Bauhaus-Universität Weimar
Fakultät Medien
Studiengang Mediensysteme

Neue Verfahren und Evaluierungsmaße für Anfragesegmentierungen

Bachelorarbeit

Anna Beyer

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

Datum der Abgabe: 15. Dezember 2011

Erklärung

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

Weimar, den 15. Dezember 2011

.....
Anna Beyer

Zusammenfassung

Die vorliegende Arbeit befasst sich mit der Segmentierung von Suchanfragen. Dabei werden die Worte einer gegebenen Anfrage in Sinneinheiten eingeteilt, um den Informationsbedarf des Nutzers zu konkretisieren und somit die Retrieval Performance zu verbessern. Nur die wenigsten Nutzer segmentieren ihre Suchanfragen, da sie die Option der Anfragesegmentierung vermutlich nicht kennen. Abhilfe schaffen Verfahren, welche die Nutzeranfragen vor der eigentlichen Suche automatisch segmentieren.

Zur Bewertung solcher Verfahren werden Evaluierungskorpora herangezogen. Diese Korpora enthalten Suchanfragen, die nachträglich von menschlichen Annotatoren segmentiert wurden. Bisher wurde der Bergsma-Wang-Corpus [BW07] zur Evaluierung eingesetzt. Er enthält jedoch nur 500 Nominalanfragen mit je 3 Segmentierungen und gilt deshalb als nicht repräsentativ. Um eine angemessene Grundlage für die Bewertung von Anfragesegmentierungen zu schaffen, führten Hagen et al. 2010 den Webis Query Segmentation Corpus ein [HPSB11]. Dieser neue Korpus umfasst repräsentative 53 437 Anfragen, die von jeweils 10 Annotatoren segmentiert wurden.

In der vorliegenden Arbeit wird der Webis Query Segmentation Corpus erstmals genauer untersucht. Neben der Konstruktion des Korpus steht dabei vor allem die Analyse des menschlichen Verhaltens beim Segmentieren von Anfragen im Vordergrund.

Die Anwendung der bisherigen Evaluierungsmaße auf den Webis Query Segmentation Corpus hätte aufgrund der vermehrten Segmentierungen pro Anfrage eine unausgewogene Bewertung der Verfahren zur Folge. Daher werden in dieser Arbeit neue Evaluierungsmaße speziell für Korpora mit einer größeren Anzahl an Segmentierungen konzipiert.

Der wichtigste Beitrag dieser Arbeit ist die Entwicklung zwei neuer Verfahren – der Wikipedia-Titel-Baseline und des hybriden Verfahrens – zur Segmentierung von Anfragen. Das hybride Verfahren kombiniert den Ansatz von Hagen et al. [HPSB11] mit der Wikipedia-Titel-Baseline und erreicht bei vergleichbarer Laufzeit eine größere Genauigkeit als die Teilverfahren bei der Segmentierung von Anfragen.

Inhaltsverzeichnis

1 Einleitung **1**

2 Anfragesegmentierung **4**
2.1 Grundlagen 4
2.2 Bestehende Segmentierungsverfahren 5
2.2.1 Naive Normalisierung 9
2.2.2 Wikipedia-basiertes Verfahren 10
2.3 Aktuelle Evaluierungskorpora 12

3 Analyse des Webis Query Segmentation Corpus **13**
3.1 Konstruktion des Korpus 13
3.2 Anfragen und Segmentierungen 15
3.3 Kritik 22

4 Evaluierung von Anfragesegmentierungen **23**
4.1 Bewertung der Genauigkeit 23
4.1.1 Genauigkeitsebenen 23
4.1.2 Evaluierungsmaße 26
4.1.3 Anwendung der Maße 31
4.2 Evaluierung der Laufzeit 34

5 Entwicklung neuer Segmentierungsverfahren **36**
5.1 Die Wikipedia-Titel-Baseline 36
5.2 Das hybride Verfahren 41

6 Zusammenfassung und Ausblick **46**

Abbildungsverzeichnis **48**

Tabellenverzeichnis **49**

Literaturverzeichnis **50**

Kapitel 1

Einleitung

Die Größe des Internets folgt einem exponentiellen Wachstum und verdoppelt sich etwa alle 5 Jahre [ZZY+08]. Die Menge der im Netz verfügbaren Informationen steigt und ebenso die Anzahl der Nutzer. Laut der *ARD/ZDF-Onlinestudie 2011* nutzen 73,3 % der Deutschen das Internet – vor 10 Jahren waren es noch etwa die Hälfte.[1] Das Auffinden von Informationen erfolgt heute vorwiegend durch die Verwendung von Suchmaschinen im Internet [AGHI09]. Dabei formuliert der Nutzer seinen Informationsbedarf in einigen Stichworten und übermittelt diese an die Suchmaschine. Allein Google bearbeitet mehr als eine Milliarde Suchanfragen täglich.[2] Die Anforderungen an die Suche im Internet werden in folgendem Zitat charakterisiert:

> "The perfect search engine, would understand exactly what you mean and give back exactly what you want."
> (Larry Page, Mitbegründer von Google).[3]

Doch die perfekte Suchmaschine, wie sie Larry Page beschreibt, gibt es noch nicht. Nur etwa 90–95 % der Anfragen liefern Ergebnisse, die den Informationsbedarf der Nutzer befriedigen können [ZCBY10]. Eine Ursache dafür ist die Uneindeutigkeit mancher Suchanfragen. So kann beispielsweise die Anfrage `new york times square dance` unterschiedlich interpretiert werden. Ist ein Zeitungsartikel in der `new york times` zum Thema `square dance` gemeint oder ein Tanzevent am `times square` in `new york`? Die Intention des Nutzers ist in diesem Beispiel ohne Segmentierung der Anfrage in Sinneinheiten nicht klar. Die Einteilung der Worte einer gegebenen Suchanfrage in solche Wortgruppen wird Anfragesegmentierung genannt. Sie bildet den thematischen Rahmen der vorliegenden Arbeit.

[1] http://www.ard-zdf-onlinestudie.de/index.php?id=onlinenutzung00 (Zugegriffen am 25.08.2011)
[2] http://www.google.com/insidesearch/underthehood.html (Zugegriffen am 19.08.2011)
[3] http://www.google.com/about/corporate/company/tenthings.html (Zugegriffen am 25.08.2011)

Aktuelle Forschungsergebnisse belegen, dass die Segmentierung von Anfragen die Trefferquote der Suchergebnisse verbessert [LHZW11]. Durch das Einschließen von Wortgruppen in Anführungszeichen listet die Suchmaschine nur Dokumente auf, welche genau die vorgegebenen Worte in entsprechender Reihenfolge und Schreibweise enthalten. Nur eine geringe Anzahl von Nutzern kennt diese Option und so wird die Segmentierung bei weniger als 1,12 % der Suchanfragen im Internet angewendet [WM07]. Abhilfe schaffen Algorithmen, welche die Nutzeranfragen vor der eigentlichen Suche automatisch segmentieren. Aber wie gut können solche Verfahren die Intention des Nutzers überhaupt erfassen?

Zur Bewertung von Segmentierungsverfahren werden manuelle Segmentierungen in sogenannten *Evaluierungskorpora* eingesetzt. Öffentlich zugängliche Korpora umfassen Suchanfragen, welche nachträglich von menschlichen *Annotatoren* segmentiert wurden. Zu jeder Anfrage liegen mehrere Segmentierungen vor. Dabei kann es, wie in Tabelle 1.1 dargestellt, zu Übereinstimmungen zwischen den Annotatoren kommen.

Tabelle 1.1: Segmentierungen für `new york times square dance`

Stimmen	Segmentierung
6	`"new york" "times square" "dance"`
3	`"new york times" "square dance"`
1	`"new" "york times" "square dance"`

In diesem Beispiel entschieden sich 6 der Annotatoren für die Segmentierung `"new york" "times square" "dance"`, während 3 andere für die Einteilung `"new york times" "square dance"` stimmten. Lediglich ein Annotator wählte möglicherweise aus Unachtsamkeit die Segmentierung `"new" "york times" "square dance"`.

In der Literatur hat sich der *Bergsma-Wang-Corpus* aus dem Jahr 2007 zur Evaluierung etabliert [BW07]. Der Bergsma-Wang-Corpus enthält 500 Anfragen aus der AOL Suchanfragen-Logdatei von 2006 [PCT06], die jeweils von 3 Annotatoren segmentiert wurden. Aufgrund dieser geringen Menge an Anfragen und Segmentierungen gilt der Bergsma-Wang-Corpus jedoch als nicht repräsentativ.

Um eine angemessene Grundlage für die Bewertung von Anfragesegmentierungen zu schaffen, führten Hagen et al. im Jahr 2010 den *Webis Query Segmentation Corpus* ein [HPSB11]. Dieser neue Korpus enthält rund 50 000 Anfragen und ist somit um zwei Größenordnungen größer als der bisherige Evaluierungskorpus. Jede der Anfragen im Webis Query Segmentation Corpus wurde von mindestens 10 Annotatoren segmentiert, wodurch deutlich mehr

Segmentierungen für die Bewertung zur Verfügung stehen als zuvor. Die bisherigen Konzepte zur Evaluierung, die sogenannten *Evaluierungsmaße*, sind jedoch auf den Bergsma-Wang-Corpus mit lediglich 3 Segmentierungen pro Anfrage zugeschnitten. Sie hätten bei der Anwendung auf dem neuen Korpus eine unausgewogene Bewertung zur Folge. Daher sind neue Evaluierungsmaße notwendig, die eine faire Beurteilung von Anfragesegmentierungen mithilfe des Webis Query Segmentation Corpus gewährleisten.

Das Ziel der vorliegenden Arbeit ist es, diese Forschungslücke zu schließen und faire Maße für die Evaluierung von Anfragesegmentierungen auf Korpora mit einer größeren Anzahl an Segmentierungen zu konzipieren. Aufbauend auf den gewonnenen Erkenntnissen sollen zudem bessere Verfahren zur Segmentierung von Anfragen entwickelt werden.

Im folgenden Kapitel 2 wird zunächst das Problem der Anfragesegmentierung formal definiert. Darüber hinaus werden bestehende Segmentierungsverfahren vorgestellt. Gegenstand von Kapitel 3 ist die erstmalige genauere Analyse des Webis Query Segmentation Corpus mit den darin enthaltenen Anfragen und Segmentierungen. Diese Untersuchung bildet die Grundlage für die Entwicklung neuer Evaluierungsmaße, welche in Kapitel 4 vorgestellt und diskutiert werden. In Kapitel 5 werden neue Verfahren zur Anfragesegmentierung entwickelt, die durch die Ergebnisse der Evaluierung bestehender Segmentierungen inspiriert wurden. Das letzte Kapitel fasst die Ergebnisse der Arbeit zusammen und zeigt weitere interessante Forschungsfragen auf.

Kapitel 2

Anfragesegmentierung

2.1 Grundlagen

Bei der Segmentierung einer Anfrage geht es um die Einteilung dieser in Wortgruppen. Dabei ist die Anfrage q eine Sequenz $w_1\ w_2\ \ldots\ w_n$ von n Worten. Eine gültige Segmentierung für q ist eine Menge an disjunkten Teilsequenzen, deren Verknüpfung wiederum q selber entspricht. Die Zahl der möglichen Segmentierungen ist abhängig von der Länge n der Anfrage. Jedes der $n-1$ aufeinander folgenden Wortpaare kann entweder getrennt oder zusammen stehen. Das bedeutet, es gibt $n-1$ mögliche Trennstellen. Somit existieren für eine Anfrage mit n Worten genau 2^{n-1} mögliche Segmentierungen.

Zur Verdeutlichung der Grundlagen soll das Beispiel $q =$ `new york times` dienen. Die Anfrage enthält mit $n = 3$ Worten folglich $n - 1 = 2$ potentielle Trennstellen. Daraus ergeben sich $2^{n-1} = 2^2 = 4$ mögliche Segmentierungen, welche in Tabelle 2.1 veranschaulicht werden. Die linke Seite der Tabelle zeigt formal alle möglichen Trennungen benachbarter Worte der Anfrage. Die resultierenden Segmentierungen sind in der rechten Spalte dargestellt.

Tabelle 2.1: Mögliche Segmentierungen für `new york times`

Trennung zwischen Wortpaaren	Segmentierung
`new|york|times`	`"new" "york" "times"`
`new|york times`	`"new" "york times"`
`new york|times`	`"new york" "times"`
`new york times`	`"new york times"`

Algorithmen zur Segmentierung von Anfragen versuchen unter allen potentiellen Segmentierungen diejenige zu finden, welche den Informationsbedarf des Nutzers am besten formuliert.

2.2 Bestehende Segmentierungsverfahren

In der Literatur findet sich eine Reihe von Konzepten für die Segmentierung von Anfragen. Die folgenden Abschnitte geben einen Überblick über die bestehenden Segmentierungsverfahren.

PMI-basierte Verfahren

Einer der ersten Ansätze zur Segmentierung von Anfragen im Internet wurde von Risvik et al. [RMB03] entwickelt. Dabei wird die beste Segmentierung anhand sogenannter Connexity Values der Segmente ausgewählt. Diese errechnen sich aus der Summe der relativen Häufigkeit der Segmente in Anfrage-Logdateien und der Pointwise Mutual Information (PMI) innerhalb der Segmente. Die PMI zweier Terme gibt an, wie wahrscheinlich das gemeinsame Auftreten dieser ist [MRS08]. Risvik et al. evaluieren allerdings die Performance des PMI-basierten Verfahrens nicht.

Der Ansatz von Jones et al. [JRMG06] wertet benachbarte Worte als Phrasen, sobald deren PMI einen bestimmten Schwellenwert überschreitet. Die Idee, Wortpaare ab einem bestimmten PMI-Wert nicht zu trennen, wird in der Literatur häufig als Baseline verwendet. Die Performance der PMI-Baseline wird dabei meist durch Verfahren, welche mehr Features einbeziehen, übertroffen.

Ein weiterer PMI-basierter Ansatz wurde von Huang et al. [HGM$^+$10] entwickelt. Dabei wird für jede Anfrage auf Grundlage von segment-basierten PMI eine Baumstruktur von Segmenten generiert. Die beste Segmentierung wird anschließend aus dieser Hierachie abgeleitet. Huang et al. evaluieren dieses Verfahren allerdings nur auf einem proprietären Anfragekorpus ohne die Performance mit anderen Methoden zu vergleichen.

CRF-basierte Ansätze

Andere Verfahren beruhen auf dem Einsatz von statischen Modellen wie Conditional Random Fields (CRF). Guo et al. benutzen ein CRF-basiertes Modell zur Neuformulierung von Anfragen [GXLC08]. Dieses bearbeitet neben der Segmentierung gleichzeitig weitere Operationen wie z. B. das Trennen und Zusammenfügen von Worten sowie Rechtschreibkorrektur. Da die meisten Untersuchungen zur Anfragesegmentierung von der korrekten Schreibweise der Anfrage ausgehen, ist das Verfahren von Guo et al. nicht vollständig mit den übrigen vergleichbar.

Ein anderer CRF-basierter Ansatz von Yu und Shi [YS09] betrachtet die Segmentierung von Anfragen für die Suche in relationalen Datenbanken. Dabei werden den Segmenten der Suchanfrage Attributwerte aus der Datenbank zugeordnet.

Ähnliche datenbankspezifische Features nutzen Kiseleva et al. [KGA+10] bei der Optimierung von Produktanfragen in Webshops. Da beide Verfahren speziell für die Segmentierung von Anfragen an Datenbanken entwickelt wurden und Datenbank-Spezifika ausnutzen, können sie nicht auf die Suche im Internet angewendet werden.

Supervised Verfahren

Das Verfahren von Bergsma und Wang [BW07] beruht auf dem Konzept des überwachten Lernens (engl. supervised learning). Grundlegend ist dabei die Verwendung von Trainingsdaten, durch die das Verfahren optimiert wird. Bergsma und Wang kombinieren in ihrer Methode unterschiedliche Features, wie z. B. absolute Häufigkeiten von Worten und Phrasen im Internet, POS-Tagging und Kontextabhängigkeiten.

Darüber hinaus veröffentlichen sie den ersten Gold-Standard, den *Bergsma-Wang-Corpus* (BWC), zur Evaluierung von Anfragesegmentierungen. Dieser Korpus enthält 500 Anfragen der AOL Suchanfragen-Logdatei aus dem Jahr 2006 [PCT06], welche nachträglich von jeweils 3 Annotatoren segmentiert wurden. Dabei stimmen alle drei Segmentierungen in 44 % der Anfragen des Korpus, in den folgenden Abschnitten *sichere Anfragen* genannt, überein. In etwa 16 % der Anfragen wählen zwei der Annotatoren die gleiche Art zu segmentieren. Drei verschiedene Segmentierungen liegen hingegen in 40 % der Anfragen vor. Der BWC wird als Gold-Standard-Korpus auch in den darauffolgenden Studien zur Evaluierung von Anfragesegmentierungen verwendet.

Die Treffergenauigkeit des Verfahrens von Bergsma und Wang ist auf den sicheren Anfragen fast doppelt so groß, wie die der PMI-Baseline. Es bestehen jedoch Bedenken bezüglich der Allgemeingültigkeit dieser Werte, da die Trainingsdaten für das Verfahren von einem Annotator generiert wurden, welcher auch den BWC segmentierte.

Dennoch verwenden Bendersky et al. [BCS09] in ihrem 2-Stufen-Verfahren erfolgreich einige Features aus der Methode von Bergsma und Wang.

Häufigkeitsbasierte Methoden

Im Gegensatz zu den Verfahren des überwachten Lernens stehen Methoden, die mit wenigen statistischen Features auskommen.

Zhang et al. [ZSH+09] generieren beispielsweise aus den relativen Häufigkeiten aller 2-Wort-Kombinationen einer Anfrage eine Häufigkeitsmatrix. Die beste Segmentierung wird nach einigen Transformationen der Matrix aus dieser abgeleitet.

Anstatt relativer Auftrittshäufigkeiten verwenden Hagen et al. in ihrem naiven Ansatz [HPSB10] absolute Wort-N-Gramm-Häufigkeiten aus dem

Google N-Gram Corpus von 2006 [BF06]. Die Häufigkeiten werden anhand der Länge des Segments gewichtet und anschließend zur Berechnung der besten Segmentierung herangezogen.

Mishra et al. [MRG+11] leiten die Wahrscheinlichkeit einer Segmentierung aus Auftrittshäufigkeiten von Worten und Phrasen innerhalb von Anfrage-Logdateien ab. Im Gegensatz zu den Verfahren von Hagen et al. [HPSB10] und Zhang et al. [ZSH+09], welche die Performance der PMI-Baseline deutlich übertreffen, schneidet die Methode von Mishra et al. im Vergleich zur PMI-Baseline wesentlich schlechter ab.

PRF-basierte Verfahren

Der von Bendersky et al. in [BCS10] und [BCS11] verwendete Ansatz basiert ebenso auf N-Gramm-Häufigkeiten. Diese werden anders als in den bisher vorgestellten Methoden durch Pseudo-Relevance Feedback (PRF) gewonnen. Dazu wird zunächst die unsegmentierte Anfrage an eine Suchmaschine gestellt. Der Rang der resultierenden Suchergebnisse verkörpert die Relevanz zur Anfrage. Bendersky et al. filtern anschließend aus jedem der k besten Suchergebnisse die Wort-N-Gramm-Häufigkeiten heraus und gewichten diese entsprechend des erlangten Rangs. Auf Grundlage dessen wird anschließend für jedes Wortpaar die Wahrscheinlichkeit für das gemeinsame Auftreten berechnet. Liegt dieses über einem definierten Schwellenwert, so gehört das Wortpaar demselben Segment an. Ein Nachteil dieser Methode ist die deutlich verlängerte Laufzeit, die durch die Verwendung von PRF entsteht. Im Unterschied zu anderen Verfahren muss die Anfrage die Retrieval-Pipeline zweimal durchlaufen. Das erste Mal um die Suchergebnisse für das PRF zu gewinnen und ein weiteres Mal um die Resultate der segmentierten Anfrage zu erhalten. Das Verfahren erreicht gute Performance-Werte – jedoch nur auf einem kleinen Evaluierungskorpus mit lediglich 250 Anfragen.

Snippet-basierte Methoden

Ein ähnlich zeitaufwändiger Ansatz mit doppeltem Retrieval wurde von Brenes et al. [BGAG10] vorgeschlagen. Brenes et al. erlangen die Häufigkeiten potentieller Segmente aus den Snippets der Suchergebnisse für die unsegmentierte Anfrage. Im Rahmen einer experimentellen Evaluierung vergleichen Brenes et al. verschiedene Versionen des Snippet-basierten Ansatzes auf dem Bergsma-Wang-Corpus.

Klickraten-basierte Ansätze

In einem aktuellen Verfahren verknüpfen Li et al. [LHZW11] Wort-N-Gramm-Häufigkeiten mit Informationen zu Klickraten. Diese Daten werden genutzt, um zusätzlich die Präferenzen der Nutzer in das statistische Modell zu integrieren. Die Methode von Li et al. erreicht auf dem BWC vielversprechende Performance-Werte.

Wikipedia-basierte Verfahren

Die Verfahren von Tan und Peng [TP08] und Hagen et al. [HPSB11] kombinieren statistische Features mit semantischen, indem sie zusätzlich zu Wort-N-Gramm-Häufigkeiten Wissen aus der Online-Enzyklopädie Wikipedia einbeziehen. Tan und Peng bilden ein Sprachmodell mittels Wort-N-Gramm Häufigkeiten aus einem großen Web-Korpus. Die Parameter des Modells werden mithilfe eines Expectation-Maximization-Algorithmus abgeschätzt. Anhand des Sprachmodells wird für jedes Segment ein Score abgeleitet, welcher zusätzlich erhöht wird, falls das Segment häufig in Wikipedia benutzt wird. Das Verfahren segmentiert die sicheren Anfragen des BWC, d. h. die Anfragen mit übereinstimmenden Segmentierungen im Korpus, mit der doppelten Treffergenauigkeit der PMI-Baseline.

Das Wikipedia-basierte Verfahren [HPSB11] von Hagen et al. ist eine Weiterentwicklung des schon erwähnten naiven Ansatzes [HPSB10]. Die Idee der beiden Verfahren von Hagen et al. ist es, für jede mögliche Segmentierung einer Anfrage eine Punktzahl zu ermitteln. Die Berechnung der Punkte basiert auf den absoluten Auftrittshäufigkeiten der enthaltenen Segmente, wobei diese nicht unverarbeitet sondern normalisiert einfließen. Mithilfe der Normalisierung haben lange Segmente, die typischerweise weniger häufig im Internet vorkommen als deren Subsegmente, auch die Möglichkeit, ausgewählt zu werden. Der Unterschied der beiden Verfahren liegt in der Art der Normalisierung der Häufigkeiten. Beim naiven Ansatz spielt neben der Häufigkeit des Segments dessen Länge eine Rolle. Die Wikipedia-basierte Normalisierung berücksichtigt zudem, ob es sich bei dem Segment um den Titel eines Wikipedia-Artikels handelt. Die Segmentierung mit der höchste Punktzahl wird am Ende der Verfahren als Lösung ausgegeben.

Der Wikipedia-basierte Ansatz von Hagen et al. erlangt auf dem BWC die beste Treffergenauigkeit, ist zugleich leicht nachvollziehbar und einfach zu implementieren. Deshalb soll dieses Verfahren in der hier vorliegenden Arbeit die Grundlage für weitere Betrachtungen bilden. Dazu werden in den folgenden Abschnitten zunächst der naive und anschließend der Wikipedia-basierte Ansatz nochmals im Detail beschrieben.

2.2.1 Naive Normalisierung

Der naive Ansatz [HPSB10] von Hagen et al. nutzt lediglich Wort-N-Gramm-Häufigkeiten zur Segmentierung von Anfragen. Für jede mögliche Segmentierung S einer Anfrage wird eine Punktzahl (engl. score) berechnet. Dieser Score setzt sich aus den Gewichten (engl. weights) der enthaltenen Segmente zusammen, welche wie folgt berechnet werden:

$$\mathit{weight}(s) = |s|^{|s|} \cdot \mathit{freq}(s). \tag{2.1}$$

Die Auftrittshäufigkeit $\mathit{freq}(s)$ des Segments s wird aus dem Google N-Gram Corpus [BF06] ermittelt. Der Normalisierungs-Faktor $|s|^{|s|}$ ist abhängig von der Länge des Segments s und gewichtet lange Segmente stärker als kurze. Somit haben auch lange Segmente, die weniger häufig im Internet vorkommen als ihre Subsegmente, die Möglichkeit, eine hohe Punktzahl zu erreichen. Die Summe der Gewichte ergibt den Score einer Segmentierung S:

$$\mathit{score}(S) = \begin{cases} \sum\limits_{s \in S, |s| \geq 2} \mathit{weight}(s) & \text{falls } \mathit{weight}(s) > 0 \\ & \text{für alle } s \in S, |s| \geq 2 \\ -1 & \text{sonst.} \end{cases} \tag{2.2}$$

Dabei wird das Gewicht von Segmenten der Länge $|s| = 1$ (triviale Segmente) implizit auf 0 gesetzt, da lediglich Segmente der Länge $|s| \geq 2$ (nicht-triviale Segmente) in die Berechnung einfließen. Somit wird die unsegmentierte Anfrage, welche nur aus trivialen Segmenten besteht, mit einer Punktzahl von $\mathit{score}(S) = 0$ bewertet.

Eine Segmentierung S erhält eine negative Punktzahl von $\mathit{score}(S) = -1$, sobald ein nicht-triviales Segment s mit $\mathit{weight}(s) = 0$ darin vorkommt. Ein Segment erhält nach Gleichung (2.1) ein Gewicht von $\mathit{weight}(s) = 0$, wenn es eine Auftrittswahrscheinlichkeit von $\mathit{freq}(s) = 0$ hat – es also nicht im Internet vorkommt. Segmentierungen mit solchen Segmenten können die Retrieval Performance einer Anfrage nicht verbessern und erhalten deshalb den niedrigsten Score. Für den Fall, dass alle möglichen Segmentierungen einer Anfragen eine Punktzahl von $\mathit{score}(S) = -1$ erhalten, wird folglich die unsegmentierte Anfrage mit $\mathit{score}(S) = 0$ als Lösung ausgegeben.

Die naive Normalisierung erreicht auf den sicheren Anfragen des BWC eine Treffergenauigkeit von 69,3 % und übertrifft somit deutlich die PMI-Baseline, welche 55,5 % der Anfragen richtig segmentiert. Demnach ist die Performance des naiven Ansatzes mit anderen Verfahren, wie beispielsweise von Bergsma und Wang [BW07], Brenes et al. [BGAG10] oder Zhang et al. [ZSH+09], welche weitaus mehr Features einsetzen, vergleichbar.

2.2.2 Wikipedia-basiertes Verfahren

Die Wikipedia-basierte Normalisierung [HPSB11] erweitert den naiven Ansatz [HPSB10] um Wissen aus Wikipedia. Die Online-Enzyklopädie Wikipedia umfasst eine Vielzahl an Themengebieten. So reicht das Spektrum der Wikipedia-Artikel beispielsweise von Songtiteln über Namen von Gemeinden bis hin zu wissenschaftlichen Theorien. Die Titel von Wikipedia-Einträgen (Wikipedia-Titel) stellen bekannte, etablierte Konzepte dar und können somit zur Segmentierung von Anfragen genutzt werden. Grundlage dafür bildet im Verfahren von Hagen et al. eine Liste von Titeln der Wikipedia-Artikel, die aus einem Dump der Englischen Wikipedia extrahiert wurden.

Genau wie bei der naiven Normalisierung ergibt sich der Score einer Segmentierung nach Gleichung (2.2) aus der Summe der Gewichte der Segmente. Die Berechnung der Gewichte unterscheidet sich hingegen und ist beim Wikipedia-basierten Ansatz wie folgt definiert:

$$weight(s) = \begin{cases} |s| \cdot \max\limits_{s' \sqsubseteq s, |s'|=2} freq(s') & \text{falls } s \text{ ein Wikipedia-Titel ist} \\ |s| \cdot freq(s) & \text{sonst.} \end{cases} \tag{2.3}$$

Dabei bedeutet $s' \sqsubseteq s$, dass s' ein in s enthaltenes Wort-N-Gramm ist.

Im Gegensatz zur naiven Normalisierung, welche nur anhand der Länge eines Segments erfolgt, werden die Häufigkeiten hier auf einer segmentspezifischen Ebene normalisiert. In diesem Zusammenhang wird unterschieden, ob das Segment ein Wikipedia-Titel ist oder nicht. Segmente, die Titel von Wikipedia-Artikeln sind, erhalten durch die Normalisierung in Gleichung (2.3) eine höhere Gewichtung. Dazu wird anstatt der Häufigkeit des Wikipedia-Titels selbst die größte Häufigkeit eines der enthaltenen Wort-2-Gramme $s' \sqsubseteq s$ verwendet. Somit wird gewährleistet, dass Wikipedia-Titel bei der Segmentierung möglichst nicht getrennt werden.

Der Wikipedia-basierte Ansatz [HPSB11] erzielt auf den sicheren Anfragen des BWC im Vergleich zur PMI-Baseline und den Verfahren von Bergsma und Wang [BW07], Zhang et al. [ZSH+09], Brenes et al. [BGAG10] und Mishra et al. [MRG+11] mit 72,0 % die beste Treffergenauigkeit.

Das Beispiel `new york times` soll die Berechnung der beiden Verfahren verdeutlichen. Die absoluten Wort-N-Gramm-Häufigkeiten der relevanten Segmente aus dem Google N-Gram Corpus betragen:

- `new york`: 165,36 Millionen,
- `york times`: 17,60 Millionen,
- `new york times`: 17,55 Millionen.

Durch Anwendung der Gleichungen (2.1 – 2.3) ergeben sich die Punktzahlen der möglichen Segmentierungen für den naiven Ansatz [HPSB10] und das Wikipedia-basierten Verfahren [HPSB11] wie in Tabelle 2.2 dargestellt. Die resultierenden Scores und die Wort-N-Gramm Häufigkeiten zur Berechnung sind dabei in Millionen angegeben.

Tabelle 2.2: Berechnung der Scores für `new york times` in Millionen

Segmentierung	Naiver Ansatz		Wikipedia-basierter Ansatz	
`new|york|times`	$0+0+0=$	$0,00$	$0+0+0=$	$0,00$
`new|york times`	$0+17,60\cdot 2^2=$	$70,40$	$0+17,60\cdot 2=$	$35,20$
`new york|times`	$165,36\cdot 2^2+0=$	**661,44**	$165,36\cdot 2+0=$	$330,72$
`new york times`	$17,55\cdot 3^3=$	$157,95$	$165,36\cdot 3=$	**496,08**

Anhand der Segmentierung $S =$ `"new york times"` sollen die Berechnungen nochmals veranschaulicht werden. Da S genau ein Segment umfasst, entspricht das Gewicht $\mathit{weight}(s)$ des Segments s der Punktzahl $\mathit{score}(S)$ der gesamten Segmentierung S.

Beim naiven Ansatz wird die Häufigkeit $\mathit{freq}(s) = 17{,}55$ Millionen von $s =$ `"new york times"` mit der Segmentlänge $|s| = 3$ anhand des Faktors $|s|^{|s|} = 3^3$ gewichtet. Somit ergibt sich beim naiven Ansatz für S eine Punktzahl von $\mathit{score}(S) = 157{,}95$ Millionen.

Das Wikipedia-basierte Verfahren identifiziert $s =$ `"new york times"` als Wikipedia-Titel. Daher wird die Häufigkeit $\mathit{freq}(s') = 165{,}36$ Millionen des häufigsten enthaltenen Wort-2-Gramms $s' =$ `new york` gewählt und mit der Länge $|s| = 3$ des Segments s multipliziert. Daraus resultiert für die Segmentierung S eine Punktzahl von $\mathit{score}(S) = 496{,}08$ Millionen.

Mit dem naiven Verfahren erreicht die Segmentierung `"new york" "times"` mit 661,44 Millionen Punkten den besten Score und wird somit als Lösung ausgegeben. Nach dem Wikipedia-basierten Ansatz ist die beste Segmentierung `"new york times"` mit einer Punktzahl von 496,08 Millionen. In diesem Beispiel gibt das Wikipedia-basierte Verfahren genau die Segmentierung aus, die wahrscheinlich auch die meisten Nutzer wählen würden.

2.3 Aktuelle Evaluierungskorpora

Hagen et al. analysieren in [HPSB11] nicht nur ihre beiden Verfahren zur Segmentierung, sondern auch den bisherigen Gold-Standard zur Evaluierung – den Bergsma-Wang-Corpus (BWC). Dabei bemängeln sie unter anderem doppelte Anfragen sowie Rechtschreib- und Zeichenkodierungsfehler in den Anfragen des BWC.

Ein weiterer Kritikpunkt ist die geringe Anzahl von lediglich 3 Segmentierungen pro Anfrage. In 40 % der Anfragen des BWC liegen 3 verschiedene Segmentierungen vor. Somit gibt es für diese Anfragen keine Mehrheitsentscheidung, die für Evaluierung benutzt werden kann.

Zur Lösung dieses Problems vermehren Hagen et al. die Segmentierungen im BWC mithilfe von *Amazon Mechanical Turk*, einer Internetplattform für bezahltes Crowdsourcing. Zuvor entfernen sie jedoch die doppelten Anfragen und korrigieren die Kodierungs- und Rechtschreibfehler. Das Ergebnis dieser Weiterentwicklung des BWC ist der sogenannte *Enriched Bergsma-Wang-Corpus*. Er enthält 492 Anfragen mit jeweils 10 manuellen Segmentierungen. Hagen et al. verwenden den Enriched Bergsma-Wang-Corpus, um bestehende Segmentierungen der Anfragen des Bergsma-Wang-Corpus mit einer repräsentativeren Anzahl an Stimmen evaluieren zu können.

Neben den wenigen Segmentierungen im BWC gibt es noch weitere Mängel. Der BWC enthält lediglich 500 Anfragen, die unter bestimmten Bedingungen aus der AOL Suchanfragen-Logdatei entnommen wurden. So umfasst der Korpus nur sogenannte Nominalanfragen, die lediglich aus Nomen, Adjektiven und Artikeln bestehen. Eine weitere Voraussetzung für die Aufnahme in den BWC war der Erfolg der Anfrage. Das bedeutet, der Nutzer, der die Anfrage stellte, klickte auch eines der Suchergebnisse an. Aufgrund dieser Auswahlkriterien und der geringen Menge an Segmentierungen erklären Hagen et al. den BWC für nicht repräsentativ.

Um eine angemessene Grundlage für die Bewertung von Anfragesegmentierungen zu schaffen, führen Hagen et al. im Jahr 2010 den *Webis Query Segmentation Corpus*, einen gänzlich neuen Evaluierungskorpus, ein [HPSB11]. Dieser umfasst 53 437 repräsentative Anfragen aus der AOL Suchanfragen-Logdatei von 2006 [PCT06] mit je mindestens 10 Segmentierungen. Die Konstruktion des Korpus und die enthaltenen Anfragen und Segmentierungen werden im nachfolgenden Kapitel dieser Arbeit erstmals genauer analysiert.

Kapitel 3

Analyse des Webis Query Segmentation Corpus

Der Webis Query Segmentation Corpus (WQSC) wurde im Jahr 2010 von Hagen et al. [HPSB11] zur Evaluierung von Anfragesegmentierungen eingeführt. Dieser neue Korpus soll den bisherigen Evaluierungskorpus – den Bergsma-Wang-Corpus (BWC) – ablösen, der unter anderem aufgrund der geringen Menge an Anfragen und Segmentierungen als nicht repräsentativ gilt. Der WQSC beinhaltet hingegen 53 437 Anfragen, die jeweils von mindestens 10 Annotatoren segmentiert wurden. In diesem Kapitel werden neben der Konstruktion des Webis Query Segmentation Corpus erstmals die enthaltenen Anfragen und Segmentierungen genauer untersucht.

3.1 Konstruktion des Korpus

Die Anfragen im WQSC stammen aus der AOL Suchanfragen-Logdatei (AOL-Log) von 2006 [PCT06]. Um die Qualität des Korpus zu sichern, entfernten Hagen et al. zunächst ungeeignete Anfragen aus dem AOL-Log.

So wurden beispielsweise Anfragen herausgenommen, die weniger als 3 oder mehr als 10 Worte umfassten. Der Grund dafür ist, dass Anfragen mit mehr als 10 Worten sehr selten gestellt werden und Anfragen mit nur 1 oder 2 Worten wenig von einer Segmentierung profitieren können. In Anfragen mit nur einem Wort gibt es keine zusammengehörigen Wortgruppen. Für 2-Wort-Anfragen wirkt die in Suchmaschinen verwendete *Proximity* der beiden Worte, die relative Nähe der Worte im Dokument [MRS08], gewissermaßen wie eine Segmentierung. Je näher zusammen die beiden Worte im Text stehen, desto höher wird das Dokument für die Suche gerankt. Somit werden für 2-Wort-Anfragen schon durch den Einfluss der Proximity zuerst die Dokumente gelistet, in denen die beiden Worte direkt nebeneinander stehen. Bei längeren Anfragen verlangt die

Proximity jedoch, dass alle Worte der Anfrage nah beieinander im Dokument vorkommen, sodass hier eine segmentierte Anfrage die geeignetere Variante ist.

Weiterhin wurden Anfragen ausgeschlossen, die neben Bindestrichen und Apostrophen innerhalb der Worte weitere nicht-alphanumerische Zeichen enthielten. Dazu gehörten unter anderem Anfragen mit Resten von URLs, bei denen der Nutzer das Eingabefeld der Suchmaschine mit der Adressleiste des Browsers verwechselt haben könnte.

Darüber hinaus entfernten Hagen et al. Anfragen, die sehr wahrscheinlich automatisch generiert wurden. Diese konnten anhand des Verhaltens der Suchenden identifiziert werden. So wurden z. B. Anfragen von Nutzern ausgesondert, welche mehr als 10 000 Anfragen stellten oder deren durchschnittliche Zeichenanzahl pro Anfrage größer als 100 war. Anfragen von Suchenden, deren durchschnittliche Zeit zwischen zwei aufeinander folgenden Anfragen weniger als eine Sekunde war, wurden ebenso herausgenommen. Neben diesen formalen Kriterien definierten Hagen et al. zusätzlich inhaltliche Auswahlbeschränkungen. So sortierten sie beispielsweise auch Anfragen aus, welche nicht in englischer Sprache gestellt waren.

Nach dem Filtern des AOL-Logs entnahmen Hagen et al. diesem, unter Berücksichtigung der Längenverteilung, zufällig 55 555 Anfragen. Die Verteilung der Häufigkeiten der Anfragen ist indirekt eingeflossen. Da oft gestellte Anfragen auch häufiger im AOL-Log auftraten, wurden diese mit einer größeren Wahrscheinlichkeit ausgewählt. Befand sich eine gewählte Anfrage bereits im Korpus, so nahmen Hagen et al. sie nicht nochmals auf. Das bedeutet, jede Anfrage kommt nur einmal im Korpus vor.

Mithilfe einer semi-automatischen Rechtschreibprüfung korrigierten Hagen et al. die ausgewählten Anfragen. Aufgrund der Korrektur wurden manche Worte zusammengefasst oder getrennt. Somit änderte sich die Länge einiger Anfragen, sodass diese weniger als 3 oder mehr als 10 Worte enthielten. Diese wurden anschließend entfernt, da sie nicht mehr den formalen Auswahlbeschränkungen des Korpus entsprachen. Nach allen Korrekturen blieben genau 53 437 Anfragen im WQSC bestehen.

Die Segmentierungen der Anfragen gewannen Hagen et al. mithilfe von *Amazon Mechanical Turk* (AMT), einer Internetplattform für bezahltes Crowdsourcing. AMT vermittelt Aufgaben von Requestern an Worker, welche für die erfolgreiche Bearbeitung einer Aufgabe Geld erhalten. Der Requester hat die Möglichkeit, die eingereichten Ergebnisse zu prüfen und abzulehnen, wenn diese nicht zufriedenstellend sind. Die Aufgabe, die Hagen et al. im Rahmen der Konstruktion des neuen Korpus mehrfach stellten, umfasste die Segmentierung von 6 Anfragen. Dabei war stets eine Testanfrage enthalten, welche die Qualität der Ergebnisse der Segmentierungen verifizieren sollte. Zur Testanfrage existierten bereits Segmentierungen, die mit der des AMT-Workers abgeglichen wurden. Segmentierte der AMT-Worker die Testanfrage anders,

so wurde seine bisherige Leistung bezüglich der Testanfragen in allen bearbeiteten Aufgaben betrachtet. Bei vielen fehlgeschlagenen Testanfragen begutachteten Hagen et al. alle Segmentierungen des Workers. Arbeitete dieser ohne ersichtliche Bemühung, so lehnten Hagen et al. die Ergebnisse der Aufgabe ab. Außerdem wurden Ergebnisse abgewiesen, welche eine ungerade Anzahl an Anführungszeichen enthielten, da in diesen Fällen nicht klar war, wie der Annotator segmentieren wollte.

Bei der Konstruktion des WQSC waren insgesamt 2 048 AMT-Worker beteiligt. Darunter befanden sich jedoch auch 253 Worker von denen alle Ergebnisse abgelehnt wurden, sodass die Segmentierungen im Korpus tatsächlich von 1 795 Annotatoren erzeugt wurden. Diese generierten rund 550 000 Segmentierungen, wobei jede davon mit 0,01 $ entlohnt wurde. Somit belaufen sich die Kosten für die Generierung der Segmentierungen des WQSC auf etwa 5 500 $. Die AMT-Worker benötigten im Durchschnitt circa 11,39 Sekunden für die Segmentierung einer Anfrage. Daraus resultiert ein durchschnittlicher Lohn von 2,85 $ pro Stunde.

Hagen et al. teilen die Anfragen im Webis Query Segmentation Corpus in zwei Sets ein. Das *Trainingsset* mit 4 850 Anfragen wird *WQSC-5k* genannt und soll im Rahmen eines Wettbewerbs für Algorithmen zur Anfragesegmentierung veröffentlicht werden. Die restlichen 48 587 Anfragen sind im *WQSC-50k*, dem *Testset*, enthalten. Dieses soll zur Evaluierung der im Wettbewerb eingereichten Algorithmen dienen. In der folgenden Analyse dieses Kapitels wird der WQSC als gesamter Korpus mit 53 437 Anfragen betrachtet.

3.2 Anfragen und Segmentierungen

Dieser Abschnitt beschäftigt sich mit der Analyse der im Webis Query Segmentation Corpus enthaltenen Anfragen und Segmentierungen. Grundlage für die Untersuchung des Korpus bilden mehrere Kriterien, die zunächst separat beleuchtet werden. Im Anschluss erfolgt eine detaillierte Analyse des Korpus durch Kombination der Kriterien.

Segmentlängen

Ein wichtiger Ansatzpunkt zur Bewertung des menschlichen Verhaltens bei der Segmentierung von Anfragen ist die Segmentlänge. Tabelle 3.1 zeigt die Verteilung der Segmentlängen in den Segmentierungen des WQSC.

Dabei ist erkennbar, dass Menschen kurze Segmente bevorzugen. Über eine Million triviale Segmente mit nur einem Wort sind in den Segmentierungen des WQSC enthalten. Darauf folgen die 2-Wort-Segmente mit einem deutlichen Abstand zu längeren Segmenten. Die besonders langen Segmenten mit 8 bis 10 Worten enthalten meist Fragen oder Zitate aus Songtexten.

Tabelle 3.1: Segmentlängen im WQSC

Segmentlänge	Anzahl
1	1 020 249
2	368 554
3	104 033
4	20 278
5	4 738
6	1 536
7	662
8	277
9	116
10	25
Gesamt	1 520 468

Eine Ursache für die Präferenz von kurzen Wortgruppen könnte ihre Übersichtlichkeit sein, denn kurze Segmente formulieren den Informationsbedarf meist klar und sind gut zu erfassen.

Länge einer Anfrage

Die Länge einer Anfrage entspricht der Menge der enthaltenen Worte. So hat beispielsweise die Anfrage `new york times square dance` aus der Einleitung die Länge 5. Der Korpus beinhaltet Anfragen der Länge 3 bis 10. Die Längen-

Tabelle 3.2: Anfragelängen im WQSC

Anfragelänge	Anfragen	Prozentualer Anteil
3	23 833	44,60 %
4	14 571	27,27 %
5	7 678	14,37 %
6	3 803	7,12 %
7	1 864	3,49 %
8	947	1,77 %
9	481	0,90 %
10	260	0,49 %
Gesamt	53 437	100,00 %

verteilung ist in Tabelle 3.2 dargestellt. Sie gleicht der des gefilterten AOL-Logs ohne automatisch generierte Suchanfragen und URL-Anfragen. Der Korpus besteht größtenteils aus 3-Wort-Anfragen. Mit steigender Anfragelänge sinkt die Menge der Anfragen pro Längenklasse.

Kategorie einer Anfrage

Zu einer Anfrage q gibt es im Korpus k Segmentierungen $S_1, \ldots, S_k$. Die Stimmen v_i (von engl. votes) einer Anfrage geben an, wie viele der Annotatoren sich für die Segmentierung S_i entschieden haben. Die Summe der Votes $v_1 + \ldots + v_k$ je Anfrage ist gleich 10.

Anhand der Verteilung der Votes pro Anfrage kann die Entscheidung der Annotatoren und somit auch die Anfrage selbst als *sicher* oder *unsicher* eingestuft werden. Bei sicheren Anfragen einigt sich die Mehrheit der Annotatoren auf eine bestimmte Segmentierung. Hingegen wird bei unsicheren Anfragen aus der Verteilung der Stimmen pro Anfrage keine klare Mehrheitsentscheidung deutlich. Die Annotatoren sind sich also unsicher, wie die Anfrage zu segmentieren ist.

Die Segmentierungen S_i einer Anfrage q seien nach der Anzahl ihrer Votes $v_1 \geq v_2 \geq \ldots \geq v_k$ sortiert. Dann gilt für die Kategorie von q:

$$Kategorie(q) = \begin{cases} \textit{sicher} & \text{falls } v_1 \in \{10, 9, 8, 7\} \text{ oder} \\ & \text{falls } (v_1, v_2) \in \{(6,3), (6,2), (6,1), (5,1)\} \\ \textit{unsicher} & \text{sonst.} \end{cases} \tag{3.1}$$

Bei Anfragen mit mehr als 10 Stimmen werden diese prozentual auf 10 Votes normalisiert, sodass die Gleichung (3.1) angewendet werden kann.

Die Verteilung der Anfragen des WQSC nach den Votes (v_1, v_2) ist in Abbildung 3.1 veranschaulicht. Sichere Anfragen sind schraffiert dargestellt und umfassen etwa 54 % des WQSC. Die übrigen 46 % der Anfragen im Korpus sind unsicher.

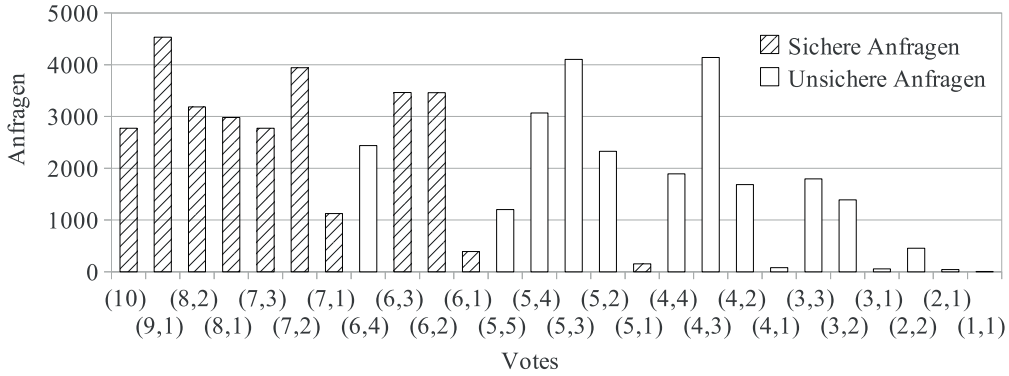

Abbildung 3.1: Verteilung der Votes (v_1, v_2) im WQSC

Das Beispiel `new york times square dance` soll die Kategorisierung verdeutlichen. Die Segmentierungen der Anfrage erhielten die Votes $v_1 = 6, v_2 = 3$ und $v_3 = 1$. Laut Gleichung (3.1) wird die Anfrage somit als sicher eingestuft.

Topvote einer Anfrage

Bei der Topvote einer Anfrage handelt es sich um die Segmentierung mit den meisten Votes. Wir untersuchen ob die Topvote einer Anfrage segmentiert oder unsegmentiert ist. Die unsegmentierte Anfrage ist ein Spezialfall der Segmentierung, da sie lediglich aus 1-Wort-Segmenten besteht, wie z. B. `"new" "york" "times" "square" "dance"`. Diese Art der Segmentierung ist in etwa 71 % der Anfragen in den Segmentierungen enthalten. Bei etwa 24 % der Anfragen im WQSC erreicht die unsegmentierte Anfrage die meisten Stimmen und ist somit die Topvote.

Anfragearten

Besteht eine Anfrage nur aus Nomen, Adjektiven und Artikeln so handelt es sich um eine Nominalanfrage. Anfragen, die zusätzlich andere Wortarten enthalten, werden als Nicht-Nominalanfragen bezeichnet. Um die Wortarten in den Anfragen zu identifizieren, wird in der vorliegenden Arbeit der POS-Tagger *QTag*[1] verwendet, welcher Texte mit einer Genauigkeit von 96,22 % taggt [TM98]. *QTag* erkennt beispielsweise in der Anfrage `new york times square dance` das Wort `new` als Adjektiv und die restlichen Worte als Nomen. Somit ist diese Anfrage eine Nominalanfrage, wie auch 44 % der Anfragen im WQSC. Die übrigen 56 % werden als Nicht-Nominalanfragen getaggt.

Kombination der Kriterien

Durch die Kombination der einzelnen Kriterien können die Eigenschaften der Anfragen und Segmentierungen des WQSC noch genauer charakterisiert werden. In Abbildung 3.2 sind die möglichen Kombinationen der relevanten Kriterien Anfrageart, Länge einer Anfrage, Segmentierung der Topvote und Kategorie einer Anfrage dargestellt.

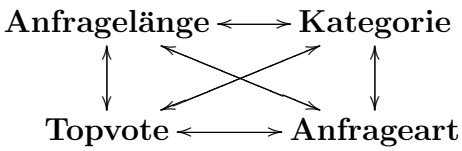

Abbildung 3.2: Kriterien zur Analyse des WQSC

Zunächst werden die Verknüpfungen der Anfragelänge mit allen anderen Kriterien analysiert. Die Eigenschaften Anfragelänge und Art der Anfrage sind in Abbildung 3.3 gegenübergestellt. Es ist erkennbar, dass etwa 2/3 der 3-

[1] http://phrasys.net/uob/om/software (Zugegriffen am 08.10.2011)

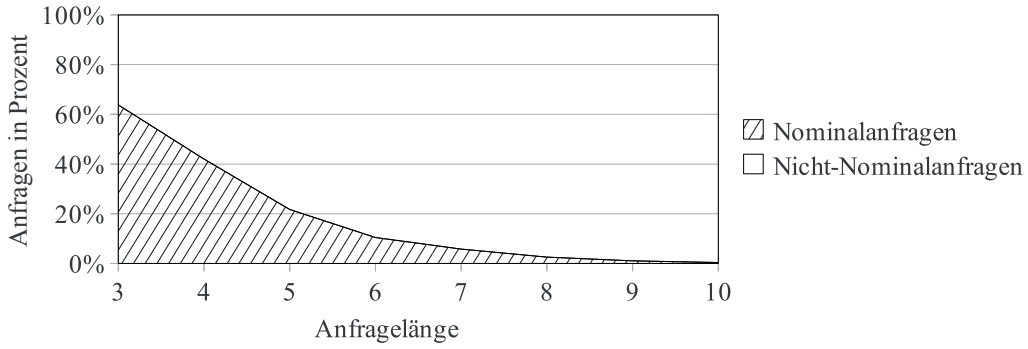

Abbildung 3.3: WQSC: Anfragelänge und Anfrageart

Wort-Anfragen Nominalanfragen sind. Ab einer Anfragelänge von 4 ändert sich allerdings die Verteilung. Die Mehrheit der Anfragen pro Längenklasse wird nun durch Nicht-Nominalanfragen bestimmt. So sind beispielsweise bei den Anfragen der Länge 6 nur noch 10 % Nominalanfragen enthalten. Lange Anfragen sind teilweise als ganze Sätze oder Fragen formuliert und deuten meist auf einen unklar ausgedrückten Informationsbedarf des Nutzers hin. In diesem Zusammenhang werden Wortarten benutzt, die per Definition nicht in Nominalanfragen enthalten sind. Somit sind lange Anfragen tendenziell Nicht-Nominalanfragen.

In Abbildung 3.4 werden die Anfragen des WQSC nach Länge und Segmentierung der Topvote eingeteilt. Entgegen der Erwartungen wird deutlich,

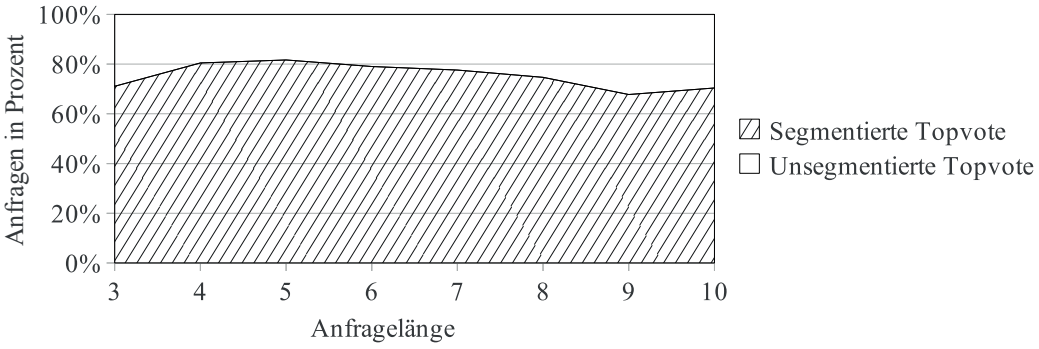

Abbildung 3.4: WQSC: Anfragelänge und Topvote

dass sich die Verteilungen von segmentierten und unsegmentierten Topvotes innerhalb der einzelnen Längenklassen stark ähneln. Dabei besitzen durchschnittlich etwa 76 % der Anfragen eine segmentierte Topvote. Folglich erhält die unsegmentierte Anfrage in 24 % des Korpus die meisten Stimmen.

Die Verknüpfung der Kriterien Anfragelänge und Kategorie ist in Abbildung 3.5 dargestellt. Es zeigt sich, dass Anfragen der Länge 3 am sichersten

Abbildung 3.5: WQSC: Anfragelänge und Kategorie

sind. Relativ sicher sind auch die 4-Wort-Anfragen. Ab einer Anfragelänge von 5 schlägt jedoch die Verteilung um, wobei nun etwa 60 % der Anfragen unsicher sind. Eine Ursache dafür könnte wiederum sein, dass lange Anfragen den Informationsbedarf unklar formulieren. Somit fällt es schwer, die ursprüngliche Intention im Nachhinein zu erfassen. Daher waren die Annotatoren bei der Segmentierung langer Anfragen unsicherer als bei kurzen Anfragen.

Die Gegenüberstellung von Art der Anfrage und Kategorie erfolgt in Tabelle 3.3. Dabei wird deutlich, dass etwa 54 % (12 604 von 23 508) der Nominal-

Tabelle 3.3: WQSC: Anfrageart und Kategorie

Anfrageart	**Kategorie**		
	Sicher	Unsicher	Gesamt
Nominalanfragen	12 604	10 904	23 508
Nicht-Nominalanfragen	16 169	13 760	29 929
Gesamt	28 773	24 664	53 437

anfragen sicher sind. Dies trifft allerdings auch auf die Nicht-Nominalanfragen zu. Betrachtet man die Tabelle spaltenweise, so sind rund 44 % (12 604 von 28 773) der sicheren Anfragen Nominalanfragen. Eine sehr ähnliche Verteilung ist auch bei den unsicheren Anfragen zu beobachten. Die Tabelle widerlegt demnach die Vermutung, dass Nominalanfragen sicherer sind als Nicht-Nominalanfragen.

Tabelle 3.4 analysiert die Anfragen im WQSC anhand der Kategorie und der Segmentierung der Topvote. Durch die Verknüpfung der beiden Kriterien stellt sich heraus, dass die Annotatoren vorwiegend sicher segmentieren und unsicher nicht segmentieren. In etwa 24 % (12 828 von 53 437) der Anfragen

Tabelle 3.4: WQSC: Topvote und Kategorie

Topvote	**Kategorie**		
	Sicher	Unsicher	Gesamt
Segmentiert	22 942	17 667	40 609
Unsegmentiert	5 831	6 997	12 828
Gesamt	28 773	24 664	53 437

des Korpus ist die Topvote unsegmentiert, dabei sind sich die Annotatoren jedoch zu 55 % (6 997 von 12 828) unsicher. Daraus ergibt sich, dass die Annotatoren unsichere Anfragen bevorzugt nicht segmentieren. Anfragen mit einer segmentierten Topvote gelten hingegen in etwa 56 % (22 942 von 40 609) der Fälle als sicher.

Die Kriterien Anfrageart und Segmentierung der Topvote werden in Tabelle 3.5 untersucht. Beim Betrachten der Zeilen der Tabelle wird klar, dass

Tabelle 3.5: WQSC: Topvote und Anfrageart

Topvote	**Anfrageart**		
	Nominalanfragen	Nicht-Nominalanfragen	Gesamt
Segmentiert	19 771	20 838	40 609
Unsegmentiert	3 737	9 091	12 828
Gesamt	23 508	29 929	53 437

Anfragen mit einer segmentierten Topvote in etwa gleichen Teilen Nominalanfragen und Nicht-Nominalanfragen sind. Hingegen handelt es sich bei Anfragen, deren Topvotes unsegmentiert sind, in 71 % (9 091 von 12 828) der Fälle um Nicht-Nominalanfragen. Anhand der Spalten der Tabelle ist erkennbar, dass die Topvotes in 16 % (3 737 von 23 508) der Nominalanfragen unsegmentiert sind. Bei den Nicht-Nominalanfragen trifft dies auf 30 % (9 091 von 29 929) zu.

3.3 Kritik

Eine Anfrage kann am besten von demjenigen segmentiert werden, der sie gestellt hat. Segmentierungen im Webis Query Segmentation Corpus wurden, wie auch im bisher verwendeten Bergsma-Wang-Corpus (BWC), im Nachhinein generiert. Daher ist der Informationsbedarf des Nutzers, der die Anfrage stellte, nicht immer nachvollziehbar. So ist es möglich, dass die Annotatoren bestimmte Wortgruppen in der Anfrage, z. B. den Namen einer Band oder den Titel eines Liedes, nicht kannten und somit auch nicht sinngemäß segmentieren konnten. Vor diesem Hintergrund kann die Grundlage zur Evaluierung einiger Anfragen im Korpus ungerecht sein. Fairer wäre ein Korpus aus Anfragen, die bereits von Nutzern bei der Suche segmentiert wurden. Damit wäre gewährleistet, dass die Anfrage genau so segmentiert ist, wie es der Suchende gemeint hatte. Da jedoch die wenigsten Nutzer ihre Suchanfragen segmentieren und Suchmaschinen nur selten Anfrage-Logdateien veröffentlichen, gibt es zu wenig solcher Anfragen, um einen repräsentativen Korpus zu erstellen. Somit muss auf Evaluierungskorpora wie den bisher verwendeten BWC oder den neuen WQSC zurückgegriffen werden.

Der WQSC enthält im Gegensatz zum BWC nicht nur 3 sondern mindestens 10 Stimmen pro Anfrage und umfasst zudem viel mehr Anfragen. Bisherige Evaluierungsmaße auf dem BWC wählen als Referenz diejenige der 3 Segmentierung, welche der zu evaluierenden Segmentierung am meisten ähnelt. Wendet man dieses Maß auf den WQSC an, so kann beispielsweise bei einer sicheren Anfrage auch die Segmentierung mit den wenigsten Votes als Referenz ausgewählt werden. So könnte z. B. bei der sicheren Anfrage `new york times square dance` mit den Votes $v_1 = 6, v_2 = 3$ und $v_3 = 1$, die Segmentierung mit nur einer Stimme als Referenz dienen. In diesem Beispiel wäre das die Segmentierung `"new" "york times" "square dance"`, wobei diese Einteilung keinen Sinn macht und der Annotator unachtsam gearbeitet haben muss und sich vermutlich verklickt hat.

Zudem ist bei 10 Stimmen die Wahrscheinlichkeit höher, dass die vorhergesagte Segmentierung selber oder eine sehr ähnliche enthalten ist. Daher erreicht der Algorithmus nach den bisherigen Evaluierungsmaßen auf dem WQSC eventuell bessere Performance-Werte als zuvor auf dem BWC. Um eine faire Evaluierung auf dem WQSC sicherzustellen, müssen neue Evaluierungsmaße entwickelt werden, die besser auf Korpora mit mehr als 3 Segmentierungen pro Anfrage zugeschnitten sind. Dieses Aufgabe wird im nachfolgenden Kapitel aufgegriffen.

Kapitel 4

Evaluierung von Anfragesegmentierungen

Algorithmen zur Anfragesegmentierung sollen dem Nutzer das Segmentieren von Anfragen abnehmen. Die Performance solcher Algorithmen wird in zwei verschiedenen Bereichen untersucht. Da die Segmentierungen on-the-fly generiert werden sollen, ist zum einen die Laufzeit der Algorithmen von großer Bedeutung. Das Hauptaugenmerk dieses Kapitels liegt jedoch zunächst auf dem zweiten Aspekt zur Evaluierung der Performance – der Genauigkeit der vorhergesagten Segmentierung. Die Bewertung der Laufzeit wird im zweiten Teil dieses Kapitels thematisiert.

4.1 Bewertung der Genauigkeit

Die Genauigkeit der vorhergesagten Segmentierung gibt an, wie gut diese den Informationsbedarf des Nutzers modellieren kann. Dabei werden auf Grundlage von menschlich erzeugten Segmentierung verschiedene Performance-Werte ermittelt.

4.1.1 Genauigkeitsebenen

Den Ausgangspunkt für die Bewertung der Genauigkeit von Anfragesegmentierungen bilden die Segmentierungen in einem Evaluierungskorpus. Diese werden bei der Evaluierung einer vorhergesagten Segmentierung S_{pred} (von engl. prediction) als Referenz S_{ref} (von engl. reference) herangezogen. Die Genauigkeit von S_{pred} kann hinsichtlich der Referenz S_{ref} auf drei Ebenen bewertet werden. Neben der Trefferquote der gesamten Segmentierung (Anfrage-Ebene) werden auch die einzelnen Segmente (Segment-Ebene) und die möglichen Trennstellen in der Anfrage (Break-Ebene) untersucht.

Tabelle 4.1: Segmentierungen für S_{pred} und S_{ref}

S	Segmentierung
S_{pred}	`"new york" "times square dance"`
S_{ref}	`"new york" "times square" "dance"`

In den folgenden Abschnitten werden die drei Genauigkeitsebenen detailliert vorgestellt. Zur Veranschaulichung der Berechnungen soll das Beispiel in Tabelle 4.1 dienen.

Anfrage-Ebene

Die vorhergesagte Segmentierung S_{pred} ist auf Anfrage-Ebene korrekt, wenn sie genau die gleichen Segmente wie die Referenz S_{ref} enthält. Diese Art der Genauigkeit wird als *Query Accuracy* bezeichnet und ist wie folgt definiert:

$$queryAcc(S_{ref}, S_{pred}) = \begin{cases} 1 & falls \quad S_{pred} = S_{ref} \\ 0 & sonst. \end{cases}$$

Auf einem gegebenen Korpus gibt die Query Accuracy den Anteil der korrekt segmentierten Anfragen an.

Betrachtet man das Beispiel aus Tabelle 4.1, so ist die Genauigkeit auf Anfrage-Ebene $queryAcc(S_{ref}, S_{pred}) = 0$, da die zu evaluierende Segmentierung S_{pred} nicht mit der Referenz S_{ref} übereinstimmt.

Segment-Ebene

Auf Segment-Ebene wird eine Segmentierung als Menge von Segmenten betrachtet. So enthält die Referenzsegmentierung S_{ref} die relevanten und die vorhergesagte Segmentierung S_{pred} die erfassten Segmente. In diesem Zusammenhang können die im Information Retrieval weit verbreiteten Maße Precision und Recall auf Segment-Ebene angewendet werden. Dabei beschreibt *Segment Precision* den Anteil der Segmente in S_{pred}, die relevant sind und *Segment Recall* beschreibt den Anteil der relevanten Segmente, die in S_{pred} enthalten sind. Die beiden Maße sind durch folgende Formeln definiert:

$$segPrec(S_{ref}, S_{pred}) = \frac{|S_{ref} \cap S_{pred}|}{|S_{pred}|} \qquad segRec(S_{ref}, S_{pred}) = \frac{|S_{ref} \cap S_{pred}|}{|S_{ref}|},$$

wobei die Schnittmenge $S_{ref} \cap S_{pred}$ genau die Segmente umfasst, welche sowohl in der Referenz S_{ref} als auch in der vorhergesagten Segmentierung S_{pred} auftreten.

Durch die Berechnung des harmonischen Mittels können diese Maße in einem Wert, dem *Segment F-Measure*, kombiniert werden:

$$segF(S_{ref}, S_{pred}) = \frac{2 \cdot segPrec(S_{ref}, S_{pred}) \cdot segRec(S_{ref}, S_{pred})}{segPrec(S_{ref}, S_{pred}) + segRec(S_{ref}, S_{pred})}.$$

Zur Veranschaulichung berechnen wir die Performance-Werte auf Segment-Ebene für das Beispiel in Tabelle 4.1. Die Segmentierung S_{pred} umfasst zwei und die Referenz S_{ref} drei Segmente, somit sind $|S_{pred}| = 2$ und $|S_{ref}| = 3$. Da die Schnittmenge $S_{ref} \cap S_{pred}$ der beiden Segmentierungen lediglich das Segment `"new york"` umfasst, ist $|S_{ref} \cap S_{pred}| = 1$. Daraus ergibt sich Segment Precision zu $segPrec(S_{ref}, S_{pred}) = \frac{1}{2}$ und Segment Recall mit $segRec(S_{ref}, S_{pred}) = \frac{1}{3}$. Durch Verknüpfung der beiden Werte entsteht $segF(S_{ref}, S_{pred}) = \frac{2}{5}$.

Break-Ebene

Eine Anfrage q mit n Worten hat genau $n - 1$ Paare aufeinander folgender Worte, die zusammen oder getrennt stehen können. Für jede dieser potenziellen Trennstellen (Break-Positionen) entscheidet die vorhergesagte Segmentierung S_{pred} entweder richtig oder falsch bezüglich der Referenz S_{ref}. Vor diesem Hintergrund berechnet sich die *Break Accuracy*, die Genauigkeit auf Break-Ebene, wie folgt:

$$breakAcc(S_{ref}, S_{pred}) = \frac{Anzahl\ richtiger\ Breaks(S_{ref}, S_{pred})}{Anzahl\ möglicher\ Breaks(S_{ref})}.$$

Dabei ist $Anzahl\ richtiger\ Breaks(S_{ref}, S_{pred})$ die Zahl der übereinstimmenden Break-Entscheidungen in der zu evaluierenden Segmentierung S_{pred} und der Referenz S_{ref}. Die $Anzahl\ möglicher\ Breaks(S_{ref})$ ist $n - 1$ für eine Anfrage mit n Worten.

Die Break Accuracy im Beispiel in Tabelle 4.1 ist $breakAcc(S_{ref}, S_{pred}) = \frac{3}{4}$, da 3 der 4 Break-Entscheidungen in der vorhergesagte Segmentierung S_{pred} bezüglich der Referenz S_{ref} richtig sind.

Anhand der Performance-Werte in den einzelnen Ebenen wird deutlich, dass die Genauigkeit auf Anfrage-Ebene am strengsten ist. Im Vergleich zu Anfrage- und Segment-Ebene erreicht eine Anfrage auf Break-Ebene stets die größeren Accuracy-Werte. Ist z. B. wie in der zu evaluierenden Segmentierung S_{pred} in Tabelle 4.1 lediglich eine Break-Entscheidung falsch, so ergibt sich die Break Accuracy noch als $\frac{3}{4}$, wobei die Genauigkeit auf Anfrage-Ebene sofort 0 ist.

Jede der vorgestellten Genauigkeitsebenen benötigt eine Referenzsegmentierung zur Bewertung der vorhergesagten Segmentierung. Die Aufgabe der Evaluierungsmaße ist es nun, diese Referenz aus den Segmentierungen im Evaluierungskorpus auszuwählen.

4.1.2 Evaluierungsmaße

Mit der Auswahl der Referenz schaffen Evaluierungsmaße die Voraussetzung für die Bewertung der Genauigkeit. Viele der Verfahren zur Anfragesegmentierung wurden auf dem etablierten Bergsma-Wang-Corpus (BWC) evaluiert. Dabei wählten die Autoren als Referenz diejenige der 3 Segmentierungen aus dem BWC, welche der vorhergesagten Segmentierung am meisten ähnelt. Der Webis Query Segmentation Corpus (WQSC) und der Enriched Bergsma-Wang-Corpus (EBWC) enthalten hingegen mindestens 10 Segmentierungen für jede Anfrage. Bei 10 Stimmen ist im Vergleich zu 3 Stimmen die Wahrscheinlichkeit höher, dass die Segmentierung des Algorithmus selber oder eine sehr ähnliche Segmentierung enthalten ist. Somit erreicht der Algorithmus nach den bisherigen Evaluierungsmaßen auf dem EBWC bessere Performance-Werte als zuvor auf dem BWC. Um eine faire Bewertung zu gewährleisten, werden daher in der vorliegenden Arbeit neue Evaluierungsmaße speziell für die Anwendung auf Korpora mit mehreren Stimmen, wie im EBWC und im WQSC, konzipiert. Diese werden in der Reihenfolge ihrer Komplexität und verbesserten Fairness in den folgenden Abschnitten vorgestellt. Dabei wird zunächst das bisherige Evaluierungsmaß *bestfit* im Detail erläutert.

bestfit

Das bisherige Evaluierungsmaß *bestfit* wählt aus allen Segmentierungen diejenige als Referenz S_{ref} aus, welche der vorhergesagten Segmentierung S_{pred} auf Break-Ebene am meisten ähnelt. Ist S_{pred} in der Menge der Segmentierungen im Korpus enthalten, wird diese als Referenz S_{ref} gewählt. Infolgedessen sind die Performance-Werte auf allen drei Ebenen gleich 1, da somit S_{pred} und S_{ref} gleich sind.

Tabelle 4.2: Evaluierung mit S_{pred} = `"los angeles" "times"`

i	Votes v_i	Segmentierung S_i	$breakAcc(S_{pred}, S_i)$
1	7	`los angeles times`	0.5
2	2	`los angeles|times`	1.0
3	1	`los|angeles|times`	0.5

Das Beispiel in Tabelle 4.2 veranschaulicht die Funktionsweise von bestfit. Gegeben sind die Votes v_i und die Segmentierungen S_i zur Anfrage `los angeles times`. Die vorhergesagte Segmentierung S_{pred} sei `"los angeles" "times"`. Im ersten Schritt von bestfit wird die Genauigkeit auf Break-Ebene $breakAcc(S_{pred}, S_i)$ aller Segmentierungen S_i bezüglich der zu evaluierenden

Segmentierung S_{pred} berechnet. Die Ergebnisse der Rechnung sind in der letzten Spalte der Tabelle 4.2 aufgelistet. Die zweite Segmentierung S_2 hat die beste Break Accuracy bezüglich der vorhergesagten Segmentierung S_{pred} und wird somit als Referenz ausgewählt. Mit der Auswahl der Referenz S_{ref} können nun auch die übrigen Performance-Werte auf Anfrage- und Segment-Ebene berechnet werden. Da die Referenz S_{ref} und S_{pred} identisch sind, erreichen die Performance-Werte auf allen drei Ebenen den Wert 1.

Anhand der Verteilung der Votes v_i in Tabelle 4.2 wird jedoch deutlich, dass die Segmentierung S_2, die als Referenz gewählt wurde, nicht die Mehrheit der Stimmen erhalten hat. Lediglich 2 Annotatoren wählten diese Art zu segmentieren. Die klare Mehrheit an Votes erhielt bei dieser sicheren Anfrage hingegen die Topvote S_1 mit $v_1 = 7$. Daher wäre es hier fairer gewesen, die Topvote S_1 als Grundlage für die Evaluierung zu wählen. Mit der faireren Referenz-Auswahl fallen die Performance-Werte deutlich kleiner aus, da die zu evaluierende Segmentierung S_{pred} und die Referenz S_{ref} nicht mehr übereinstimmen. Der Mangel des Maßes bestfit ist daher, dass trotz klarer Stimmenmehrheit eine Segmentierung als Referenz gewählt werden kann, die nicht die Topvote ist. Dabei können bessere Performance-Werte entstehen als bei einer fairen Evaluierung gegen die Topvote.

bestfit3

Das Maß *bestfit3* ähnelt dem bisherigen bestfit sehr. Dabei wird die Referenzsegmentierung wiederum anhand der Break Accuracy ausgewählt. In diesem Maß wird jedoch nicht aus allen Segmentierungen im Evaluierungskorpus, sondern nur aus den dreien mit den meisten Stimmen gewählt. Bei Anfragen zu denen lediglich 3 verschiedene Segmentierungen im Korpus enthalten sind, wie im Beispiel in Tabelle 4.2, entspricht bestfit3 dem Maß bestfit.

Ebenso wie bei bestfit ist bei diesem Evaluierungsmaß zu bemängeln, dass trotz klarer Stimmenmehrheit eine Segmentierung mit deutlich weniger Votes im Vergleich zur Topvote als Referenz gewählt werden kann.

bestfit normalized

Eine fairere Version von bestfit ist das neu entwickelte Maß *bestfit normalized*. Die Referenz wird ebenso wie bei bestfit anhand der Break Accuracy aus allen Segmentierungen im Korpus gewählt. Nach Berechnung der Performance-Werte werden diese bei bestfit normalized zusätzlich mit dem Normalisierungskoeffizienten N_{coeff} multipliziert. Der Koeffizient ergibt sich aus den Votes v_{ref} für die Referenzsegmentierung und den Stimmen $v_{topvote}$ für die Topvote:

$$N_{coeff} = \frac{v_{ref}}{v_{topvote}}.$$

Durch die Normalisierung wird die Verteilung der Stimmen pro Anfrage in die Bewertung der Performance einbezogen.

Zur Verdeutlichung der Wirkungsweise von bestfit normalized betrachten wir das Beispiel in Tabelle 4.2. Das Maß bestfit normalized wählt die Segmentierung S_2 als Referenz S_{ref} aus, da sie die beste Break Accuracy bezüglich der vorhergesagten Segmentierung S_{pred} besitzt, und evaluiert S_{pred} gegen S_2. Alle Performance-Werte nehmen dabei den Wert 1 an, da S_2 und S_{pred} übereinstimmen. Die Referenz S_2 erhielt im Korpus zwei Votes, somit ist $v_{ref} = v_2 = 2$. Für die Topvote S_1 stimmten $v_{topvote} = v_1 = 7$ Annotatoren. Demzufolge ergibt sich ein Normalisierungskoeffizient von $N_{coeff} = \frac{2}{7}$. Durch die Normalisierung sind die Performance-Werte nun $\frac{2}{7}$ der Werte von bestfit. So ist beispielsweise die Genauigkeit auf Anfrage-Ebene nur noch $queryAcc(S_2, S_{pred}) = 1 \cdot \frac{2}{7} = \frac{2}{7} = 0,29$ statt vorher 1.

Das Maß bestfit normalized relativiert oder bestätigt die guten Performance-Werte von bestfit anhand der Verteilung der Votes der Annotatoren. Dennoch können bei sicheren Anfragen bessere Ergebnisse erreicht werden, als bei einer fairen Evaluierung gegen die Topvote. Angenommen die vorhergesagte Segmentierung S_{pred} entscheidet an jeder Trennstelle genau anders als die Topvote und es existiert mindestens noch eine zweite Segmentierung im Korpus. So wird die Topvote nicht als Referenz gewählt, da sie die schlechteste Break Accuracy bezüglich S_{pred} besitzt. Demzufolge dient eine andere Segmentierung mit mindestens einer richtigen Trennstelle als Referenz, wodurch sich beispielsweise auf Break-Ebene eine Genauigkeit größer als 0 ergibt. Bei einer fairen Evaluierung mit der Topvote als Referenzsegmentierung ist die Break Accuracy jedoch genau 0.

category

Um die Nachteile der bestfit-Verfahren auszugleichen, wählt das neue Evaluierungsmaß *category* die Referenz in Abhängigkeit von der Kategorie der Anfrage. Bei sicheren Anfragen mit einer klaren Stimmenmehrheit wird die Topvote als Referenzsegmentierung gewählt. Handelt es sich um eine unsichere Anfrage, so ist die Referenz, wie beim Evaluierungsmaß bestfit, die Segmentierung mit der besten Break Accuracy bezüglich der vorhergesagten Segmentierung. Durch die Unterscheidung nach der Kategorie der Anfrage wird gewährleistet, dass bei einer klaren Stimmenmehrheit die Topvote als Referenz zur Evaluierung dient. Allerdings fehlt durch die Anwendung von bestfit wieder die Normalisierung der Votes bei unsicheren Anfragen.

category normalized

Die Methode *category normalized* erweitert category um Wissen zur Verteilung der Stimmen der Annotatoren. Sichere Anfragen werden wiederum gegen die Topvote evaluiert. Bei unsicheren Anfragen kommt das Maß bestfit normalized zum Einsatz. Dabei werden, wie oben beschrieben, alle Performance-Werte mit den Votes der Referenzsegmentierung und den Stimmen für die Topvote normalisiert. Diese Kombination aus den Maßen category und bestfit normalized berücksichtigt sowohl bei sicheren als auch bei unsicheren Anfragen die Verteilung der Stimmen pro Anfrage und bewertet somit unter den bisher vorgestellten Evaluierungsmaßen am fairsten.

newbreak

Einen gänzlich neuen Ansatz zur Evaluierung von Anfragesegmentierungen verwendet das Maß *newbreak*. Andere Maße wählen eine Referenzsegmentierung, die zur Anwendung der etablierten Performance-Maße benötigt wird. Das Maß newbreak arbeitet hingegen nur auf Break-Ebene und produziert lediglich einen Performance-Wert.

Tabelle 4.3: Segmentierungen für $q =$ `new york city news`

Votes	Segmentierung
4	`new york city|news`
3	`new york|city news`
2	`new york|city|news`
1	`new york city news`

Tabelle 4.4: Aufgesammelte Votes ($Break_i$ | $NonBreak_i$) je Trennstelle b_i in $q =$ `new york city news`

	b_1		b_2		b_3	
`new`		`york`		`city`		`news`
	0	10	5	5	6	4

Anhand der Segmentierungen und Stimmen für eine Anfrage im Korpus wird für jede mögliche Trennstelle ermittelt, wie viele Annotatoren getrennt (Break) und wie viele nicht getrennt (NonBreak) haben. Tabelle 4.3 zeigt eine Beispiel-Anfrage mit Segmentierungen aus dem Korpus. Die Anzahl der Votes für Break und NonBreak je Trennstelle sind in Tabelle 4.4 dargestellt. Betrachtet man das erste Paar benachbarter Worte `new york` der Beispiel-Anfrage, so entschieden sich laut Tabelle 4.3 alle 10 Annotatoren an dieser Trennstelle für NonBreak. Das zweite Wortpaar `york city` wurde von 5 Annotatoren getrennt und von 5 anderen hingegen als zusammengehörig gekennzeichnet. Bei der letzten Trennstelle zwischen `city news` wählten 6 Annotatoren Break, die übrigen 4 entschieden sich für NonBreak.

Die Idee des Maßes newbreak ist es nun, für eine gegebene Segmentierung S_{pred} die Votes pro Break-Position aufzusammeln. Angenommen die vorhergesagte Segmentierung S_{pred} ist `new york|city news`. Diese sammelt an der

Trennstelle `new york` 10 Votes für NonBreak, bei `york|city` 5 Votes für Break und an der letzten Break-Position `city news` 4 Votes für NonBreak auf. Somit erhält S_{pred} insgesamt $v_{pred} = 10 + 5 + 4 = 19$ Votes.

Die maximal erreichbaren Votes v_{max} ergeben sich durch Addition der maximalen Votes für Break oder NonBreak je Trennstelle $b_i \in S$:

$$v_{max} = \sum_{i=1}^{n-1} \max\left(Break_i, NonBreak_i\right).$$

Im Beispiel in Tabelle 4.4 erhält man dementsprechend $v_{max} = NonBreak_1 + Break_2 + Break_3 = 10 + 5 + 6 = 21$.

Für eine Anfrage mit n Worten und 10 Stimmen sind die maximal erreichbaren Votes v_{max} mindestens $(n-1) \cdot 5$ und höchstens $(n-1) \cdot 10$. Dabei gilt, je größer v_{max} ist, desto sicherer waren sich die Annotatoren beim Segmentieren. Die Anfrage in unserem Beispiel enthält 4 Worte, daraus folgt $v_{max} \in [15, 30]$. Mit $v_{max} = 21$ sind sich die Annotatoren relativ unsicher bei der Segmentierung der Anfrage.

Der Performance-Wert des Maßes newbreak beschreibt das Verhältnis der gesammelten Votes v_{pred} der vorhergesagten Segmentierung S_{pred} zu den maximal erreichbaren Votes v_{max}:

$$newbreak = \frac{v_{pred}}{v_{max}}.$$

Dieses Verhältnis gibt an, wie gut die vorhergesagte Segmentierung mit der Mehrheitsentscheidung der Annotatoren auf Break-Ebene übereinstimmt. Im Beispiel ergibt sich somit ein Performance-Wert von $newbreak = \frac{19}{21} = 0,9$. Die vorhergesagte Segmentierung S_{pred} erhält also 90 % der maximal erreichbaren Votes und stimmt somit sehr gut mit der Meinung der Annotatoren auf Break-Ebene überein.

Der Vorteil des Maßes newbreak ist die übersichtliche und faire Bewertung der Genauigkeit. Dabei wird anhand der akkumulierten Votes auf Break-Ebene die Güte der vorhergesagten Segmentierung in nur einem Wert bestimmt. Demzufolge ist newbreak jedoch nicht direkt mit den anderen Evaluierungsmaßen, die die Performance auf den drei etablierten Genauigkeitsebenen bewerten, vergleichbar.

corpus

Das Maß *corpus* greift die Idee von newbreak auf und erweitert diese, sodass ein Evaluierungsmaß mit den typischen Genauigkeitsmaßen entsteht. Dabei wird aus den Mehrheitsentscheidungen pro Break-Position b_i die Referenzsegmentierung S_{ref} generiert. Somit kann wieder auf den drei etablierten Genauigkeitsebenen bewertet werden.

Betrachten wir nochmals das Beispiel in Tabelle 4.3 und 4.4 aus dem vorherigen Abschnitt. An der ersten Break-Position `new york` entscheidet sich corpus, wie alle Annotatoren, für NonBreak. Die zweite Trennstelle `york city` stellt einen Spezialfall dar, wobei 5 Annotatoren für Break und 5 andere für NonBreak stimmten. Bei einer solchen $(5, 5)$-Entscheidung fügt das Maß corpus einen Break ein. Im Zweifelsfall handelt das Maß also defensiv und gruppiert benachbarte Worte nicht. Die letzten beiden Worte `city news` trennt corpus, da sich die Mehrheit von 6 Annotatoren dafür entschieden hat. Somit ergibt sich die Referenzsegmentierung S_{ref} = `new york|city|news`, die zur Evaluierung auf den drei gewohnten Genauigkeitsebenen genutzt werden kann.

Die generierte Referenz S_{ref} repräsentiert die Stimmenmehrheit auf Break-Ebene. Bei sicheren Anfragen hält die Topvote und daher auch die enthaltenen Break-Entscheidungen die Mehrheit der Stimmen. Somit wird an jeder Trennstelle die Break-Entscheidung der Topvote gewählt. Demzufolge entspricht die generierte Referenzsegmentierung S_{ref} bei sicheren Anfragen genau der Topvote. Bei unsicheren Anfragen ist es hingegen möglich, dass die generierte Referenz S_{ref} selbst nicht im Korpus enthalten ist. Dennoch bleiben Segmente, die von der Mehrheit der Annotatoren gewählt wurden, in der Referenzsegmentierung erhalten.

Mit der Meinung der Stimmenmehrheit als Referenzsegmentierung schafft das Maß corpus eine faire Grundlage zur Evaluierung der Performance. Die Bewertung nach sicheren und unsicheren Anfragen, wie in den Maßen category und category normalized, erfolgt bei corpus implizit. Somit besitzt das Evaluierungsmaß corpus die gleichen Vorteile wie das bisher fairste Maß category normalized.

4.1.3 Anwendung der Maße

In Kapitel 2.2 wurden eine Reihe von Algorithmen zur Anfragesegmentierung vorgestellt, wobei ein Großteil dieser Verfahren von den entsprechenden Autoren auf dem Bergsma-Wang-Corpus (BWC) evaluiert wurde. Die meisten der Autoren überließen uns die Ergebnisse ihrer Segmentierungsverfahren für die Anfragen des BWC. So können wir neben den beiden Verfahren von Hagen et al. [HPSB10, HPSB11] zusätzlich die Methoden von Bergsma und Wang [BW07], Zhang et al. [ZSH+09], Brenes et al. [BGAG10], Mishra et al. [MRG+11] und Li et al. [LHZW11] mit den neuen Evaluierungsmaßen auf dem Enriched Bergsma-Wang-Corpus (EBWC) bewerten.

Die Ergebnisse der Evaluierung auf dem EBWC sind in Tabelle 4.5 dargestellt. Es wird deutlich, dass der Algorithmus von Bergsma und Wang [BW07] mit dem bisherigen Evaluierungsmaß bestfit am besten abschneidet. Dieses Verfahren wurde allerdings mit den Segmentierungen eines Annotators trai-

niert, welcher auch den BWC segmentierte und ist daher bezüglich dieses Annotators optimiert. Vermutlich wählt bestfit immer genau die Segmentierung dieses Annotators als Referenz und schneidet deshalb so gut in der Evaluierung ab. Bei den übrigen Maßen erreichen Hagen et al. mit ihrem Wikipedia-basierten Ansatz [HPSB11] die beste Query Accuracy.

Tabelle 4.5: Evaluierung auf dem EBWC (492 Anfragen)

Maß		**Algorithmus**							
		PMI	[BW07]	[ZSH+09]	[BGAG10]	[HPSB10]	[MRG+11]	[HPSB11]	[LHZW11]
bestfit	query	0,752	**0,821**	0,803	0,734	0,789	0,533	**0,821**	0,742
	segPrec	0,794	**0,869**	0,858	0,797	0,845	0,636	**0,869**	0,798
	segRec	0,796	**0,879**	0,858	0,799	0,847	0,685	0,865	0,811
	segF	0,795	**0,874**	0,858	0,798	0,846	0,659	0,867	0,804
	break	0,884	**0,925**	0,916	0,870	0,906	0,769	0,918	0,879
bestfit3	query	0,705	0,783	0,770	0,693	0,760	0,520	**0,799**	0,699
	segPrec	0,757	0,839	0,834	0,766	0,824	0,627	**0,856**	0,764
	segRec	0,758	**0,851**	0,833	0,767	0,824	0,677	**0,851**	0,778
	segF	0,758	0,845	0,833	0,767	0,824	0,651	**0,853**	0,771
	break	0,866	0,910	0,904	0,853	0,894	0,764	**0,911**	0,862
best-norm	query	0,575	0,682	0,662	0,600	0,671	0,346	**0,713**	0,576
	segPrec	0,616	0,730	0,717	0,663	0,727	0,450	**0,760**	0,632
	segRec	0,619	0,740	0,717	0,664	0,729	0,499	**0,756**	0,645
	segF	0,617	0,735	0,717	0,664	0,728	0,473	**0,758**	0,638
	break	0,706	0,786	0,775	0,736	0,788	0,583	**0,809**	0,713
category	query	0,608	0,715	0,697	0,638	0,711	0,364	**0,744**	0,602
	segPrec	0,677	0,798	0,785	0,733	0,792	0,518	**0,815**	0,699
	segRec	0,686	**0,821**	0,788	0,739	0,796	0,590	0,810	0,722
	segF	0,682	0,809	0,786	0,736	0,794	0,552	**0,812**	0,710
	break	0,833	0,889	0,880	0,835	0,878	0,702	**0,892**	0,829
cat-norm	query	0,531	0,654	0,634	0,573	0,650	0,304	**0,691**	0,539
	segPrec	0,600	0,736	0,722	0,667	0,730	0,458	**0,762**	0,636
	segRec	0,609	**0,759**	0,725	0,673	0,735	0,530	0,757	0,658
	segF	0,605	0,747	0,723	0,670	0,733	0,491	**0,759**	0,647
	break	0,756	0,827	0,817	0,770	0,816	0,642	**0,838**	0,766
corpus	query	0,429	0,577	0,530	0,510	0,583	0,252	**0,587**	0,445
	segPrec	0,549	0,712	0,676	0,651	0,709	0,449	**0,715**	0,602
	segRec	0,556	**0,739**	0,668	0,650	0,708	0,539	0,700	0,621
	segF	0,552	**0,725**	0,672	0,651	0,708	0,490	0,707	0,611
	break	0,766	**0,842**	0,818	0,781	0,827	0,662	0,837	0,773
newbreak		0,846	**0,901**	0,887	0,846	0,885	0,726	0,899	0,842

Bei der Betrachtung der Performance-Werte fällt auf, dass diese sinken je fairer das jeweilige Evaluierungsmaß ist. Weiterhin kann man eine Veränderung der Rangfolge der Verfahren bezüglich der Query Accuracy erkennen.

So teilen sich beispielsweise das Verfahren von Bergsma und Wang [BW07] und der Wikipedia-basierte Ansatz [HPSB11] bei bestfit mit einer Query Accuracy von 0,821 Platz 1 im Ranking. Bei der Anwendung des Maßes corpus erreichen Bergsma und Wang [BW07] lediglich den dritten Platz in der Rangfolge hinter dem naiven Verfahren [HPSB10] und dem Wikipedia-basierten Ansatz [HPSB11] mit der besten Query Accuracy von 0,587. Diese Änderung des Rankings der Algorithmen zeigt, dass sich die Evaluierungsmaße tatsächlich unterscheiden und nicht nur strenger bewerten.

Tabelle 4.6: Evaluierung auf dem WQSC-5k (4 850 Anfragen)

Maß		**4 850 Anfragen**			**2 142 Nominalanfragen**		
		PMI	[HPSB10]	[HPSB11]	PMI	[HPSB10]	[HPSB11]
bestfit	query	**0,756**	0,584	0,626	0,845	0,846	**0,865**
	segPrec	**0,833**	0,724	0,761	0,870	0,881	**0,898**
	segRec	**0,821**	0,694	0,731	0,868	0,876	**0,892**
	segF	**0,827**	0,708	0,746	0,869	0,878	**0,895**
	break	**0,887**	0,803	0,826	0,918	0,913	**0,924**
bestfit3	query	**0,699**	0,542	0,589	0,778	0,800	**0,822**
	segPrec	**0,789**	0,695	0,736	0,813	0,846	**0,866**
	segRec	**0,776**	0,663	0,703	0,810	0,840	**0,859**
	segF	**0,782**	0,679	0,719	0,811	0,843	**0,863**
	break	**0,859**	0,784	0,808	0,882	0,889	**0,902**
best-norm	query	**0,572**	0,452	0,498	0,628	0,677	**0,708**
	segPrec	**0,648**	0,592	0,633	0,652	0,712	**0,741**
	segRec	**0,636**	0,563	0,603	0,651	0,707	**0,736**
	segF	**0,642**	0,577	0,617	0,652	0,709	**0,739**
	break	**0,703**	0,672	0,698	0,701	0,744	**0,767**
category	query	**0,626**	0,498	0,542	0,686	0,737	**0,763**
	segPrec	**0,735**	0,663	0,701	0,736	0,798	**0,821**
	segRec	**0,723**	0,629	0,666	0,736	0,790	**0,811**
	segF	**0,729**	0,646	0,683	0,736	0,794	**0,816**
	break	**0,824**	0,762	0,784	0,838	0,860	**0,873**
cat-norm	query	**0,537**	0,430	0,475	0,585	0,647	**0,680**
	segPrec	**0,645**	0,594	0,634	0,634	0,708	**0,738**
	segRec	**0,634**	0,561	0,599	0,635	0,700	**0,728**
	segF	**0,640**	0,577	0,616	0,635	0,704	**0,733**
	break	**0,734**	0,694	0,717	0,737	0,770	**0,790**
corpus	query	**0,413**	0,347	0,371	0,438	0,536	**0,580**
	segPrec	**0,589**	0,569	0,588	0,543	0,663	**0,669**
	segRec	**0,572**	0,522	0,537	0,539	0,639	**0,658**
	segF	**0,580**	0,545	0,561	0,541	0,651	**0,663**
	break	**0,722**	0,688	0,698	0,709	0,755	**0,777**
newbreak		**0,838**	0,805	0,817	0,834	0,868	**0,875**

Die Anwendung der Evaluierungsmaße auf dem WQSC-5k, dem Trainingsset des Webis Query Segmentation Corpus mit 4 850 Anfragen, ist im linken Teil von Tabelle 4.6 dargestellt. Dabei zeigt sich, dass die PMI-Baseline sowohl das naive Verfahren [HPSB10] als auch den Wikipedia-basierten Ansatz [HPSB11] von Hagen et al. übertrifft. Auf dem EBWC, welcher nur Nominalanfragen enthält, schneidet das Wikipedia-basierte Verfahren [HPSB11] jedoch in den meisten Fällen am besten ab. Dies führt zu der Annahme, dass dieses Verfahren besonders bei Nominalanfragen gut funktioniert. Daher wurden die drei Verfahren im rechten Teil von Tabelle 4.6 nur auf den 2 142 Nominalanfragen des Trainingssets des WQSC evaluiert. Wie erwartet erreicht der Wikpedia-basierte Ansatz [HPSB11] die besten Performance-Werte.

In den nachfolgenden Kapiteln der vorliegenden Arbeit betrachten wir lediglich noch die Performance-Werte in den Evaluierungsmaßen category normalized, corpus und newbreak, da diese am fairsten evaluieren und die übrigen Maße somit vernachlässigt werden können.

4.2 Evaluierung der Laufzeit

Neben der Genauigkeit ist die Laufzeit ein weiterer wichtiger Aspekt zur Bewertung der Performance von Segmentierungsverfahren. Die Laufzeit wird als Datendurchsatz angegeben, wobei die Anzahl der verarbeiteten Anfragen pro Sekunde gemessen wird.

Anders als bei der Evaluierung der Genauigkeit untersuchen wir lediglich die Laufzeit der beiden Verfahren von Hagen et al. [HPSB10, HPSB11] und der PMI-Baseline, da die Implementierungen der anderen Verfahren fehlen. Um ein repräsentatives Ergebnis zu erzielen, messen wir die Laufzeit der Algorithmen für das Segmentieren der 48 587 Anfragen des Testsets des WQSC auf einem Standard-PC. Dabei werden die Häufigkeiten der Wort-N-Gramme für den naiven Ansatz [HPSB10] und das Wikipedia-basierte Verfahren [HPSB11] von Hagen et al. in einer externen Hashtabelle bereitgestellt, die in 12 GB Hauptspeicher passt. Für die Berechnungen der PMI-Baseline stellen wir die Auftrittswahrscheinlichkeiten der Wort-1-Gramme in Text-Dateien zur Verfügung.

Die beste Laufzeit erreicht die PMI-Baseline mit rund 27 000 Anfragen je Sekunde. Ein Grund für den hohen Datendurchsatz der PMI-Baseline ist der geringe Rechenaufwand dieser Methode. Sowohl das naive Verfahren [HPSB10] als auch der Wikipedia-basierte Ansatz [HPSB11] segmentieren etwa 3 300 Anfragen in einer Sekunde. Das entspricht umgerechnet auf einen Tag etwa einem Viertel der eine Milliarde Suchanfragen, die Google täglich beantwortet.

Durch Vernetzung mehrerer Rechner in einem Cluster wird eine höhere Rechenleistung erzielt, sodass die Menge der Anfragen an Google mit den Verfahren von Hagen et al. on-the-fly segmentiert werden können. Dabei muss beachtet werden, dass die Anzahl der Suchanfragen nicht zu jeder Tageszeit gleich ist [PCT06]. Zudem sind etwa 46 % der Suchanfragen 1- und 2-Wort-Anfragen[1], die von einer Segmentierung wenig profitieren würden. Daher würde vermutlich ein Cluster von 10 Standard-PCs ausreichen, um die eine Milliarde Suchanfragen an Google mit den Verfahren von Hagen et al. [HPSB10, HPSB11] zu segmentieren.

Die Ergebnisse der Laufzeitmessung zeigen, dass sowohl das naive Verfahren [HPSB10] als auch der Wikipedia-basierte Ansatz [HPSB11] zur on-the-fly Segmentierung von Suchanfragen geeignet sind.

[1] http://www.hitwise.com/index.php/us/about-us/press-center/press-releases/google-searches-mar-10/ (Zugegriffen am 01.12.2011)

Kapitel 5

Entwicklung neuer Segmentierungsverfahren

Auf Grundlage der in Kapitel 4 gewonnenen Erkenntnisse werden nun zwei neue Verfahren zur Segmentierung von Anfragen entwickelt – die *Wikipedia-Titel-Baseline* und das *hybride Verfahren*. Beide Algorithmen verwenden den Wikipedia-basierten Ansatz von Hagen et al. [HPSB11] als Teilverfahren. In den folgenden Abschnitten werden zunächst die Wikipedia-Titel-Baseline und anschließend das hybride Verfahren vorgestellt und bezüglich der Performance analysiert.

5.1 Die Wikipedia-Titel-Baseline

Verschiedene Studien zeigen, dass die Verwendung von Wissen aus Wikipedia bei der automatischen Segmentierung von Anfragen die Genauigkeit steigert. Dabei werden meist Titel von Wikipedia-Artikeln benutzt, um zusammengehörige Worte in einer Anfrage zu identifizieren. Der Vorteil des Einsatzes von Wikipedia wird auch bei den beiden Verfahren von Hagen et al. deutlich. So erreicht der Wikipedia-basierte Ansatz [HPSB11] bei der Evaluierung bessere Performance-Werte als das naive Verfahren [HPSB10] ohne Wissen aus Wikipedia.

Noch nie wurden jedoch nur die Titel von Wikipedia-Einträgen in einer Anfrage segmentiert – eine Lücke, welche die neu entwickelte Wikipedia-Titel-Baseline schließt.

Konzept

Die Idee der Wikipedia-Titel-Baseline (WT-Baseline) ist es, in einer Anfrage q nur die enthaltenen Titel $t_1 \dots t_m$ von Wikipedia-Artikeln (Wikipedia-Titel) zu segmentieren. Dabei werden Wikipedia-Titel t_i mit mindestens zwei Worten $|t_i| \geq 2$ einbezogen, da diese von einer Segmentierung profitieren. Die Wikipedia-Titel werden anhand einer Liste der Titel von Wikipedia-Einträgen, die aus einem Dump der Englischen Wikipedia extrahiert wurden, identifiziert. Worte, die hingegen nicht zu einem Wikipedia-Titel in der Anfrage gehören, bleiben als triviale Segmente unsegmentiert.

So ist beispielsweise in der Anfrage q = `san francisco visitor tourism statistics` der einzige relevante Wikipedia-Titel t_1 = `san francisco`. Daher segmentiert die WT-Baseline die Anfrage q wie folgt: `"san francisco" "visitor" "tourism" "statistics"`.

Ein Spezialfall tritt auf, wenn in der Anfrage mehrere Wikipedia-Titel enthalten sind, die sich überlappen. Dies kann dabei an mehreren Stellen auftreten, wie in Abbildung 5.1 dargestellt. Die waagerechten Linien unter den Worten w_i symbolisieren die Wikipedia-Titel in der Anfrage.

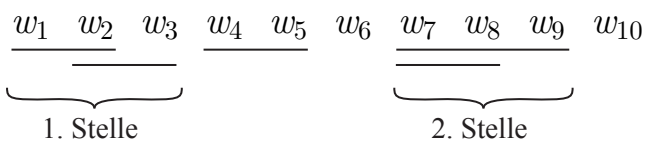

Abbildung 5.1: Wikipedia-Titel in einer Anfrage

Die Stellen überlappender Wikipedia-Titel, im Beispiel in Abbildung 5.1 w_1 w_2 w_3 und w_7 w_8 w_9, werden jeweils einzeln mit dem Wikipedia-basierten Ansatz von Hagen et al. [HPSB11] segmentiert. Disjunkte Wikipedia-Titel ohne Überlappung, wie w_4 w_5 im Beispiel, behandelt die WT-Baseline als ein Segment. Die übrigen Worte, die keinem Wikipedia-Titel angehören, bleiben als 1-Wort-Segmente unsegmentiert. Dies trifft im Beispiel auf die beiden Worte w_6 und w_{10} zu.

Im Testset des WQSC beinhalten rund 18 % der Anfragen überlappende Wikipedia-Titel. Ein Beispiel dafür ist die Anfrage q = `how much costs new york times`, welche die relevanten Wikipedia-Titel t_1 = `new york times` und t_2 = `new york` enthält. Die Vereinigungsmenge der beiden überlappenden Wikipedia-Titel $t_1 \cup t_2$ = `new york times` wird mit dem Wikipedia-basierten Verfahren [HPSB11] segmentiert. Wie im Beispiel in Kapitel 2.2 ist das Ergebnis dieser Segmentierung `"new york times"`. Folglich gibt die WT-Baseline die Segmentierung `"how" "much" "costs" "new york times"` als Lösung aus.

Algorithmus Wikipedia-Titel-Baseline

Input: Anfrage q
Output: Segmentierung S der Anfrage q
1: Ermittelung der Wikipedia-Titel t_i in q
2: **for all** Stellen überlappender Wikipedia-Titel **do**
3: Segmentierung mit Wikipedia-basierten Ansatz
4: **end for**
5: **for all** Disjunkte Wikipedia-Titel **do**
6: Segmentierung der Wikipedia-Titel
7: **end for**
8: **output**

Abbildung 5.2: Konzept der Wikipedia-Titel-Baseline

In Abbildung 5.2 ist die Arbeitsweise der Wikipedia-Titel-Baseline für eine Anfrage q nochmals veranschaulicht. Dabei ermittelt die WT-Baseline zunächst die Wikipedia-Titel t_i und ihre Überlappungen. Alle Stellen überlappender Wikipedia-Titel werden jeweils einzeln mit dem Wikipedia-basierten Verfahren von Hagen et al. [HPSB11] segmentiert. Disjunkte Wikipedia-Titel segmentiert die WT-Baseline jeweils als ein Segment. Die übrigen Worte, die nicht Teil eines Wikipedia-Titels sind, bleiben unsegmentiert.

Bewertung der Genauigkeit

Die Evaluierung der Segmentierungsverfahren in Kapitel 4 zeigte, dass der Wikipedia-basierte Ansatz von Hagen et al. [HPSB11] in den relevanten Evaluierungsmaßen die beste Performance erreicht. Daher vergleichen wir die WT-Baseline in diesem Abschnitt lediglich mit dem Wikipedia-basierten Verfahren. Zudem untersuchen wir mithilfe des etablierten *Student´s t-Test* [Hul93], ob die Unterschiede in den Performance-Werten der beiden Verfahren statistisch signifikant sind. In diesem Zusammenhang wird der sogenannte p-Wert ermittelt, der als Indikator für die statistische Signifikanz dient. Für die Berechnung des p-Werts verwenden wir die Python-Bibliothek *scipy.stats*[1]. Eine statistische Signifikanz der Werte-Differenz liegt vor, wenn $p \leq \frac{\alpha}{2}$ oder $p \geq (1 - \frac{\alpha}{2})$, wobei α das vorgegebene Signifikanzniveau ist. Wie in vielen anderen Studien wählen wir das Signifikanzniveau mit $\alpha = 0,05$. Demzufolge ist die Differenz der Performance-Werte für $p \leq 0,025$ oder $p \geq 0,975$ statistisch signifikant. In den folgenden Tabellen sind signifikante p-Werte fett gedruckt.

[1] http://docs.scipy.org/doc/scipy/reference/stats.html (Zugegriffen am 27.09.2011)

Tabelle 5.1 zeigt zunächst die Ergebnisse der Evaluierung auf dem Enriched Bergsma-Wang-Corpus (EBWC). Es wird deutlich, dass der Wikipedia-basierte Ansatz [HPSB11] auf dem EBWC, welcher nur aus Nominalanfragen besteht, statistisch signifikant bessere Performance-Werte erzielt als die WT-Baseline. Aus der Analyse des Webis Query Segmentation Corpus (WQSC) in Kapitel 3 ging hervor, dass Nominalanfragen tendenziell eher kurze Anfragen sind. Hingegen sind Nicht-Nominalanfragen typischerweise lange Anfragen, welche den Informationsbedarf des Nutzers unklar formulieren. Gerade solche Anfragen könnten von einer Segmentierung mit der WT-Baseline profitieren.

Tabelle 5.1: Evaluierung und Signifikanztest für [HPSB11] und WT-Baseline auf dem EBWC (492 Anfragen)

Maß		**Algorithmus**		**Signifikanz**
		[HPSB11]	WT-Baseline	p-Wert
cat-norm	query	**0,691**	0,567	**0,000**
	segPrec	**0,762**	0,670	**0,000**
	segRec	**0,757**	0,710	**0,003**
	segF	**0,759**	0,689	**0,000**
	break	**0,838**	0,788	**0,000**
corpus	query	**0,587**	0,427	**0,000**
	segPrec	**0,701**	0,586	**0,000**
	segRec	**0,696**	0,640	**0,000**
	segF	**0,699**	0,612	**0,000**
	break	**0,837**	0,770	**0,000**
newbreak		**0,899**	0,858	**0,000**

Zur Untersuchung dieser Hypothese evaluieren wir die WT-Baseline und das Wikipedia-basierte Verfahren [HPSB11] auf den Nominalanfragen und den Nicht-Nominalanfragen des WQSC-5k, des Trainingssets des WQSC mit 4 850 Anfragen, separat.

Die Ergebnisse der Bewertung auf den 2 142 Nominalanfragen des WQSC-5k sind in Tabelle 5.2 dargestellt. Es zeigt sich, dass die WT-Baseline und das Wikipedia-basierte Verfahren teilweise ähnliche Performance-Werte erzielen. Im Evaluierungsmaß category normalized (cat-norm) ist lediglich ein Wert statistisch signifikant. Dabei erreicht die WT-Baseline einen besseren Segment Recall mit $segRec = 0,758$. Die Ergebnisse im Maß corpus sprechen hingegen für das Wikipedia-basierte Verfahren [HPSB11]. Dieses segmentiert 58 % der Anfragen korrekt, wobei die WT-Baseline lediglich 49,7 % richtig segmentiert.

Tabelle 5.2: Evaluierung und Signifikanztest für [HPSB11] und WT-Baseline auf den Nominalanfragen des WQSC-5k (2 142 Anfragen)

Maß		**Algorithmus**		**Signifikanz**
		[HPSB11]	WT-Baseline	p-Wert
cat-norm	query	**0,680**	0,678	0,819
	segPrec	**0,738**	0,737	0,863
	segRec	0,728	**0,758**	**0,000**
	segF	0,733	**0,747**	0,077
	break	0,790	**0,794**	0,486
corpus	query	**0,580**	0,497	**0,000**
	segPrec	**0,669**	0,621	**0,000**
	segRec	0,658	**0,665**	0,364
	segF	**0,663**	0,642	**0,003**
	break	**0,777**	0,750	**0,000**
newbreak		0,875	**0,878**	0,448

Die Differenz der Werte im Maß newbreak ist nicht statistisch signifikant. Dennoch fällt auf, dass die Rangfolge der Verfahren im Vergleich zur Break Accuracy bei corpus – in der Zeile darüber – wechselt, obwohl beide Werte auf Break-Ebene evaluieren. Eine Erklärung dafür ist das unterschiedliche Verhalten der Maße bei einer (5,5)-Entscheidung. Das Maß corpus fügt in einem solchen Zweifelsfall einen Break in der Referenzsegmentierung ein. Somit muss die vorhergesagte Segmentierung für eine gute Bewertung an dieser Stelle auch trennen. Bei newbreak ist es hingegen egal, ob die vorhergesagte Segmentierung trennt oder nicht, weil der Performance-Wert direkt aus der Anzahl der Stimmen der Annotatoren je Trennstelle generiert wird.

In Tabelle 5.3 sind die Resultate der Evaluierung auf den 2 708 Nicht-Nominalanfragen des WQSC-5k aufgelistet. Wie erwartet zeigt sich, dass die WT-Baseline auf Nicht-Nominalanfragen deutlich besser segmentiert als das Wikipedia-basierte Verfahren [HPSB11]. So erreicht die WT-Baseline im Maß corpus mit 49,2 % fast die doppelte Genauigkeit auf Anfrage-Ebene im Vergleich zum Wikipedia-basierten Ansatz [HPSB11]. Diese Ergebnisse untermauern die Hypothese, dass die WT-Baseline besonders auf Nicht-Nominalanfragen gut funktioniert.

Tabelle 5.3: Evaluierung und Signifikanztest für [HPSB11] und WT-Baseline auf den Nicht-Nominalanfragen des WQSC-5k (2 708 Anfragen)

Maß		**Algorithmus**		**Signifikanz**
		[HPSB11]	WT-Baseline	p-Wert
cat-norm	query	0,313	**0,643**	**0,000**
	segPrec	0,552	**0,759**	**0,000**
	segRec	0,497	**0,766**	**0,000**
	segF	0,523	**0,762**	**0,000**
	break	0,659	**0,805**	**0,000**
corpus	query	0,268	**0,492**	**0,000**
	segPrec	0,529	**0,688**	**0,000**
	segRec	0,475	**0,723**	**0,000**
	segF	0,500	**0,705**	**0,000**
	break	0,666	**0,788**	**0,000**
newbreak		0,771	**0,897**	**0,000**

5.2 Das hybride Verfahren

Aus der Evaluierung im vorherigen Abschnitt geht hervor, dass die WT-Baseline dem Wikipedia-basierten Ansatz von Hagen et al. [HPSB11] und somit auch den anderen Algorithmen auf Nicht-Nominalanfragen überlegen ist. Hingegen übertrifft das Wikipedia-basierte Verfahren [HPSB11] alle anderen Verfahren einschließlich der WT-Baseline bei der Segmentierung von Nominalanfragen. Die Stärken der beiden Algorithmen macht sich das hybride Verfahren zunutze.

Konzept

Die Idee des hybriden Verfahrens ist es, den Wikipedia-basierten Ansatz von Hagen et al. [HPSB11] mit der WT-Baseline zu kombinieren und je nach Art der Anfrage das Segmentierungsverfahren auszuwählen. Abbildung 5.3 veranschaulicht die Arbeitsweise des hybriden Verfahrens. Dabei wird zunächst festgestellt, ob es sich bei der Anfrage q um eine Nominalanfrage handelt oder nicht. Hierfür verwenden wir, wie in Kapitel 3, den POS-Tagger *QTag*. Nominalanfragen werden mit dem Wikipedia-basierten Ansatz [HPSB11] segmentiert. Hingegen verwendet das hybride Verfahren für die Segmentierung der Nicht-Nominalanfragen die WT-Baseline.

Anfrage q = Nominalanfrage	
JA	NEIN
Wikipedia-basiertes Verfahren	Wikipedia-Titel-Baseline

Abbildung 5.3: Konzept des hybriden Verfahrens

Evaluierung der Genauigkeit

Bei der Bewertung der Genauigkeit des hybriden Verfahrens verzichten wir auf eine Evaluierung auf dem Enriched Bergsma-Wang-Corpus (EBWC). Der EBWC besteht lediglich aus Nominalanfragen. Diese Anfragen segmentiert das hybride Verfahren mit dem Wikipedia-basierten Ansatz von Hagen et al. [HPSB11]. Daher stimmt die Genauigkeit des hybriden Verfahrens auf dem EBWC mit der des Wikipedia-basierten Ansatz [HPSB11] übereinstimmt.

Die Idee des hybriden Verfahrens entstand durch die Evaluierung der Teilverfahren auf dem WQSC-5k, dem Trainigsset des WQSC mit 4 850 Anfragen. Um eine repräsentative Analyse der Performance sicherzustellen, evaluieren wir das hybride Verfahren daher auf den 48 587 Anfragen des Testsets des WQSC. Dabei vergleichen wir das Verfahren mit seinen beiden Teilmethoden und untersuchen die statistische Signifikanz der Performance-Unterschiede.

Tabelle 5.4: Signifikanztest für [HPSB11] und hybrides Verfahren auf dem WQSC-50k (48 587 Anfragen)

Maß		**Algorithmus**		**Signifikanz**
		[HPSB11]	hybrid	p-Wert
cat-norm	query	0,490	**0,664**	**0,000**
	segPrec	0,644	**0,754**	**0,000**
	segRec	0,608	**0,754**	**0,000**
	segF	0,626	**0,754**	**0,000**
	break	0,720	**0,798**	**0,000**
corpus	query	0,412	**0,535**	**0,000**
	segPrec	0,606	**0,693**	**0,000**
	segRec	0,565	**0,703**	**0,000**
	segF	0,585	**0,698**	**0,000**
	break	0,717	**0,785**	**0,000**
newbreak		0,810	**0,885**	**0,000**

Zunächst stellen wir die Genauigkeit des hybriden Verfahrens mit der des Wikipedia-basierten Verfahrens [HPSB11] in Tabelle 5.4 gegenüber. Es zeigt sich, dass das hybride Verfahren in allen Maßen deutlich bessere Werte erzielt als der Wikipedia-basierte Ansatz [HPSB11]. So erreicht das hybride Verfahren unter Anwendung des corpus-Maßes beispielsweise eine Query Accuracy von 53,5 % und übertrifft somit das Wikipedia-basierte Verfahren, welches nur 41,2 % der Anfragen korrekt segmentiert.

Tabelle 5.5: Signifikanztest für WT-Baseline und hybrides Verfahren auf dem WQSC-50k (48 587 Anfragen)

Maß		**Algorithmus**		**Signifikanz**
		WT-Baseline	hybrid	p-Wert
cat-norm	query	0,655	**0,664**	**0,000**
	segPrec	0,749	**0,754**	**0,000**
	segRec	**0,762**	0,754	**0,000**
	segF	**0,756**	0,754	0,234
	break	**0,799**	0,798	0,486
corpus	query	0,502	**0,535**	**0,000**
	segPrec	0,673	**0,693**	**0,000**
	segRec	**0,708**	0,703	**0,000**
	segF	0,690	**0,698**	**0,000**
	break	0,778	**0,785**	**0,000**
newbreak		**0,891**	0,885	**0,000**

Die Ergebnisse des Vergleichs zwischen dem hybriden Verfahren und der WT-Baseline sind in Tabelle 5.5 dargestellt. Obwohl sich die Performance-Werte der beiden Verfahren sehr ähneln, gibt es aufgrund des großen Stichprobenumfangs signifikante Unterschiede. So hat das hybride Verfahren sowohl im Evaluierungsmaß category normalized (cat-norm) als auch bei corpus eine signifikant bessere Query Accuracy.

Die Tabellen 5.4 und 5.5 zeigen, dass das hybride Verfahren bei der Segmentierung von Anfragen eine bessere Genauigkeit erreicht als seine Teilverfahren. Um die Bewertung der Performance abzurunden, wird im folgenden Abschnitt die Laufzeit und der Speicherbedarf des hybriden Verfahrens untersucht.

Analyse von Laufzeit und Speicherbedarf

Im Rahmen der Bewertung der Laufzeit vergleichen wir die Performance des hybriden Verfahrens mit der seiner Teilverfahren. Die Ergebnisse der Messung auf dem WQSC-50k, dem Testset des WQSC mit 48 587 Anfragen, sind in Tabelle 5.6 aufgelistet.

Tabelle 5.6: Zeitmessung auf dem WQSC-50k (48 587 Anfragen)

Laufzeit	**Algorithmus**		
	[HPSB11]	WT-Baseline	hybrid
Anfragen/s	3 358	3 797	2 918

Es zeigt sich, dass die WT-Baseline mit 3 797 Anfragen pro Sekunde den besten Datendurchsatz erreicht. Darauf folgt das Wikipedia-basierte Verfahren [HPSB11], welches 3 358 Anfragen pro Sekunde segmentiert. Das hybride Verfahren hat mit 2 918 Anfragen je Sekunde den vergleichbar geringsten Datendurchsatz und somit die längste Laufzeit. Diese ist dennoch der Größenordnung nach mit der Performance der beiden anderen Verfahren vergleichbar.

Neben der Laufzeit ist der Speicherbedarf ein weiterer wichtiger Aspekt bei der Bewertung der Performance. Das Wikipedia-basierte Verfahren von Hagen et al. [HPSB11] benötigt 12 GB Hauptspeicher für eine Hashtabelle, welche die normalisierten Häufigkeiten der Wort-N-Gramme bereithält. Sowohl die WT-Baseline als auch das hybride Verfahren nutzen den Ansatz von Hagen et al. als Teilverfahren und verwenden dabei weniger Wort-N-Gramme zur Segmentierung. Bei der WT-Baseline werden lediglich die Häufigkeiten der Wort-N-Gramme benötigt, die in Wikipedia-Titeln vorkommen. Das hybride Verfahren benötigt zudem die Häufigkeiten aller Wort-N-Gramme, welche nur aus Nomen, Adjektiven und Artikeln bestehen. Tabelle 5.7 zeigt die Nutzung der Wort-N-Gramme der drei Verfahren bei der Segmentierung der 48 587 Anfragen des WQSC-50k. Der Wikipedia-basierte Ansatz von Hagen et al. [HPSB11] fragt 1 056 452 Mal die Auftrittshäufigkeiten von Wort-N-Grammen in der Hashtabelle ab. Lediglich 164 522 dieser Zugriffe beziehen

Tabelle 5.7: Nutzung Wort-N-Gramme auf dem WQSC-50k (48 587 Anfragen)

Zugriffe auf Hashtabelle	**Algorithmus**		
	[HPSB11]	WT-Baseline	hybrid
Verschiedene Wort-N-Gramme	164 522	18 894	63 549
Gesamt	1 056 452	67 727	162 954

sich dabei auf verschiedene Wort-N-Gramme. Deutlich weniger Zugriffe benötigen die beiden anderen Verfahren. Die WT-Baseline nutzt nur 11,5 % (18 894 von 164 522) und das hybride Verfahren 38,6 % (63 549 von 164 522) der Wort-N-Gramme des Wikipedia-basierten Ansatzes. Anhand der Messung auf dem WQSC-50k versuchen wir den Speicherbedarf der Verfahren abzuschätzen. So benötigt die WT-Baseline etwa 1,4 GB (11,5 % von 12 GB) und das hybride Verfahren circa 4,6 GB (38,6 % von 12 GB) Hauptspeicher für die Hashtabelle mit den normalisierten Häufigkeiten der Wort-N-Gramme.

Durch die Kombination der Vorteile der WT-Baseline und des Wikipedia-basierten Ansatzes von Hagen et al. [HPSB11] erreicht das neue hybride Verfahren bei der Segmentierung von Anfragen eine höhere Genauigkeit als die Teilverfahren. Im Vergleich zum Wikipedia-basierten Ansatz benötigt das hybride Verfahren weniger als die Hälfte des Speichers und erreicht eine ähnlich gute Laufzeit. Somit ist das hybride Verfahren zur on-the-fly Segmentierung von Suchanfragen geeignet.

Kapitel 6

Zusammenfassung und Ausblick

Die vorliegende Arbeit beschäftigte sich zum einen mit der Evaluierung von automatisch generierten Anfragesegmentierungen. Dabei wurde der neue Evaluierungskorpus Webis Query Segmentation Corpus (WQSC) erstmals genauer untersucht. Neben der Konstruktion des Korpus stand vor allem die Analyse der enthaltenen Anfragen und Segmentierungen anhand ausgewählter Kriterien im Mittelpunkt. Es wurde deutlich, dass sich die Annotatoren beim Segmentieren der Anfragen des WQSC in etwa zu gleichen Teilen sicher und unsicher waren. Dabei präferierten sie vor allem kurze Segmente.

Die Analyse des Korpus zeigte zudem, dass die bisherigen Evaluierungsmaße aufgrund der vermehrten Segmentierungen im WQSC eine unausgewogene Bewertung der Genauigkeit zur Folge hätte. Daher wurden in dieser Arbeit Maße entwickelt, die auf Evaluierungskorpora mit mehr Stimmen zugeschnitten sind. Der Vorteil der neuen Maße ist es, dass bei der Evaluierung die Verteilung der Stimmen der Annotatoren im Korpus einbezogen wird und somit eine fairere Performance-Bewertung sichergestellt ist.

Eine umfangreiche Evaluierung von acht Segmentierungsverfahren auf dem Enriched Bergsma-Wang-Corpus (EBWC) verdeutlichte den Unterschied zwischen den bisherigen und den neuen Evaluierungsmaßen. Dabei wurde die Bewertung der Genauigkeit mit steigender Komplexität der Maße nicht nur strenger, sondern das Ranking der Verfahren änderte sich gleichzeitig. In den fairsten Evaluierungsmaßen erreichte das Wikipedia-basierte Verfahren von Hagen et al. [HPSB11] die beste Treffergenauigkeit auf dem EBWC. Mit der Evaluierung auf dem Trainingsset des WQSC stellte sich heraus, dass das Wikipedia-basierte Verfahren besonders bei der Segmentierung von Nominalanfragen eine hohe Trefferquote erzielt.

Im zweiten Teil dieser Arbeit wurden zwei neue Verfahren zur Segmentierung von Anfragen entwickelt. Die Wikipedia-Titel-Baseline (WT-Baseline) segmentiert lediglich die Titel von Wikipedia-Einträgen in einer Anfrage und

erreicht besonders bei Nicht-Nominalanfragen eine hohe Genauigkeit. Das hybride Verfahren kombiniert die WT-Baseline mit dem Wikipedia-basierten Ansatz von Hagen et al. [HPSB11] und wählt je nach Anfrageart aus den beiden Verfahren aus. Die Analyse der Performance auf dem Testset des WQSC zeigte, dass das hybride Verfahren bei vergleichbarer Laufzeit mit einer höheren Trefferquote segmentiert als dessen Teilverfahren. Weiterhin benötigt das hybride Verfahren nur etwa 40 % des Speichers im Vergleich zum Wikipedia-basierten Ansatz von Hagen et al. [HPSB11].

Um das hybride Verfahren in Zukunft weiter zu optimieren, könnte die Anfrage vor der Segmentierung noch genauer untersucht werden. Bisher unterscheiden wir zwischen Nominalanfragen und Nicht-Nominalanfragen. Dabei werden in Nicht-Nominalanfragen lediglich Titel von Wikipedia-Einträgen (Wikipedia-Titel) segmentiert. Kommt jedoch kein Wikipedia-Titel in der Anfrage vor, so bleibt diese unsegmentiert. Eine denkbare Weiterentwicklung wäre es, in Nicht-Nominalanfragen ohne Wikipedia-Titel die Nominal-Anteile zu identifizieren und diese mit dem Wikipedia-basierten Ansatz zu segmentieren. Im Rahmen der Evaluierung muss anschließend festgestellt werden, ob dies tatsächlich zur Steigerung der Genauigkeit führt.

Ein weiterer Ansatzpunkt für die Verbesserung des hybriden Verfahrens ist die Klassifizierung der Anfrageart und daher das POS-Tagging der enthaltenen Worte. Bisher verwenden wir im hybriden Verfahren den POS-Tagger QTag, welcher Texte und Sätze nachweislich gut taggt. Beim Taggen von Anfragen entstehen jedoch Fehler, da keine Kontextinformationen wie z. B. in Sätzen verfügbar sind. Mit einem POS-Tagger, der auf kurze Wortsequenzen spezialisiert ist, könnten die Worte und somit auch die Anfrageart mit höherer Genauigkeit kategorisiert werden. Daher ist anzunehmen, dass ein solcher POS-Tagger die Performance des hybriden Verfahrens hinsichtlich der Genauigkeit der Segmentierungen verbessern kann. Der zweite Aspekt der Performance, die Laufzeit, kann durch einen schnelleren POS-Tagger optimiert werden.

Neben den Optimierungen des hybriden Verfahrens bleibt außerdem die Frage zu klären, in welchem Maß die Segmentierung von Anfragen die Retrieval Performance verbessern kann. Ein Ansatzpunkt für diese Problematik ist unter anderem, herauszufinden ob verschiedene Segmentierungen einer Anfrage auch unterschiedliche Suchergebnisse liefern. Dabei muss zudem die Relevanz der Suchergebnisse bezüglich der Anfrage analysiert werden.

Abbildungsverzeichnis

3.1 Verteilung der Votes (v_1, v_2) im WQSC 17
3.2 Kriterien zur Analyse des WQSC 18
3.3 WQSC: Anfragelänge und Anfrageart 19
3.4 WQSC: Anfragelänge und Topvote 19
3.5 WQSC: Anfragelänge und Kategorie 20

5.1 Wikipedia-Titel in einer Anfrage 37
5.2 Konzept der Wikipedia-Titel-Baseline 38
5.3 Konzept des hybriden Verfahrens 42

Tabellenverzeichnis

1.1 Segmentierungen für `new york times square dance` 2

2.1 Mögliche Segmentierungen für `new york times` 4
2.2 Berechnung der Scores für `new york times` in Millionen 11

3.1 Segmentlängen im WQSC 16
3.2 Anfragelängen im WQSC 16
3.3 WQSC: Anfrageart und Kategorie 20
3.4 WQSC: Topvote und Kategorie 21
3.5 WQSC: Topvote und Anfrageart 21

4.1 Segmentierungen für S_{pred} und S_{ref} 24
4.2 Evaluierung mit S_{pred} = `"los angeles" "times"` 26
4.3 Segmentierungen für q = `new york city news` 29
4.4 Aufgesammelte Votes ($Break_i$ | $NonBreak_i$) je Trennstelle b_i in q = `new york city news` 29
4.5 Evaluierung auf dem EBWC (492 Anfragen) 32
4.6 Evaluierung auf dem WQSC-5k (4 850 Anfragen) 33

5.1 Evaluierung und Signifikanztest für [HPSB11] und WT-Baseline auf dem EBWC (492 Anfragen) 39
5.2 Evaluierung und Signifikanztest für [HPSB11] und WT-Baseline auf den Nominalanfragen des WQSC-5k (2 142 Anfragen) 40
5.3 Evaluierung und Signifikanztest für [HPSB11] und WT-Baseline auf den Nicht-Nominalanfragen des WQSC-5k (2 708 Anfragen) 41
5.4 Signifikanztest für [HPSB11] und hybrides Verfahren auf dem WQSC-50k (48 587 Anfragen) 42
5.5 Signifikanztest für WT-Baseline und hybrides Verfahren auf dem WQSC-50k (48 587 Anfragen) 43
5.6 Zeitmessung auf dem WQSC-50k (48 587 Anfragen) 44
5.7 Nutzung Wort-N-Gramme auf dem WQSC-50k (48 587 Anfragen) 44

Literaturverzeichnis

[AGHI09] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson und Samuel Ieong. Diversifying search results. In Ricardo A. Baeza-Yates, Paolo Boldi, Berthier A. Ribeiro-Neto und Berkant Barla Cambazoglu (Hrsg.), *Proceedings of the Second International Conference on Web Search and Web Data Mining, WSDM 2009, Barcelona, Spain, February 9-11, 2009*, Seiten 5–14. ACM, 2009.

[BCS09] Michael Bendersky, W. Bruce Croft und David A. Smith. Two-stage query segmentation for information retrieval. In James Allan, Javed A. Aslam, Mark Sanderson, ChengXiang Zhai und Justin Zobel (Hrsg.), *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, Seiten 810–811. ACM, 2009.

[BCS10] Michael Bendersky, W. Bruce Croft und David A. Smith. Structural annotation of search queries using pseudo-relevance feedback. In Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson und Aijun An (Hrsg.), *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, Seiten 1537–1540. ACM, 2010.

[BCS11] Michael Bendersky, W. Bruce Croft und David A. Smith. Joint annotation of search queries. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, Seiten 102–111. The Association for Computer Linguistics, 2011.

[BF06] Thorsten Brants und Alex Franz. Web 1T 5-gram Version 1. Linguistic Data Consortium, Philadelphia, 2006.

[BGAG10] David J. Brenes, Daniel Gayo-Avello und Rodrigo Garcia. On the fly query entity decomposition using snippets. In *Proceedings of the First Spanish Conference on Information Retrieval, CERI 2010, June 15-16, 2010, Madrid, Spain*, 2010.

[BW07] Shane Bergsma und Qin Iris Wang. Learning noun phrase query segmentation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2007, June 28-30, 2007, Prague, Czech Republic*, Seiten 819–826. ACL, 2007.

[GXLC08] Jiafeng Guo, Gu Xu, Hang Li und Xueqi Cheng. A unified and discriminative model for query refinement. In Sung-Hyon Myaeng, Douglas W. Oard, Fabrizio Sebastiani, Tat-Seng Chua und Mun-Kew Leong (Hrsg.), *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, Seiten 379–386. ACM, 2008.

[HGM+10] Jian Huang, Jianfeng Gao, Jiangbo Miao, Xiaolong Li, Kuansan Wang, Fritz Behr und C. Lee Giles. Exploring web scale language models for search query processing. In Michael Rappa, Paul Jones, Juliana Freire und Soumen Chakrabarti (Hrsg.), *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, Seiten 451–460. ACM, 2010.

[HPSB10] Matthias Hagen, Martin Potthast, Benno Stein und Christof Bräutigam. The power of naïve query segmentation. In Hsin-Hsi Chen, Efthimis N. Efthimiadis, Jaques Savoy, Fabio Crestani und Stéphane Marchand-Maillet (Hrsg.), *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, Seiten 797–798. ACM, Juli 2010.

[HPSB11] Matthias Hagen, Martin Potthast, Benno Stein und Christof Bräutigam. Query segmentation revisited. In Sadagopan Srinivasan, Krithi Ramamritham, Arun Kumar, M. P. Ravindra, Elisa Bertino und Ravi Kumar (Hrsg.), *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011*, Seiten 97–106. ACM, 2011.

[Hul93] David Hull. Using statistical testing in the evaluation of retrieval experiments. In Robert Korfhage, Edie M. Rasmussen und Peter

Willett (Hrsg.), *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR 1993, Pittsburgh, PA, USA, June 27 - July 1, 1993*, Seiten 329–338. ACM, 1993.

[JRMG06] Rosie Jones, Benjamin Rey, Omid Madani und Wiley Greiner. Generating query substitutions. In Les Carr, David De Roure, Arun Iyengar, Carole A. Goble und Michael Dahlin (Hrsg.), *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006*, Seiten 387–396. ACM, 2006.

[KGA+10] Julia Kiseleva, Qi Guo, Eugene Agichtein, Daniel Billsus und Wei Chai. Unsupervised query segmentation using click data: Preliminary results. In Michael Rappa, Paul Jones, Juliana Freire und Soumen Chakrabarti (Hrsg.), *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, Seiten 1131–1132. ACM, 2010.

[LHZW11] Yanen Li, Bo-June (Paul) Hsu, ChengXiang Zhai und Kuansan Wang. Unsupervised query segmentation using clickthrough for information retrieval. In Wei-Ying Ma, Jian-Yun Nie, Ricardo A. Baeza-Yates, Tat-Seng Chua und W. Bruce Croft (Hrsg.), *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, Seiten 285–294. ACM, 2011.

[MRG+11] Nikita Mishra, Rishiraj Saha Roy, Niloy Ganguly, Srivatsan Laxman und Monojit Choudhury. Unsupervised query segmentation using only query logs. In Sadagopan Srinivasan, Krithi Ramamritham, Arun Kumar, M. P. Ravindra, Elisa Bertino und Ravi Kumar (Hrsg.), *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011 (Companion Volume)*, Seiten 91–92. ACM, 2011.

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

[PCT06] Greg Pass, Abdur Chowdhury und Cayley Torgeson. A picture of search. In Xiaohua Jia (Hrsg.), *Proceedings of the 1st International Conference on Scalable Information Systems, Infoscale 2006, Hong*

Kong, May 30-June 1, 2006, Band 152 aus *ACM International Conference Proceeding Series*, Paper 1. ACM, 2006.

[RMB03] Knut Magne Risvik, Tomasz Mikolajewski und Peter Boros. Query segmentation for web search. In *Proceedings of the 12th International World Wide Web Conference, WWW 2003, Budapest, Hungary, May 20-24, 2003*, 2003.

[TM98] Dan Tufis und Oliver Mason. Tagging romanian texts: a case study for qtag, a language independent probabilistic tagger. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*, Seiten 589–596. Citeseer, 1998.

[TP08] Bin Tan und Fuchun Peng. Unsupervised query segmentation using generative language models and Wikipedia. In Jinpeng Huai, Robin Chen, Hsiao-Wuen Hon, Yunhao Liu, Wei-Ying Ma, Andrew Tomkins und Xiaodong Zhang (Hrsg.), *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*, Seiten 347–356. ACM, 2008.

[WM07] Ryen W. White und Dan Morris. Investigating the querying and browsing behavior of advanced search engine users. In Wessel Kraaij, Arjen P. de Vries, Charles L. A. Clarke, Norbert Fuhr und Noriko Kando (Hrsg.), *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2007, Amsterdam, The Netherlands, July 23-27, 2007*, Seiten 255–262. ACM, 2007.

[YS09] Xiaohui Yu und Huxia Shi. Query segmentation using conditional random fields. In M. Tamer Özsu, Yi Chen und Lei Chen (Hrsg.), *Proceedings of the First International Workshop on Keyword Search on Structured Data, KEYS 2009, Providence, Rhode Island, USA, June 28, 2009*, Seiten 21–26. ACM, 2009.

[ZCBY10] Hugo Zaragoza, Berkant Barla Cambazoglu und Ricardo A. Baeza-Yates. Web search solved?: All result rankings the same? In Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson und Aijun An (Hrsg.), *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, Seiten 529–538. ACM, 2010.

[ZSH+09] Chao Zhang, Nan Sun, Xia Hu, Tingzhu Huang und Tat-Seng Chua. Query segmentation based on eigenspace similarity. In *ACL*

2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore, Short Papers, Seiten 185–188. The Association for Computer Linguistics, 2009.

[ZZY+08] Guo-Qing Zhang, Guo-Qiang Zhang, Qing-Feng Yang, Su-Qi Cheng und Tao Zhou. Evolution of the internet and its cores. *New Journal of Physics*, 2008.