Verfahren bestimmte Liste von oben nach unten betrachtet. Die erste Segmentierung der Liste ist automatisch Teil des Resultatpaars und wird mit ihren Nachfolgern verglichen. Sobald sich eine dieser Segmentierungen stark genug von der ersten unterscheidet, wird sie in das Segmentierungspaar aufgenommen und die Suche ist abgeschlossen. Als ausreichend verschieden werden zwei Segmentierungen angesehen, wenn die folgende Differenzierungsvorschrift den Wert 1 ergibt:

$$\mathit{Diff}(T_1, T_2) = \begin{cases} 1 & \text{falls } (T_1 \Delta T_2) \cap T_1 \neq \varnothing \ \wedge \ (T_1 \Delta T_2) \cap T_2 \neq \varnothing \\ 0 & \text{sonst} \end{cases}$$

Hierbei sind T_1 und T_2 die Mengen der Trennstellen der beiden zu vergleichenden Segmentierungen. Die Operation Δ bezeichnet die symmetrische Differenz der beiden Mengen. Diese umfasst alle Elemente, die in T_1 und nicht in T_2 sind, vereinigt mit allen Elementen, die in T_2 und nicht in T_1 sind.

Diese Vorschrift verhindert, dass Segmentierungen als verschieden erkannt werden, wenn eine von ihnen größtenteils aus Segmenten der anderen besteht und der Unterschied zwischen beiden sich lediglich daraus ergibt, dass auf wenige Trennstellen verzichtet wurde.

Für das vorherige Beispiel ergibt sich folgender Ablauf:

1. Die Segmentierung `"new family sports" "center weimar"` hat den höchsten Wert erhalten. Sie ist demnach die erste der beiden resultierenden Segmentierungen. T_1 wird anhand ihrer Trennstellen bestimmt: $T_1 = \{2\}$.
2. T_2 wird aus den Trennstellen der zweiten Segmentierung der Liste (`"new family sports" "center" "weimar"`) bestimmt: $T_2 = \{2, 3\}$
3. *Diff*(T_1, T_2) wird berechnet. Zunächst ist die symmetrische Differenz $T_1 \Delta T_2 = \{3\}$. Das heißt, T_1 und T_2 haben die Trennstelle 3 nicht gemeinsam. $(T_1 \Delta T_2) \cap T_1 = \varnothing$, der erste Teil der Bedingung ist also nicht erfüllt und die Segmentierung `"new family sports" "center" "weimar"` wird als Kandidat für die zweite gesuchte Segmentierung verworfen.
4. T_2 wird aus den Trennstellen der dritten Segmentierung der Liste (`"new" "family sports center" "weimar"`) bestimmt: $T_2 = \{0, 3\}$
5. *Diff*(T_1, T_2) wird berechnet. Dieses Mal beinhaltet die Ergebnismenge der symmetrischen Differenz Trennstellen aus beiden Mengen. Deshalb sind die jeweiligen Schnittmengen nicht leer und die Segmentierung `"new" "family sports center" "weimar"` ist das zweite Resultat dieses Schritts.

Auf diese Weise liefert der Segmentierungsschritt zwei Segmentierungen (nachfolgend auch als s_1 und s_2 bezeichnet), die im weiteren Verlauf der Pipeline benötigt werden.

4.3 Kategoriefilter

Der nächste Schritt der Pipeline ermittelt anhand der beiden gefundenen Segmentierungen die Kategorie (vgl. Abschnitt 3.3.1) der Anfrage q. Das Ziel ist es, analog zur Uneinigkeit menschlicher Segmentierer, Unsicherheiten mittels der Bewertungen der verschiedenen Segmentierungen einer Anfrage durch das gewählte Verfahren zu finden. Dementsprechend verwirft dieser Filter alle Anfragen, bei denen eine der beiden gefundenen Segmentierungen deutlich besser ist als die andere und deren Kategorie somit nicht *uncertain* ist. Zur Kategorisierung von q finden die beiden Segmentierungen s_1 und s_2 sowie deren Scores Verwendung. Eine Anfrage soll dann als *uncertain* gelten, wenn die Scores von s_1 und s_2 die gleiche Größenordnung haben, da dann ähnlich „starke“ Segmente in beiden auftauchen.

Es sei $Score(s)$ die errechnete Bewertung einer Segmentierung s. Die Zuweisung einer Anfrage q zu einer der Kategorien *certain*, *semi* oder *uncertain* erfolgt nach der Vorschrift

$$Cat(q) = \begin{cases} certain & \text{falls } Quot(s_1, s_2) \leq 0{,}01 \\ semi & \text{falls } 0{,}01 < Quot(s_1, s_2) \leq 0{,}1 \\ uncertain & \text{sonst} \end{cases} .$$

Die Funktion $Quot(s_1, s_2)$ ist definiert als

$$Quot(s_1, s_2) = \frac{Score(s_2)}{Score(s_1)} .$$

Die Berechnung dieses Quotienten gibt Aufschluss über den Anteil der Bewertung von Segmentierung s_2 an der von s_1. Je ähnlicher sich die beiden Werte sind, desto näher liegt $Quot(s_1, s_2)$ an 1, je mehr sie voneinander abweichen, desto näher liegt $Quot(s_1, s_2)$ an 0.

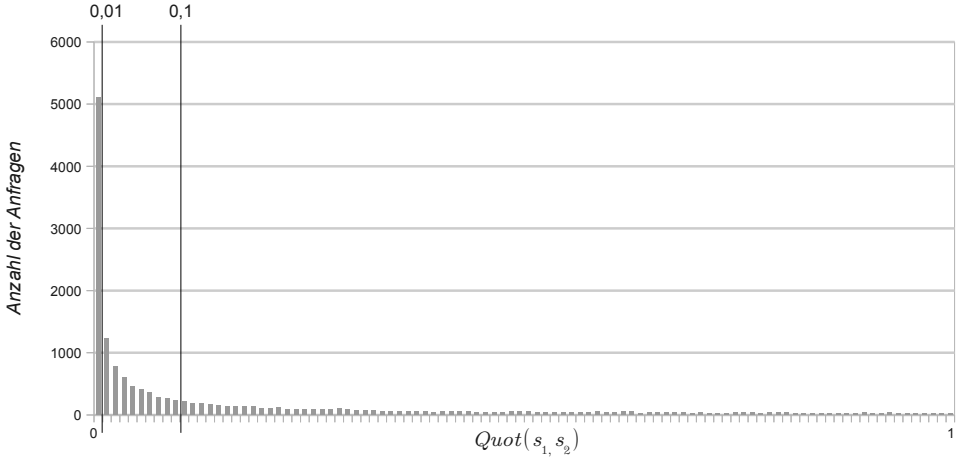
Die Funktion $Cat(q|s_1, s_2)$ markiert eine Anfrage genau dann als *uncertain*, wenn die Bewertungen ihrer beiden gewählten Segmentierungen die gleiche Größenordnung haben, sich also um weniger als Faktor 10 (dies entspricht $Quot(s_1, s_2) > 0{,}1$) unterscheiden. Ist $Score_p(s_1)$ um das Einhundertfache, bzw. zwei Größenordnungen, höher als $Score_p(s_2)$, gilt die Anfrage als *certain*. Andernfalls wird sie der Kategorie *semi* zugeordnet.

Wendet man diese Vorschrift auf die Beispielanfrage `new family sports center weimar` mit ihren in Abschnitt 4.2 ausgewählten Segmentierungen an, ergibt sich folgendes Bild:

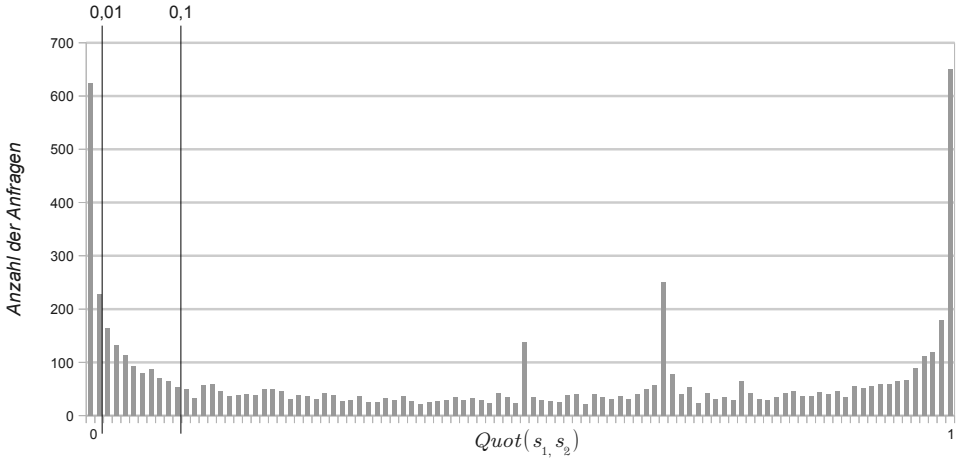
s_1 =`"new family sports" "center weimar"`
s_2 =`"new" "family sports center" "weimar"`

$Score(s_1)$	=	22 000
$Score(s_2)$	=	13 500
$Quot(s_1, s_2)$	=	0,613 636 364

Demnach unterscheiden sich die Scores der beiden Segmentierungen um weniger als eine Größenordnung und die Anfrage wird der Kategorie *uncertain* zugeteilt.



(a) Verteilung für Drei-Wort-Anfragen



(b) Verteilung für Fünf-Wort-Anfragen

Abbildung 4: Verteilung von $Quot(s_1, s_2)$ für Anfragen aus dem Webis-QSeC-10 unter Verwendung des Wikipedia-basierten Verfahrens

Dass dies kein Einzelfall ist, zeigt Abbildung 4, in der exemplarisch für Anfragen aus dem Webis-QSeC-10 die Verteilungen von $Quot(s_1, s_2)$ illustriert sind. Die Bewertungen der Segmentierungen wurden mit Hilfe des Wikipedia-basierten Verfahrens errechnet. Die Schwellwerte für die Berechnung von $Cat(q)$ sind durch senkrechte Linien in den einzelnen Diagrammen gekennzeichnet. Ungefähr 10 % der Fünf-Wort-Anfragen sind als *certain*, 16 % als *semi* und 74 % als *uncertain* kategorisiert. Im Falle der Drei-Wort-Anfragen werden ca. 34 % als *certain*, 31 % als *semi* und 35 % als *uncertain* eingestuft. Insbesondere längere Anfragen (d.h. mehr als drei Worte) haben also häufig zwei gewählte Segmentierungen, deren Scores in der gleichen Größenordnung liegen.

Nachdem der Kategorisierungsfilter eine Anfrage verarbeitet hat, sind die Prä-Retrievalschritte der Pipeline abgeschlossen. Alle anschließenden Teile des Verfahrens beruhen auf den Suchergebnissen für die beiden gewählten Segmentierungen.

4.4 Retrievalschritt

Hat eine Anfrage eine gewisse Mindestlänge überschritten, wurden ihre beiden „besten“ Segmentierungen ermittelt und sie infolgedessen als *uncertain* kategorisiert, bestimmt der Retrievalschritt der Pipeline die Suchergebnisse von s_1 und s_2. Da die weiteren Schritte der Pipeline analog zur Analyse in Abschnitt 3.3.2 ausschließlich die Ergebnislisten und nicht die eigentlichen Dokumente des Retrievalresultats betrachten, ist es für das Verfahren zur Diversifizierung ausreichend, diese Listen zu ermitteln. Die Erbebnislisten der Segmentierungen s_1 und s_2, im Folgenden auch als r_1 und r_2 bezeichnet, bilden die Grundlage für die weiteren Schritte der Pipeline.

4.5 Ergebnisfilter

Nachdem die Ergebnislisten der beiden „besten“ Segmentierungen einer Anfrage ermittelt wurden, entscheidet ein letzter Filter, ob die Ergebnisse tatsächlich diversifiziert werden sollen.

Wie bereits in Abschnitt 3.3.2 erwähnt, ist ein Suchmaschinennutzer in der Regel an den ersten fünf bis zehn Einträgen der Ergebnisliste interessiert. Befindet sich unter diesen kein befriedigendes Resultat, wird die Anfrage häufig eher neu formuliert, als dass weitere Ergebnisse im restlichen Teil der Liste untersucht werden. Die Ergebnislisten der Segmentierungen s_1 und s_2 können sich gerade in den ersten fünf bis zehn Einträgen stark ähneln, auch wenn durch die bisherige Verarbeitung in der Pipeline sichergestellt wurde, dass die einzelnen Segmente möglichst verschieden sind (vgl. Abschnitt 4.2). Deshalb liegt es in der Verantwortung des Ergebnisfilters, nur Anfragen in der Pipeline zu behalten, für die eine Diversifizierung der Ergebnisse Änderungen im Vergleich zur Ergebnisliste der vom Segmentierungsverfahren ermittelten Top-Segmentierung bewirkt. Wenn man die ersten Einträge einer aus r_1 und r_2 gemischten Liste betrachtet und mit r_1 vergleicht, so sollten sich beide in mindestens einem Eintrag unterscheiden, da andernfalls ebenso gut r_1 als Resultat verwendet werden kann.

Formal ausgedrückt: Es sei $Div(r_1, r_2, k)$ ein beliebig geartetes Verfahren, das die beiden Ergebnislisten r_1 und r_2 erhält und aus deren ersten Einträgen eine neue, diversifizierte Liste r_D der Länge k erstellt. Es sei weiterhin $r(i)$ der i-te Eintrag einer Resultatliste r. Der binäre Ergebnisfilter $Res(r_1, r_D, \kappa)$ ist wie folgt definiert:

$$Res(r_1, r_D, \kappa) = \begin{cases} 0 & \text{falls } \forall i \in [1, \kappa] : r_1(i) = r_D(i) \\ 1 & \text{sonst} \end{cases} .$$

Für κ gilt: $1 \leq \kappa \leq k$.

Der Ergebnisfilter vergleicht die ersten κ Einträge der Ergebnisliste der Top-Segmentierung (r_1) mit der aus r_1 und r_2 durch das Verfahren $Div(r_1, r_2, k)$ erstellten Liste. Sobald sich beide in einem Eintrag unterscheiden, werden sie (und somit die ursprüngliche Anfrage q) für die tatsächliche Diversifizierung freigegeben. Falls jedoch alle κ Einträge beider Listen identisch sind, wird q an dieser Stelle der Pipeline verworfen. Der Parameter κ reguliert dabei, unter wie vielen Einträgen nach Unterschieden gesucht werden soll.

Betrachtet man erneut die Beispielanfrage `new family sports center weimar`, so könnten die ersten fünf Einträge der Ergebnislisten r_1 und r_2 der beiden gewählten Segmentierungen folgendermaßen aussehen:

	r_1
1	`curling-weimar.de`
2	`weimar.info/tourist/sports`
3	`bowling-weimar.org`
4	`sports-in-weimar.de`
5	`weimar.inf/center`

	r_2
1	`sports-center-weimar.de/family`
2	`curling-weimar.de`
3	`best-sports.de/weimar`
4	`bowling-weimar.org`
5	`sports-in-weimar.de`

Nach dem Mischen beider ist die folgende Liste r_D denkbar:

	r_D
1	`sports-center-weimar.de/family`
2	`curling-weimar.de`
3	`weimar.info/tourist/sports`
4	`sports-in-weimar.de`
5	`bowling-weimar.org`

Wendet man nun $Div(r_1, r_D, 5)$ an, werden die Einträge an den ersten fünf Stellen beider Listen verglichen. Da r_D und r_1 sich in jedem Eintrag unterscheiden, liefert der Filter den Wert 1 und die Anfrage q wird für die weitere Verarbeitung freigegeben.

4.6 Diversifizierungsschritt

Der letzte Schritt der Pipeline ist das Anwenden eines Diversifizierungsverfahrens $Div(r_1, r_2|k)$ auf die ermittelten Ergebnisse r_1 und r_2. Die resultierende Liste beinhaltet dann eine Mischung der ersten Ergebnisse der beiden ursprünglichen Resultatlisten und kann als Antwort auf die Anfrage q angezeigt werden.

Im Rahmen dieser Arbeit findet ein einfaches Vorgehen nach dem Reißverschlussprinzip Anwendung. Dieses ist in Abbildung 5 in Pseudo-Code formuliert und wird im Folgenden als $Zip(r_1, r_2, k)$ bezeichnet. Der Algorithmus vereint die Einträge der Ergebnisliste r_1 mit denen von r_2 zu einer neuen Liste r_D unter Berücksichtigung einer gewünschten Länge k. Dabei gilt es zu beachten, dass Ergebnisse aus r_1 auch in r_2 enthalten sein können und in der resultierenden Liste nicht doppelt auftauchen dürfen.

```
1  BEGIN PROCEDURE Zip(r1, r2, k)
2
3     rD = []
4     pos_in_r1 = 0
5     pos_in_r2 = 0
6
7     WHILE length of rD < k DO
8
9        IF r1 at pos_in_r1 is not in rD THEN
10          add r1 at pos_in_r1 to rD
11          pos_in_r1 += 1
12       END IF
13
14       IF length of rD < k THEN
15          IF r2 at pos_in_r2 is not in rD THEN
16             add r2 at pos_in_r2 to rD
17             pos_in_r2 += 1
18          END IF
19       END IF
20
21    END DO
22
23    RETURN rD
24
25 END PROCEDURE
```

Abbildung 5: Algorithmus zur Diversifizierung zweier Ergebnislisten

Das Verfahren $Zip(r_1, r_2, k)$ wählt abwechselnd ein Ergebnis aus beiden Listen aus und prüft, ob dieses bereits in der resultierenden Liste enthalten ist. Falls nicht, wird es hinzugefügt. Anschließend werden die nächsten Einträge aus r_1 und r_2 betrachtet. Dies geschieht so lange, bis die Resultatliste r_D die gewünschte Länge k erreicht hat.

Wendet man dieses Verfahren auf die Listen r_1 und r_2 aus Abschitt 4.5 an und setzt die gewünschte Länge $k = 5$, ergibt sich der folgende Ablauf:

1. Der Eintrag `curling-weimar.de` wird aus r_1 entnommen und zur Ergebnisliste r_D hinzugefügt.
2. Der Eintrag `sports-center-weimar.de/family` wird aus r_2 entnommen. Da er noch nicht in r_D enthalten ist, wird er ebenfalls hinzugefügt.
3. Auch der Eintrag `weimar.info/tourist/sports` aus r_1 ist noch nicht in r_D enthalten und wird folglich in die Ergebnisliste aufgenommen.
4. Der Eintrag `curling-weimar.de` aus r_2 ist schon in r_D und wird deshalb ausgelassen.
5. Der Liste r_1 wird `bowling-weimar.org` entnommen. Da er noch kein Element von r_D ist, wird er der Ergebnisliste hinzugefügt.
6. Aus r_2 wird `best-sports.de/weimar` in r_D aufgenommen. Die Ergebnisliste hat nun die gewünschte Länge 5 erreicht.

Das Resultat ist die Liste r_D:

	r_D
1	`curling-weimar.de`
2	`sports-center-weimar.de/family`
3	`weimar.info/tourist/sports`
4	`bowling-weimar.org`
5	`best-sports.de/weimar`

Nach dem Anwenden aller Filter auf eine Anfrage q bzw. deren Segmentierungen s_1 und s_2 mit den zugehörigen Ergebnislisten r_1 und r_2 liefert das Verfahren $Zip(r_1, r_2, k)$ das Ergebnis der Diversifizierungspipeline.

Eine Anfrage, die die Pipeline komplett passiert, wird somit im Sinne des Verfahrens als ambig eingestuft, da ihre beiden gewählten Segmentierungen unterschiedliche Ergebnisse liefern und eine Diversifizierung beider Listen Neuerungen in die finale Liste einführt. Ob sich dadurch tatsächlich Verbesserungen erzielen lassen, soll das nächste Kapitel klären.

Kapitel 5

Evaluierung des Verfahrens

Die in Kapitel 4 beschriebene Pipeline vereint verschiedene Filterprozesse, um bestimmte Anfragen auszuwählen und ihre Ergebnisse zu diversifizieren. Es stellt sich die Frage, ob mit diesem Verfahren tatsächlich bessere Suchergebnisse erzielt werden können. In den folgenden Abschnitten wird untersucht, ob die diversifizierten Ergebnisse eine höhere Bewertung nach Anwendung einer der in Abschnitt 2.3 vorgestellten Metriken erhalten haben als die Resultate der durch das verwendete Segmentierungsverfahren gewählten besten Segmentierung.

5.1 Evaluierungskorpus

Um mit Hilfe einer Metrik bestimmen zu können, wie gut eine Ergebnisliste zu einer Anfrage passt, ist es essentiell zu wissen, welche Dokumente relevant bezüglich dieser Anfrage sind. Derartige Informationen liegen für den Webis-QSeC-10 nicht vor. Die Anfragen des *Diversity Tasks* im *TREC Web Track* (vgl. Abschnitt 2.1.3) bieten theoretisch eine Grundlage zur Evaluierung, da für sie Relevanzbewertungen vorliegen. Allerdings gab es in den Jahren 2009 bis 2011 insgesamt nur 12 Anfragen, die als ambig ausgeschrieben wurden und drei oder mehr Wörter beinhalten. Da dies viel zu wenig ist, um fundierte Aussagen treffen zu können, muss eine alternative Sammlung von Anfragen verwendet werden.

Die Autoren Roy et al. stellen in ihrer Arbeit [RGCL12] eine 500 Anfragen umfassende Sammlung (im Folgenden auch als Roy12-Korpus bezeichnet) vor. Diese wurden im Mai 2010 nach dem Zufallsprinzip aus den Query Logs der australischen Version der Suchmaschine Bing extrahiert. Alle so gewonnenen Anfragen beinhalten zwischen fünf und acht Wörter. Analog zum Vorgehen aus Abschnitt 3.2 ermittelten die Autoren für alle Segmentierungen dieser Anfragen die ersten zehn Ergebnisse der Suche mit Bing. Insgesamt wurden mit dieser Vorgehensweise 14 171 Dokumente gefunden, wobei auf eine Anfrage im Schnitt 28 Ergebnisse fallen.

Für jedes Dokument beurteilten drei unabhängige Personen, wie relevant dieses im Bezug auf die Anfrage ist, für die es als Ergebnis geliefert wurde. Für diese Beurteilung fand ein ternäres System Verwendung: Ein Dokument erhält den Relevanzwert 0, wenn es im Bezug auf die Anfrage *irrelevant*, den Wert 1, wenn es *teilweise relevant* und den Wert 2, wenn es *hochrelevant* ist. Mit Hilfe dieser Daten ist es möglich, Bewertungsmetriken auf die Ergebnisse einer Anfrage bzw. diversifizierte Ergebnislisten anzuwenden. Allerdings muss berücksichtigt werden, dass der Roy12-Korpus nicht für die Evaluierung von Diversifizierungsverfahren entwickelt wurde und dementsprechend keine Relevanzbewertungen der Dokumente hinsichtlich der einzelnen Intentionen einer Anfrage vorliegen. Da aber keine alternativen Korpora vergleichbaren Umfangs existieren, dient der Roy12-Korpus als Ersatz und die Ergebnisse der Evaluierungsmetriken müssen mit Vorsicht betrachtet werden.

5.2 Konfiguration der Pipeline

Der Roy12-Korpus bildet die Grundlage der Evaluierung des in Kapitel 4 vorgestellten Verfahrens zur Auswahl bestimmter Anfragen und Diversifizierung ihrer Ergebnisse. Dieser Abschnitt beschreibt die gewählte Konfiguration der verschiedenen Filterschritte der Pipeline.

Konfiguration des Längenfilters

Alle Anfragen im verwendeten Korpus beinhalten mindestens fünf Worte und können als ausreichend lang betrachtet werden. Eine weitere Einschränkung der Länge ist nicht nötig. Der Schwellwert ℓ (Länge, die eine Anfrage überschreiten muss, vgl. Abschnitt 4.1) des Längenfilters ist somit implizit auf 4 gesetzt. Folglich werden bei der Evaluierung alle Anfragen des Roy12-Korpus diesen Filter passieren.

Konfiguration des Segmentierungsschrittes

Bei der Bestimmung der beiden besten Segmentierungen einer Anfrage finden die Verfahren aus Abschnitt 2.2 Anwendung. Alle drei bewerten die verschiedenen Segmentierungen einer Anfrage auf Grundlage von Wikipedia-Titeln. Wenn eine Segmentierung den Wert -1 erhält, ist sie im Sinne des Verfahrens ungültig. Somit führt die Verwendung der Methoden automatisch einen weiteren Filter ein, da natürlich nur Segmentierungen verwendet werden, die im Sinne des Verfahrens gültig sind.

Die Wikipedia-basierte Vorgehensweise erweist sich als „liberalste“ Methode, da die Bewertung einer Segmentierung nur dann -1 wird, wenn keines ihrer Segmente in den Google-n-Grammen auftaucht. Das Wikipedia-Titel-Verfahren hingegen segmentiert eine Anfrage nur, wenn sie einen oder mehrere Wikipedia-Titel enthält. Zusätzlich werden nur Wikipedia-Titel als Segmente erlaubt. Deshalb wird die Anzahl der Anfragen, die die Pipeline passieren, beim Verwenden dieser Methode sehr stark eingeschränkt. Als „Auflockerung“ lässt das Wikipedia-Titel-SNP-Verfahren auch Strict-Noun-Phrases (vgl. Abschnitt 2.2) als Segmente zu und ist somit ein Mittelweg zwischen Wikipedia-basiertem und Wikipedia-Titel-Verfahren.

Konfiguration des Kategoriefilters

Der Kategoriefilter wird so verwendet, wie er in Kapitel 4.3 beschrieben wurde: Anfragen fallen genau dann in die Kategorie *uncertain*, wenn die Scores der beiden gewählten Segmentierungen die gleiche Größenordnung besitzen. Da für die Evaluierung nur Anfragen relevant sind, die die gesamte Pipeline passieren, werden solche, die in den Kategorien *semi* oder *certain* landen, im weiteren Verlauf nicht betrachtet.

Konfiguration des Retrievalschrittes

Das Retrieval erfolgt auf der vorliegenden Dokumentensammlung (ca. 14 000 Dokumente) des Roy12-Korpus. Die Autoren Roy et al. verwendeten für den Aufbau eines Suchindexes und die anschließende Suche die Software-Bibliothek Lucene[1] in ihrer Standardkonfiguration [RGCL12]. Zur besseren Vergleichbarkeit der Resultate wird für die Indexierung und Suche im Rahmen dieser Arbeit ebenfalls Lucene verwendet. Die Ergebnisse der beiden gewählten Segmentierungen einer Anfrage werden auf diese Weise basierend auf den vorliegenden Dokumenten bestimmt.

[1] `http://lucene.apache.org/core/` (letzter Zugriff am 07.02.2013)

Konfiguration des Diversifizierungsschritts und des Ergebnisfilters

Die im Retrievalschritt ermittelten Ergebnisse r_1 und r_2 der beiden Segmentierungen s_1 und s_2 werden nach dem Passieren des Ergebnisfilters gemischt. Für die Diversifizierung wird das in Abschnitt 4.6 beschriebene Verfahren $Zip(r_1, r_2, k)$ verwendet, um gemischte Ergebnislisten r_D der Länge fünf bzw. zehn zu erzeugen. Der Ergebnisfilter prüft vorher mit Hilfe der Funktion $Res(r_1, r_D, \kappa)$, ob sich r_D von r_1 unterscheidet. Für diesen Vergleich werden im Verlauf der Evaluierung immer die ersten fünf Ergebnisse betrachtet ($\kappa = 5$).

Alle 500 Anfragen des Roy12-Korpus werden in der auf diese Weise konfigurierten Diversifizierungspipeline verarbeitet. Unter Verwendung der Relevanzbewertungen kann anschließend mit Hilfe verschiedener Bewertungsmetriken berechnet werden, wie gut eine Ergebnisliste den Informationsbedarf einer bestimmte Anfrage abdeckt. Insbesondere wird sich zeigen, ob die diversifizierten Ergebnisse besser sind als die der Top-Segmentierung eines Verfahrens.

5.3 Verwendete Metrik

Für die Bewertung von Ergebnislisten basierend auf den Relevanzbewertungen der einzelnen Dokumente wurden in Abschnitt 2.3 verschiedene Metriken vorgestellt. Es handelt sich dabei um verbreitete Methoden zur Evaluierung von Suchergebnissen in Bezug auf eine bestimmte Anfrage. In Abschnitt 2.3 wurde darauf eingegangen, wie die Metriken *Mean Average Precision*, *Mean Reciprocal Rank* und *Normalized Discounted Cumulative Gain* im Hinblick auf verschiedene Intentionen einer ambigen Anfrage modifiziert werden können, um Resultatlisten entsprechend präziser zu bewerten. Für diese angepassten Varianten ist es notwendig, die Beurteilung der Relevanz eines Dokuments nicht nur in Bezug auf eine einzelne Anfrage, sondern vielmehr für jede Intention dieser Anfrage zu kennen. Da weder der Roy12-Korpus noch eine andere Dokumentensammlung hinreichender Größe diese Informationen liefert, muss auf die herkömmlichen Varianten (d.h. ohne Berücksichtigung der Intentionen einer Anfrage) der Metriken zurückgegriffen werden.

Unter den drei vorgestellten ist der *Normalized Discounted Cumulative Gain (nDCG)* die am häufigsten verwendete Methode. Sie bezieht sowohl die Relevanzen aller Dokumente innerhalb eines bestimmten Cutoffs (vgl. Abschnitt 2.3) als auch deren Position in der Liste mit ein. Da sie als Standardmethode im Bereich der Retrievalmetriken gilt, wird sie in den folgenden Betrachtungen verwendet.

$\|q\|$	Wikipedia-basiert	Wikipedia-Titel	Wikipedia-Titel-SNP
5	173 (34,6 %)	12 (2,4 %)	85 (17,0 %)
6	51 (10,2 %)	2 (0,4 %)	28 (5,6 %)
7	9 (1,8 %)	1 (0,2 %)	3 (0,6 %)
8	5 (1,0 %)	1 (0,2 %)	5 (1,0 %)
Gesamt	238 (47,6 %)	16 (3,2 %)	121 (24,2 %)

Tabelle 2: Anzahl der Anfragen aus dem Roy12-Korpus, die die Pipeline passiert haben, eingeteilt nach Anfragelänge $|q|$ und Segmentierungsverfahren

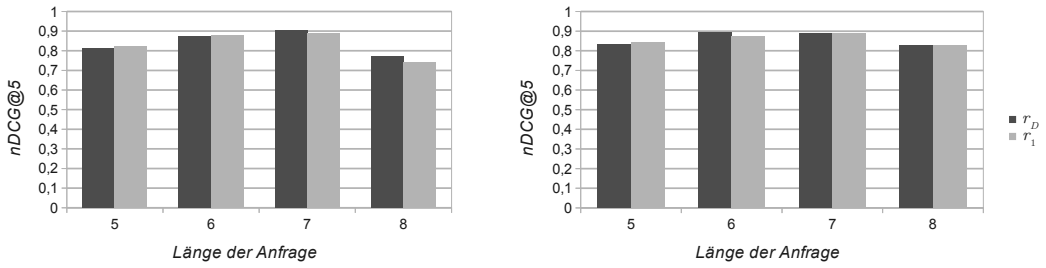
5.4 Evaluierung

Zur Evaluierung des Verfahrens werden die einzelnen Filter der Pipeline auf jede der 500 Anfragen des Roy12-Korpus angewandt. Die Tabelle 2 zeigt, wie viele Anfragen in Abhängigkeit vom gewählten Segmentierungsverfahren das Ende der Pipeline erreichen. Weiterhin ist erkenntlich, wie sich diese Anzahl auf die verschiedenen Längen der Anfragen verteilt. Anhand der Zahlen wird die bereits erwähnte Filterwirkung der Segmentierungsmethoden deutlich. Während unter Verwendung des Wikipedia-basierten Verfahrens fast die Hälfte der 500 Anfragen auf Basis der Segmentierungsbewertungen als *uncertain* kategorisiert wird, sind es bei den Verfahren Wikipedia-Titel und Wikipedia-Titel-SNP deutlich weniger. Die Verteilung über die Anfragelängen spiegelt im Mittel in etwa den von Roy et al. in [RGCL12] ermittelten Wert von 5,29 wider.

Anhand der Tabelle wird klar, dass es wenig Sinn ergibt, Vergleiche und Schlüsse auf Grundlage des Wikipedia-Titel-Verfahrens als Segmentierungsschritt zu ziehen, da zu wenige Anfragen die Pipeline passieren. Deshalb werden im weiteren Verlauf lediglich die Ergebnisse für das Wikipedia-basierte und das Wikipedia-Titel-SNP-Verfahren betrachtet.

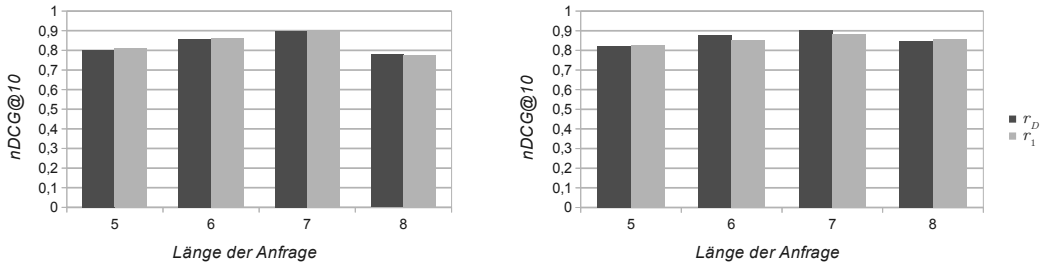
Für die Berechnung der *nDCG*-Werte werden die ternäre Relevanzbewertungen des Roy12-Korpus in binäre Werte überführt. Dies hat seine Ursache darin, dass anhand der Ausführungen der Arbeit [RGCL12] nicht erkenntlich wird, wodurch sich *teilweise relevante* Dokumente auszeichnen bzw. von *hochrelevanten* und *irrelevanten* unterscheiden. Aus diesem Grund gilt im Folgenden ein Dokument genau dann als relevant und erhält den Relevanzwert 1, wenn die Mehrheit der Beurteilenden (also mindestens zwei von drei) es als *teilweise relevant* oder *hochrelevant* eingestuft hat. Andernfalls ist es irrelevant und erhält den Wert 0.

Ausgehend von diesen Bewertungen lassen sich die *nDCG*-Werte der Ergebnisliste r_1 der besten Segmentierung einer Anfrage und der diversifizierten Ergebnisliste r_D berechnen und vergleichen.



(a) Wikipedia-basiertes Verfahren (b) Wikipedia-Titel-SNP-Verfahren

Abbildung 6: Mittlere *nDCG*-Werte der ersten fünf Ergebnisse der Listen r_D und r_1 unter Verwendung verschiedener Segmentierungsverfahren

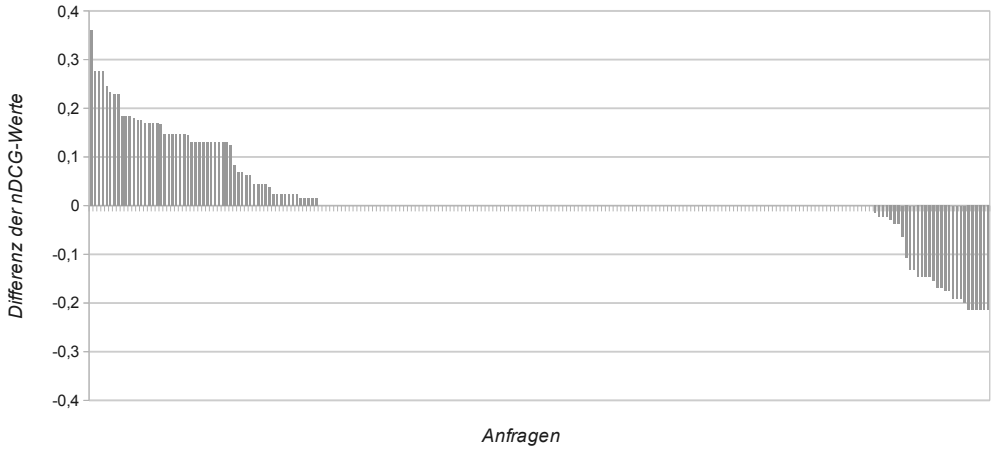


(a) Wikipedia-basiertes Verfahren (b) Wikipedia-Titel-SNP-Verfahren

Abbildung 7: Mittlere *nDCG*-Werte der ersten zehn Ergebnisse der Listen r_D und r_1 unter Verwendung verschiedener Segmentierungsverfahren

Die Abbildung 6 illustriert die errechneten Werte beim Betrachten der ersten fünf Ergebnisse einer Liste, aufgeschlüsselt nach der Länge der Anfrage und getrennt nach dem jeweils gewählten Segmentierungsverfahren. Die Werte sind für die beiden Listen jeder Anfrage berechnet und über alle Anfragen gleicher Länge gemittelt. Es wird deutlich, dass der *nDCG*-Wert einer Liste den der anderen im Durchschnitt nie stark überragt. Die maximalen Differenzen betragen ca. 0,03 für das Wikipedia-basierte und 0,02 für das Wikipedia-Titel-SNP-Verfahren, also 3,4 % bzw. 2,4 % gemessen am jeweils erreichten maximalen Wert.

Allerdings sind die *nDCG*-Werte der gemischten Ergebnisse der Fünf-Wort-Anfragen, die den Großteil der gefilterten Anfragen ausmachen, jeweils leicht niedriger (ca. 0,009 für Wikipedia-basiert, 0,007 für Wikipedia-Titel-SNP) als die nicht gemischten der besten Segmentierung.



(a) Wikipedia-basiertes Verfahren

(b) Wikipedia-Titel-SNP-Verfahren

Abbildung 8: Differenzen der $nDCG$-Werte der ersten fünf Ergebnisse der Listen r_D und r_1 pro Anfrage unter Verwendung verschiedener Segmentierungsverfahren

Ein ähnlicher Eindruck ergibt sich beim Betrachten der Abbildung 7, die im Grunde das Gleiche darstellt wie Abbildung 6, wobei jedoch die ersten zehn statt fünf Ergebnisse für die Berechnung der $nDCG$-Werte berücksichtigt werden. Die maximalen Differenzen sind auch hier ähnlich klein (ca. 0,008 für Wikipedia-basiert, 0,027 für Wikipedia-Titel-SNP). Ebenso sind die gemischten Ergebnisse der Fünf-Wort-Anfragen etwas schlechter als die der besten Segmentierung (ca. 0,008 für Wikipedia-basiert, 0,004 für Wikipedia-Titel-SNP).

Diese Analyse zeigt deutlich, dass eine Diversifizierung entsprechend der Pipeline durchschnittlich keine nennenswerte Verbesserung der Ergebnisse zur Folge hat. Da jedoch das arithmetische Mittel, welches zum Berechnen der Werte pro Anfragelänge verwendet wird, für kleine Stichproben wie diese ein relativ anfälliges Maß in Bezug auf unerwartete Werte (sogenannte Ausreißer) ist, kann ein Blick auf die *nDCG*-Werte der einzelnen Anfragen lohnenswert sein.

Die Abbildung 8 veranschaulicht die Differenz des *nDCG*-Wertes der Ergebnisliste r_D mit dem von r_1 für jede Anfrage, die die Pipeline passiert hat. Abermals werden die ersten fünf Ergebnisse für die Berechnung des *nDCG*-Wertes betrachtet. Die Anfragen sind auf der x-Achse angetragen und absteigend nach dem errechneten Differenzwert sortiert. Ist der letztgenannte positiv, so ist der *nDCG*-Wert von r_D größer als der von r_1 und die gemischte Ergebnisliste folglich besser. Für negative Werte ist das Gegenteil der Fall.

Es ist aus der Darstellung klar erkennbar, dass sich neben einigen Anfragen, für die das Mischen eine klare Verbesserung (ca. 24,18 % Wikipedia-basiert, 18,18 % Wikipedia-Titel-SNP) oder Verschlechterung (ca. 17,21 % Wikipedia-basiert, 14,05 % Wikipedia-Titel-SNP) bewirkt, der *nDCG*-Wert von r_D dem von r_1 für den Großteil der Anfragen (ca. 58,61 % Wikipedia-basiert, 67,77 % Wikipedia-Titel-SNP) gleicht. In diesen Fällen spielt die Ergebnisdiversifizierung hinsichtlich der Relevanz der in den Listen enthaltenen Ergebnisse keine Rolle. Außerdem zeigt sich, dass es für beide Segmentierungsverfahren mehr Anfragen gibt, bei denen die Diversifizierung die Ergebnisse verbessert, als solche, bei denen sie die Ergebnisse verschlechtert. Allerdings wirkt sich die Verschlechterung des *nDCG*-Wertes im Mittel (ca. 0,18 Wikipedia-basiert, 0,16 Wikipedia-Titel-SNP) etwas stärker aus, als die Verbesserung (ca. 0,12 Wikipedia-basiert, 0,13 Wikipedia-Titel-SNP). Die Ursache dafür können unter anderem Ausreißer sein.

5.5 Bemerkungen

Es lässt sich festhalten, dass für den größten Teil der vorliegenden Anfragen eine Diversifizierung der Ergebnisse entweder Verbesserungen bewirkt oder zumindest keinen Nachteil hinsichtlich der Relevanz einer Ergebnisliste mit sich bringt. Offenbar gelangen aber auch einige Anfragen durch die Filter der Pipeline, bei denen die Diversifizierung mit dem beschriebenen Verfahren deutlich schlechtere Ergebnisse hervorbringt als die beste Segmentierung einer Anfrage. Demnach ist die verwendete Vorgehensweise offenbar noch nicht optimal, da im Schnitt keine Verbesserung der Ergebnisse erzielt wird. Wie in Abschnitt 5.1 erwähnt, kann dies aber auch am verwendeten Korpus liegen, da dieser im Grunde nicht für die Evaluierung diversifizierender Verfahren erstellt wurde. Dementsprechend würden Experimente auf einem besser geeigneten Korpus, d.h. mit Relevanzbewertungen für jede Intention einer Anfrage, möglicherweise andere Evaluierungsergebnisse erzielen.

Dieses Kapitel hat für eine Sammlung von Anfragen und zugehörigen Dokumenten (Roy12-Korpus) gezeigt, inwiefern das vorgeschlagene Pipeline-Konzept die Suchergebnisse bestimmter Anfragen verbessern kann. Im letzten Kapitel werden die Inhalte und Ergebnisse dieser Arbeit zusammengefasst und kritisch bewertet.

Kapitel 6

Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde ein neues Verfahren in Form einer Pipeline vorgestellt, welches anhand der verschiedenen Segmentierungen einer langen Anfrage Rückschlüsse auf den Grad ihrer Ambiguität zieht und auf dieser Grundlage gegebenenfalls diversifizierte Ergebnisse ermittelt. Motiviert ist dieser Ansatz durch Unsicherheiten, die bei der Auswahl von Segmentierungen durch Menschen festgestellt wurden. Verschiedene Filterschritte innerhalb der Pipeline sollen diese Unsicherheit basierend auf automatischen Verfahren modellieren. So werden Anfragen gefunden, deren verschiedene Intentionen durch Segmentierungen ausgedrückt werden können.

Die Evaluierung hat gezeigt, dass mit Hilfe der Pipeline gefundene diversifizierte Anfrageergebnisse im Schnitt etwas schlechter sind, als die der besten Segmentierung der Anfrage im Kontext des verwendeten Segmentierungsverfahrens. Allerdings bieten sowohl die einzelnen Filterschritte als auch der verwendete Korpus Raum zur Verbesserung.

Der *Längenfilter* kann beispielsweise aufgewertet werden, indem eine Obergrenze für die Anfragelänge empirisch ermittelt und hinzugefügt wird. Für die Evaluierung war diese Obergrenze implizit durch die im Roy12-Korpus gegebene Maximallänge von acht gegeben. Die Autoren Roy et al. begründeten dies damit, dass eine Anfrage mit mehr als acht Worten in der Regel entweder Fehlermeldungen, ein kompletter Satz oder ganze Liedtexte seien und deshalb im Rahmen ihrer Arbeit entfallen [RGCL12]. Möglicherweise lässt sich dieser Wert auch unter Verwendung der Segmentierungsverfahren von Hagen et al. bestätigen [HPSB11]. Als Untegrenze wurde eine Länge von fünf Worten gewählt, da auf diese Weise eine ausreichende Menge potentieller Segmentierungen vorhanden ist. Eine genauere Analyse könnte eine andere Länge für die Untergrenze ergeben.

Die Auswahl der beiden besten Segmentierungen einer Anfrage im *Segmentierungsschritt* bietet ebenfalls Möglichkeiten zur Verbesserung. Aktuell werden die von Segmentierungsverfahren am besten bewertete und die erste sich deutlich von dieser unterscheidende Segmentierung gewählt. Mit Sicherheit werden dadurch nicht immer die beiden Segmentierungen ausgesucht, die später auch die unterschiedlichsten Ergebnisse liefern. Demnach könnte auf Basis der gewählten Verfahren ein besserer Weg zur Auswahl gefunden werden.

Ähnliches gilt für den *Kategoriefilter*. Möglicherweise ist der Unterschied der Größenordnungen der Segmentierungsbewertungen kein ausreichend starker Indikator für die Kategorisierung der Anfragen. Vielleicht besteht sogar noch Verbesserungspotential bei den Bewertungsmethoden der gewählten Segmentierungsverfahren selbst, um in manchen Fällen klarere Unterschiede feststellen zu können.

Das gewählte Verfahren im *Diversifizierungsschritt* ist zwar durchaus zweckdienlich, aber auch verhältnismäßig simpel gehalten. Es ist denkbar, dass ein solches Reißverschlussverfahren für den Nutzer interessante Dokumente häufig zu weit in den unteren Rängen der Ergebnisliste platziert. Mit besseren Diversifizierungsmethoden, die Dokumente nach anderen Maßstäben in die Resultatliste einfügen, ist auch an dieser Stelle eine Verbesserung denkbar.

Nicht zuletzt muss nochmals erwähnt werden, dass der für die Evaluierung verwendete Korpus nicht für die Analyse von Diversifizierungsverfahren erstellt wurde und daher weder annotierte Intentionen einer Anfrage noch entsprechende Relevanzbewertungen vorliegen. Dementsprechend konnten die in Abschnitt 2.3 Bewertungsmetriken nur in ihrer herkömmlichen Form, d.h. ohne Berücksichtigung der Anfrageabsichten, verwendet werden. Somit beschreiben die in Abschnitt 5.4 ermittelten Werte lediglich, wie gut die diversifizierten Ergebnisse den Informationsbedarf der Anfragen als solche befriedigen und geben keinen Aufschluss darüber, ob und zu welchem Grad einzelne Intentionen abgedeckt werden. Für eine bessere Evaluierung wird deshalb eine Anfragesammlung benötigt, die einen ähnlichen Umfang wie der Roy12-Korpus hat und Relevanzbewertungen bezüglich aller Intentionen der enthaltenen Anfragen liefert.

Eventuell ist es möglich, unter Berücksichtigung der genannten Verbesserungsmöglichkeiten bessere Resultate zu erzielen. Dennoch zeigt diese Arbeit, dass es prinzipiell funktioniert, anhand der verschiedenen Segmentierungen einer Anfrage Rückschlüsse auf deren Ambiguität zu ziehen und die Suchergebnisse in einigen Fällen zu verbessern. Zudem ist die vorgestellte Pipeline ein leicht erweiterbares Grundgerüst mit verschiedenen Konfigurationsmöglichkeiten. Durch das Einfügen neuer oder Austauschen bestehender Filterschritte lässt sich das Verfahren beliebig modifizieren. Wenn es in Zukunft gelingt, noch mehr Anfragen auszuschließen, für die eine Diversifizierung eine Verschlechterung der Ergebnisse bedeutet, so kann dieses Verfahren die Suche deutlich erleichtern.

Literaturverzeichnis

[AGHI09] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In Ricardo A. Baeza-Yates, Paolo Boldi, Berthier A. Ribeiro-Neto, and Berkant Barla Cambazoglu, editors, *WSDM*, pages 5–14. ACM, 2009.

[CCS09] Charles L. A. Clarke, Nick Craswell, and Ian Soboroff. Overview of the trec 2009 web track. In Voorhees and Buckland [VB09].

[CKC+08] Charles L. A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In Sung-Hyon Myaeng, Douglas W. Oard, Fabrizio Sebastiani, Tat-Seng Chua, and Mun-Kew Leong, editors, *SIGIR*, pages 659–666. ACM, 2008.

[CSA+09] Paul Clough, Mark Sanderson, Murad Abouammoh, Sergio Navarro, and Monica Lestari Paramita. Multiple approaches to analysing query diversity. In James Allan, Javed A. Aslam, Mark Sanderson, ChengXiang Zhai, and Justin Zobel, editors, *SIGIR*, pages 734–735. ACM, 2009.

[CTC02] Steve Cronen-Townsend and W. Bruce Croft. Quantifying query ambiguity. In *Proceedings of the second international conference on Human Language Technology Research*, HLT '02, pages 104–109, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[HPBS12] Matthias Hagen, Martin Potthast, Anna Beyer, and Benno Stein. Towards Optimum Query Segmentation: In Doubt Without. In *21st ACM International Conference on Information and Knowledge Management (CIKM 12)*. ACM, October 2012.

[HPSB11] Matthias Hagen, Martin Potthast, Benno Stein, and Christof Bräutigam. Query segmentation revisited. In Sadagopan Srinivasan, Krithi Ramamritham, Arun Kumar, M. P. Ravindra, Elisa Bertino, and Ravi Kumar, editors, *WWW*, pages 97–106. ACM, 2011.

[RGCL12] Rishiraj Saha Roy, Niloy Ganguly, Monojit Choudhury, and Srivatsan Laxman. An ir-based evaluation framework for web search query segmentation. In William R. Hersh, Jamie Callan, Yoelle Maarek, and Mark Sanderson, editors, *SIGIR*, pages 881–890. ACM, 2012.

[SMO10] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Selectively diversifying web search results. In Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An, editors, *CIKM*, pages 1179–1188. ACM, 2010.

[SMO11] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Intent-aware search result diversification. In Wei-Ying Ma, Jian-Yun Nie, Ricardo A. Baeza-Yates, Tat-Seng Chua, and W. Bruce Croft, editors, *SIGIR*, pages 595–604. ACM, 2011.

[SPMO10] Rodrygo L. T. Santos, Jie Peng, Craig Macdonald, and Iadh Ounis. Explicit search result diversification through sub-queries. In Cathal Gurrin, Yulan He, Gabriella Kazai, Udo Kruschwitz, Suzanne Little, Thomas Roelleke, Stefan M. Rüger, and Keith van Rijsbergen, editors, *ECIR*, volume 5993 of *Lecture Notes in Computer Science*, pages 87–99. Springer, 2010.

[VB09] Ellen M. Voorhees and Lori P. Buckland, editors. *Proceedings of The Eighteenth Text REtrieval Conference, TREC 2009, Gaithersburg, Maryland, USA, November 17-20, 2009*, volume Special Publication 500-278. National Institute of Standards and Technology (NIST), 2009.