Bauhaus-Universität Weimar
Fakultät Medien
Studiengang Mediensysteme

Ein neuer Ansatz für Clusterlabeling: Was war die Suchanfrage?

Bachelorarbeit

Maximilian Michel

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

Datum der Abgabe: 30. März 2012

Erklärung

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

Weimar, den 30. März 2012

.....
Maximilian Michel

Zusammenfassung

In dieser Arbeit stellen wir ein neues Clusterlabeling-Verfahren vor. Unsere Idee ist es, Clusterlabel und Cluster-Dokumente genauso zu verknüpfen, wie eine Suchmaschine eine Suchanfrage mit den Ergebnisdokumenten verknüpft. Hierfür entwickeln wir zunächst ein Verfahren, das aus einer gegebenen Dokumentmenge eine Suchanfrage generieren kann. Dabei wird nur die Schnittstelle der Suchmaschine benutzt. Das heißt, dass näheres Wissen zu Retrievalmodell oder indiziertem Dokument-Korpus der Suchmaschine nicht gebraucht wird.

Aufbauend auf diesem Verfahren entwickeln wir das Clusterlabeling-Verfahren und testen dieses auf einem selbst erstellten Cluster-Korpus. Für die Evaluation nutzen wir einerseits maschinelle Methoden zum Ähnlichkeitsvergleich der generierten Labels mit den Referenzlabels des Korpus und führen außerdem eine Nutzerstudie durch, bei der Probanden eine Auswahl von Labels bewerten. Für den Ähnlichkeitsvergleich verwenden wir sowohl klassische Maße, wie Jaccard-Index, F-Measure und Kosinus-Ähnlickeit, sowie die Explicit Semantic Analysis, ein Verfahren zur semantischen Erweiterung der Kosinus-Ähnlichkeit. Letzteres Verfahren eignet sich gut für den Vergleich kurzer Texte und wurde bisher noch nicht für die Evaluation von Clusterlabels verwendet.

Unser Anfrage-Rekonstruktions-Verfahren schneidet etwa so gut ab, wie das χ^2-Labeling, dem Besten der anderen in der Evaluation genutzten Clusterlabeling-Verfahren. Eine Schwachstelle unseres Verfahrens ist die Länge des Clusterlabels. Diese sind nämlich mit einer Länge von teilweise über acht Wörtern zu lang. Der Nutzerstudie zufolge werden eher Labels mit einer Länge von zwei bis drei Wörtern für den Titel eines Clusters bevorzugt .

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Stellt man eine Anfrage an eine Suchmaschine, liefert diese als Ergebnis eine Menge von Dokumenten. Das erhaltene Suchergebnis stellt dabei eine Teilmenge von Dokumenten aus einem großen Dokumentkorpus (etwa dem Web) dar, die für die Anfrage relevant ist und dem Nutzer als Ergebnisliste präsentiert wird.

Ziel dieser Bachelorarbeit ist es, den Zusammenhang zwischen Suchanfrage und Menge von relevanten Dokumenten aus einer anderen Richtung zu betrachten. Der Ausgangspunkt ist hierbei eine Menge von Dokumenten, die sich beispielsweise inhaltlich ähnlich sind oder sich thematisch unter einer Überschrift zusammenfassen lassen. Die Aufgabe ist es, für diese Dokumente eine passende Suchanfrage zu generieren, deren Suchergebnis möglichst alle Dokumente der ursprünglichen Menge enthält und gleichzeitig möglichst wenig andere Dokumente, die nicht in der Ursprungsmenge enthalten sind. Als Hilfsmittel dienen dabei die Dokumente der Ausgangsmenge und eine Suchmaschine, die als eine Art *Black Box* funktioniert. Auf einen bestimmten Input (eine Suchanfrage) liefert die Blackbox einen bestimmten Output (das passende Suchergebnis), ohne dass näheres Wissen über das Retrieval-Modell oder die indizierte Dokumentmenge vorhanden sein muss. Ein solches Vorgehen ist mit allen aktuellen Suchmaschinen möglich und garantiert, dass die in dieser Arbeit diskutierten Methoden auch auf alle aktuellen Retrieval-Modelle und Dokumentmengen anwendbar sind.

Als Anwendungsgebiet der Suchanfragen-Rekonstruktion wenden wir das entwickelte Verfahren für *Clusterlabeling* an. Dabei beschreibt das *Clustering* den Prozess des Unterteilen von Dokumentmengen in Unterdokumentmengen (sogenannte *Cluster*). Clusterlabeling-Verfahren, wie beispielsweise das Maximum Term Weight Labeling, generieren für jeden Cluster ein Label, welches den Inhalt des Clusters kurz beschreibt. Dabei beziehen sich klassische Clusterlabeling-Verfahren meist nur auf statistische Eigenschaften des Clusterings. Der Ansatz, Clusterlabels aus dem Retrievalkontext zu erstellen, ist hingegen neu. Dazu erweitern wir das Suchanfragen-Rekonstruktions-Verfahren so, dass eine Suchanfrage konstruiert wird, die ein Cluster von anderen gegebenen Clustern abgrenzt. Hierbei müssen Terme für die Suchanfrage gewählt werden, die viele Dokumente des einen Clusters wiederfinden und gleichzeitig möglichst viele Dokumente der anderen Clustern ausschließen.

Die Qualität der auf diesem Wege erstellten Clusterlabels wird im letzten Teil dieser Arbeit vergleichend zu anderen Clusterlabeling-Verfahren untersucht. Dabei werden die erstellten Labels mit den jeweiligen Referenzlabels verglichen. Neben klassischen Methoden zur Evaluierung, wie dem F-Measure oder dem Jaccard-Index, entwickeln wir auch neue Ähnlichkeitsmaße. Die meisten Ähnlichkeitsmaße, wie beispielsweise die traditionell verwendete Kosinus-Ähnlichkeit, liefern die besten Ergebnisse beim Vergleich von langen Texten. Um einen besseren Ähnlichkeitsvergleich für die relativ kurzen Clusterlabels zu ermöglichen, schlagen wir die *Explicit Semantic Analysis* zur Erweiterung der Kosinus-Ähnlichkeit vor. Bisher wurde dieses Verfahren noch nicht für die Evaluierung von Clusterlabels verwendet.

Die Arbeit ist wie folgt gegliedert: Kapitel 2 zeigt den aktuellen Forschungsstand im Gebiet der Suchanfragen-Analyse und des Clusterlabelings. In Kapitel 3 wird der Reverted Index als Werkzeug zur Suchanfragen-Rekonstruktion eingeführt und gezeigt, wie damit Suchanfragen rekonstruiert werden können. In dem darauffolgenden Kapitel 4 zeigen wir, wie das Verfahren für Clusterlabeling angepasst und genutzt werden kann. In Kaptiel 5 wird die Evaluierung des Clusterlabeling-Verfahren beschrieben und die Explicit Semantic Analysis als Erweiterung der klassischen Kosinus-Ähnlichkeit vorgestellt. In Kapitel 6 fassen wir die Ergebnisse der Arbeit zusammen und geben einen Ausblick, wie mit der Suchanfragen-Rekonstruktion zukünftig weiter gearbeitet werden kann.

2 Verwandte Arbeiten

Das Erzeugen von Suchanfragen aus Dokumentmengen wurde bereits von Jordan et al. näher betrachtet [JWG06]. Hierbei wurde ein Ansatz entwickelt, der eine Menge von möglichen Queries mit Hilfe der Entropie der *Sprach-Modelle* der Ursprungsdokumentmenge und des Dokumentkorpus generiert. Diese so genannten *Query Environments* werden genutzt, um zwei Algorithmen des *Blind Relevance Feedbacks* zu evaluieren. Dafür wurden Ursprungsdokumentmengen von Hand zusammengestellt bzw. kategorisiert. Für die Umsetzung dieser Technik ist jedoch ein Zugriff auf den Gesamtkorpus notwendig, da Sprach-Modelle für diesen erstellt werden müssen. Ziel dieser Arbeit ist es jedoch, ohne direkten Zugriff auf den Gesamtdokumentkorpus allein mit Hilfe einer als Black Box funktionierenden Suchmaschine, Suchanfragen zu generieren. Die Techniken von Jordan et al. sind deshalb nicht umsetzbar für unser Ziel.

Auch Bonchi et al. beschäftigen sich mit der Generierung von Suchanfragen [BCDG08]. Der Ausgangspunkt ist hierbei eine Suchanfrage eines Nutzers und die passende Suchergebnisliste von Dokumenten. Mit Hilfe eines Anfragelogs wird im Ansatz von Bonchi et al. die ursprüngliche Suchanfrage in eine Menge von Teilsuchanfragen zerlegt, welche jeweils kleinere Dokumentmengen finden. Aus diesen Teilsuchanfragen werden diejenigen heraus gesucht, die zusammen genommen die Ergebnisliste der ursprünglichen Suchanfrage *abdecken*. Das heißt, dass die Suchanfragen so gewählt sind, dass die Vereinigung ihrer Ergebnislisten der Ursprungsergebnisliste entspricht. Das Zerlegen wird also als ein *Set Cover Problem* betrachtet. In einem anderen Ansatz werden zuerst die Dokumente der Suchergebnisliste in Cluster geteilt und für jedes dieser Cluster Suchanfragen aus einem Anfragelog ausgewählt, die die enthaltenen Dokumente finden. Die erhaltene Menge an Suchanfragen kann dazu genutzt werden, um dem Nutzer Vorschläge anzubieten, seine ursprüngliche Suchanfrage konkreter zu formulieren. Unsere Problemstellung ist ähnlich der von Bonchi et. al [BCDG08]. Der große Unterschied ist jedoch, dass wir mit unserem Verfahren einzelne Suchanfragen rekonstruieren, welche alle Ursprungsdokumente finden. Außerdem kennen wir die ursprüngliche Suchanfrage nicht. Der Ansatz, für eine passende Suchanfrage ein Anfragelog zu analysieren, hat den Vorteil, dass die Suchanfragen von Menschen formuliert wurden, birgt jedoch

die Schwierigkeit, dass nicht unbedingt alle Dokumente mit den Suchanfragen des Logs abgedeckt werden.

Ein weiterer Ansatz, um Suchterme aus einer Dokumentmenge zu erzeugen, wurde von Pickens et al. diskutiert [PCG10]. Im Gegensatz zum klassischen Information Retrieval, bei dem in einem *Inverted Index* einzelne Suchterme auf eine Menge von Dokumenten abgebildet werden, wird hier eine neue Struktur eingeführt: der *Reverted Index*. Dieser bringt umgekehrt Dokumente mit einer Menge von Suchtermen in Verbindung. Um den Reverted Index zu erstellen, werden so genannte *Basisanfragen* an eine Suchmaschine gestellt. Die Dokumente der erhaltenen Ergebnisliste werden als Schlüssel des Reverted Index hinzugefügt und mit den Suchanfragen assoziiert. Jedes Dokument in dem Reverted Index wird also auf eine Menge von Suchanfragen abgebildet, mit denen man das Dokument gut finden kann. Mit Hilfe dieses Index können Terme ermittelt werden, die für *Query Expansion* oder *(Blind) Relevance Feedback* genutzt werden können. Da wir den Reverted Index für die Suchanfragen-Rekonstruktion benutzen, wird die genaue Vorgehensweise zur Erstellung und Nutzung des Reverted Index in Abschnitt 3.1 näher erläutert. Der Reverted Index ist für diese Arbeit ein wichtiges Werkzeug, um Suchterme aus Dokumentmengen zu extrahieren, da dieser allein die Schnittstelle einer Suchmaschine nutzt und keine weiteren Zugriffe auf den Gesamtdokumentkorpus gebraucht werden.

Der Ansatz, nur mit Hilfe einer Suchmaschinen-Schnittstelle Suchanfragen zu formulieren, wird auch von Hagen und Stein verwendet [HS10]. Es handelt sich hierbei um ein Verfahren, um Suchanfragen eines Nutzers zu verbessern. Dabei wird eine Suchanfrage generiert, die möglichst viele Terme der ursprünglichen Suchanfrage enthält und gleichzeitig eine *angemessene* Anzahl von Dokumenten findet: eine so genannte *Maximum Query*. Hierfür werden zwei Algorithmen vorgestellt. Der Erste betrachtet alle Kombinationen der Suchterme der ursprünglichen Anfrage und überprüft die Anzahl an Dokumenten, die eine solche kombinierte Suchanfrage als Ergebnismenge liefert. Die Maximum Query entspricht der Suchtermkombination, die am längsten ist und gleichzeitig eine Anzahl von Dokumenten findet, die bestimmte Schwellenwerte weder über- noch unterschreitet. Der zweite Algorithmus reduziert die Anzahl an nötigen Suchanfragen, in dem er mit Hilfe eines *Co-occurence Graphs* betrachtet, welche Termkombinationen die Anzahl an gefundenen Dokumenten wie stark beeinflusst. So können Suchtermkombinationen ausgeschlossen werden, die die Ergebnisliste zu stark verkleinern. Der Begriff der Maximum Query ist auch für unser Verfahren interessant, wenn es dazu kommt Suchtermkandidaten zu einer Suchanfrage sinnvoll zusammenzustellen. Die von Hagen und Stein vorgeschlagene Vorgehensweise, Ergebnisdokumente mit den Ursprungs-

dokumenten zu vergleichen und als einzigen Feedback-Kanal die Suchmaschine zu benutzen, ist gut für unser Verfahren umsetzbar.

Im Fokus dieser Arbeit steht das Verwenden des Suchanfragen-Rekonstruktions-Verfahren für Clusterlabeling. Muhr et al. vergleichen bereits etablierte Algorithmen für Clusterlabeling [MKG10]. Beim *Maximum Term Weight Labeling* werden alle Terme eines Clusters betrachtet und die k häufigsten für das passende Label ausgewählt. Dieser einfache Ansatz kann durch das Beachten der *Inverse Document Frequency* jedes Terms erweitert werden. Ein Term ist dann gut geeignet für ein Clusterlabel, wenn dieser ein hohes $tf \cdot idf$-Gewicht besitzt. Dieses ist hoch, wenn ein Term sehr häufig im Cluster, aber selten im Gesamtdokumentkorpus vorkommt. Das *Inverse Cluster Weight Labeling* erweitert das Maximum Term Weight Labeling durch einen zusätzlichen Faktor, der angibt wie häufig ein Term in anderen Clustern vorkommt. Ein Term eignet sich mehr für das Clusterlabel, wenn sich dieser seltener in den benachbarten Clustern (Geschwister-Clustern) befindet. Ein weiteres Verfahren zum Clusterlabeling beachtet, ob eine statistisch signifikante Abhängigkeit zwischen einem Term und einer Dokumentmenge, bzw. dem Cluster, besteht. Diese Abhängigkeit wird mit Hilfe des so genannten χ^2-*Tests* oder der Jensen-Shannon Divergenz [FT04] nachgewiesen und die Terme aus den Clustern entsprechend gewichtet. Ebenfalls werden hier die k Terme für das Label ausgewählt, die das höchste Gewicht haben.

Zur Evaluierung dieser Verfahren wurden verschiedene Cluster-Korpusse, wie z.B. das Open Directory Project (ODP) oder TREC Ohsumed, verwendet. Diese enthalten Dokumente, die auf verschiedene Weise hierarchisch geclustert und gelabelt wurden. Für jede dieser Cluster wurden mit den eben genannten Verfahren Clusterlabels erstellt und deren Qualität durch *Mean Average Precision (MAP)* berechnet. Da sowohl die von Muhr et al. verwendeten Cluster-Korpusse, als auch das Evaluationsverfahren auf hierarchische Clusterlabeling-Verfahren ausgelegt sind, schlagen wir für unser Verfahren, das nur für flache Clusterhierachien funktioniert, eine andere Vorgehensweise für die Evaluation vor. Das *Ambient Dataset* enthält Dokumente für eine flache Clusterstruktur. Für jedes der 44 Themen enthält dieser Korpus jeweils bis zu 37 Unterthemen. Wir generieren für jedes Cluster der Unterthemen ein Clusterlabel und vergleichen mit Hilfe verschiedener Ähnlichkeitsmaße diese Labels mit den Referenzlabels des Ambient Datasets. Je ähnlicher ein generiertes Clusterlabel dem Referenzlabel ist, umso besser bewerten wir das jeweilige Clusterlabeling-Verfahren.

2.1 Die ChatNoir Suchmaschine

Das wichtigste Hilfsmittel zur Rekonstruktion einer Suchanfrage aus einer Dokumentmenge ist die Suchmaschine. Diese sollte als eine Art *Black Box* funktionieren und auf eine bestimmte Suchanfrage die passende Ergebnismenge von Dokumenten zurück geben. Für die Reproduzierbarkeit und Vergleichbarkeit von verschiedenen Versuchen ist zusätzlich erforderlich, dass auf die gleiche Suchanfrage die gleiche Ergebnismenge ausgegeben wird. Etablierte und häufig genutzte Suchmaschinen, wie Google und Bing besitzen zwar Schnittstellen, die zur Implementierung von Versuchen genutzt werden können, jedoch basieren diese auf dynamischen Dokumentkorpussen. Das bedeutet, dass die Ergebnisliste für eine bestimmte Suchanfrage nicht zu jeder Zeit gleich ist. Für die Versuche in dieser Arbeit benötigen wir eine Suchmaschine, die auf einem statischen Korpus von Dokumenten arbeitet.

Wir verwenden deshalb die Suchmaschine ChatNoir.[1] Diese basiert auf dem englischen Teil des ClueWeb09-Korpus.[2] Hierbei handelt es sich um einen statischen Ausschnitt des Internets, der Anfang 2009 von dem Language Technologies Institute der Carnegie Mellon University erstellt wurde und etwa 500.000.000 englischsprachige Webseiten umfasst. Das entspricht einer Datenmenge von etwa 12 Terabyte. ChatNoir verwendet die Ranking-Funktion Bm25f [ZCT+04]. Es handelt sich hierbei um eine Erweiterung der probabilistischen Ranking-Funktion Okapi Bm25 durch die Betrachtung von Anker- und Titeltexten sowie des PageRanks eines Dokuments. Suchanfragen werden bei der ChatNoir Suchmaschine standardmäßig mit und verknüpft. Das heißt ein Dokument gehört zu der Ergebnismenge, wenn es alle Terme der Suchanfrage enthält. Diese Suchmaschine erzeugt reproduzierbare Ergebnisse und ist deshalb geeignet für die Experimente in dieser Arbeit.

[1] http://webis15.medien.uni-weimar.de/chatnoir/
[2] http://boston.lti.cs.cmu.edu/clueweb09/

3 Was war die Suchanfrage?

Dieser Teil der Bachelorarbeit beschäftigt sich mit Methoden zur Rekonstruktion einer Suchanfrage basierend auf einer Dokumentmenge, die durch eine Suchmaschinensuche erhalten wurde. Die rekonstruierte Suchanfrage ist so zusammengesetzt, dass deren Ergebnisliste alle Dokumente der Ausgangsmenge und keine fremden Dokumente des Korpus enthält. Zur Rekonstruktion verwendet unser Verfahren nur die gegebene Dokumentmenge und die Suchmaschine. Genauere Kenntnisse über den gesamten Dokumentkorpus, Retrieval-Modell und andere statistische Informationen, wie beispielsweise Anfragelogs der Suchmaschine, beachten wir dabei nicht. Dies garantiert die Entwicklung eines unabhängigen Verfahrens, welches auf sämtliche Dokumentmengen, Suchmaschinen und Retrieval-Modelle angewendet werden kann.

3.1 Reverted Indexing

Bei der Rekonstruktion einer Suchanfrage ist es nicht sehr gewinnbringend, sich allein darauf zu konzentrieren, die vorhandene Retrieval-Funktion zu analysieren und inhaltlich umzukehren. Ein solchermaßen optimiertes System wäre nur für eine spezielle Suchmaschine anwendbar. Dies würde nicht unserer ursprünglichen Idee entsprechen, mögliche relevante und aussagekräftige Terme allein aus den Dokumenten der Ausgangsmenge zu extrahieren und diese zu einer Suchanfrage zusammenzustellen.

Deswegen entwickeln wir einen Ansatz, der unabhängig von der Beschaffenheit der Suchmaschine arbeitet. Dafür speichern wir in einer Hilfs-Datenstruktur zunächst, welche Dokumente sich mit welchen Termen am besten in der genutzten Suchmaschine finden lassen. Ausgehend von einer Dokumentmenge können wir so Suchtermkandidaten bestimmen, mit welchen sich eine passende Suchanfrage zusammenstellen lässt. Ein nützliches Werkzeug für die Umsetzung dieses Ansatzes ist der von Pickens et. al vorgestellte Reverted Index [PCG10]. Hierbei werden im Gegensatz zum Inverted Index Dokumente auf eine Menge von Suchtermen abgebildet.

Tabelle 3.1 zeigt anhand einer Beispiel-Dokumentmenge den Zusammenhang zwischen Inverted- und Reverted Index. Der Inverted Index (links), als Herzstück jeder modernen Suchmaschine, bildet unter anderen die Terme *aida,*

aida	$\rightarrow$	$\{d_4, d_2 \ldots d_7\}$		d_1	$\mapsto$	{**verdi,** *guiseppe* ...}
verdi	$\rightarrow$	$\{d_1, d_2, \ldots d_{14}\}$	$\longrightarrow$	d_2	$\mapsto$	{**aida, verdi, tickets** ...}
$\vdots$				$\vdots$		
tickets	$\rightarrow$	$\{d_2, d_N \ldots d_{16}\}$		d_N	$\mapsto$	{*shop*, **tickets** ...}

Tabelle 3.1: Reverted Index: Ausgehend von den Basisanfragen q_b werden Dokumente d_n gefunden, die zu Einträgen im Reverted Index werden (vgl. Abbildung 1 in [PCG10])

verdi und *tickets* auf eine Menge von Dokumenten ab. Bekommt eine Suchmaschine beispielsweise die Anfrage „aida verdi“, so verarbeitet diese die Dokumentlisten dieser beiden Terme zu einer Ergebnisliste. Bei der Suchanfragen-Rekonstruktion verarbeiten wir eine Menge von Dokumenten. In dem Reverted Index (rechts) finden wir zu jedem Dokument aus der Dokumentmenge, für die eine Suchanfrage rekonstruiert werden soll, Suchterme, aus denen wir eine Suchanfrage zusammenstellen können.

Ausgangspunkt für die Konstruktion des Reverted Index ist eine Menge von sogenannten *Basisanfragen*, welche an eine Suchmaschine gestellt werden. Für diese Basisanfragen können entweder Anfragen aus einem Anfragelog entnommen werden, oder man extrahiert diese aus den Dokumenten der Ursprungsmenge. Bei letzterem Vorgehen kann man beispielsweise Terme oder Phrasen mit bestimmten Termhäufigkeiten für die Basisanfragen auswählen.

Jede der Suchanfragen aus der Menge der Basisanfragen wird an eine Suchmaschine gestellt. Die erhaltene Ergebnisliste wird anschließend auf eine bestimmte Größe, beispielsweise auf die Top 1000 Dokumente, reduziert, da wir annehmen, dass ein Dokument mit sehr niedrigem Rang, wenig relevant für die Suchanfrage ist. Jedes der Dokumente aus den gekürzten Ergebnislisten wird ein Schlüssel in dem Reverted Index. Die passenden Wertemengen bilden jeweils die Terme der Basisanfragen. An dieser Stelle ist es zusätzlich möglich, die Terme der Basisanfragen zu gewichten. Erscheint ein Dokument in der Ergebnisliste an erster Stelle, so ist dieses sehr relevant für die Suchanfrage und kann deshalb hoch gewichtet werden. Da in dieser Arbeit jedoch nur mit einer kleinen Menge von Basisanfragen gearbeitet wird, wird auf die Gewichtung der Terme an dieser Stelle verzichtet.

3.2 Bestimmung der Basisanfragen

Das Erstellen des Reverted Index benötigt eine Menge von so genannten Basisanfragen. Diese repräsentieren wichtige Terme aus dem Dokumentkorpus und bilden die Grundlage für die Rekonstruktion der Suchanfragen. Jeder der Ter-

	w_1	w_2	w_3	w_4	w_5
d_1	3	4	2	0	0
d_2	0	0	0	6	2
d_3	0	9	2	3	0
d_c	1	4.33	1.33	3	*0.66*

Tabelle 3.2: Beispiel für die Bestimmung des Centroid-Dokuments d_c der Dokumentmenge d_1,d_2 und d_3. Der Wert für w_5 ist < 1 und wird deshalb nicht in d_c aufgenommen.

me wird an eine Suchmaschine gestellt und das Ergebnis für den Reverted Index verarbeitet. Ein Problem dieser Vorgehensweise ist, dass jede Suchanfrage Zeit kostet und den Prozess der Erstellung des Reverted Index deutlich verlangsamt. Deswegen wählen wir die Menge der Basisanfragen so, dass so wenig Suchanfragen durchgeführt werden müssen wie nötig. Gleichzeitig soll die Menge der Basisanfragen so groß sein, dass mit einer hohen Wahrscheinlichkeit alle Terme der ursprünglichen Anfrage, also der Anfrage mit der die Menge der Ursprungsdokumente erhalten wurde, enthalten sind.

Wir schlagen für die Extrahierung der Basisanfragen aus einer gegebenen Menge von Dokumenten zwei Verfahren vor. Das erste Verfahren verwendet für die Basisanfragen die n häufigsten Terme der Dokumentmenge (Termhäufigkeits-Verfahren). Beim zweiten Verfahren verwenden wir alle Terme des Centroid-Dokuments (Centroid-Verfahren). Bei letzterem Verfahren werden die Dokumente der gegebenen Menge als tf-Vektor repräsentiert. Das Centroid-Dokument bildet den arithmetischen Mittelpunkt des Vektorraums der Dokumentvektoren. Es werden also beim Centroid-Verfahren die Terme für die Basisanfragen verwendet, die durchschnittlich mindestens einmal in jedem Dokument der gegebenen Menge vorkommen. Tabelle 3.2 zeigt an einem Beispiel die Bestimmung des Centroid-Dokuments d_c aus den Dokumenten d_1, d_2 und d_3. Jede Komponente des Centroid-Vektors berechnet sich aus der durchschnittlichen Häufigkeit eines Terms aus der Dokumentmenge. Für die Basisanfragen verwenden wir die Terme mit der durchschnittlichen Häufigkeit ≥ 1. Da in dem Beispiel in Tabelle 3.2 der Term w_5 eine durchschnittliche Häufigkeit < 1 hat, wird dieser nicht in dem Centroid-Dokument aufgenommen.

Für die vergleichende Evaluation des Termhäufigkeits-Verfahrens und des Centroid-Verfahrens, erstellen wir Basisanfragen aus Dokumentmengen eines selbst zusammengestellten Dokumentkorpus. Hierfür haben wir die 791 Unterthemen des Ambient Datasets an die Suchmaschine Bing[1] gestellt und jeweils die Top 50 Dokumente der Ergebnisliste zu einem Cluster zusammengefasst.

[1] http://www.bing.com

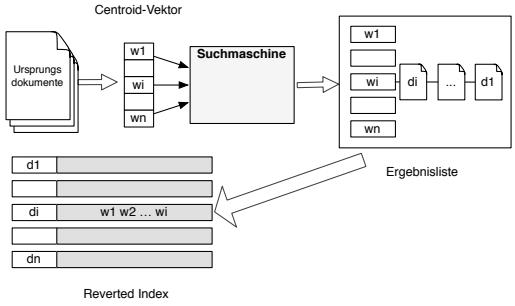


Abbildung 3.1: Konstruktion des Reverted Index

Diesen Dokumentkoprus verwenden wir an späterer Stelle auch für die Evaluation von Clusterlabeling-Verfahren (siehe Kapitel 5). Für jede der Dokumentmengen bestimmen wir das Centroid-Dokument. Dieses hat eine durchschnittliche Länge von 87,23 Wörtern. Das heißt, dass mit dem Centroid-Verfahren durschnittlich 87,23 Basisanfragen gestellt werden müssen, um den Reverted Index zu konstruieren. Unter diesen Anfragen befinden sich durchschnittlich 90% der Terme der ursprünglichen Anfragen. Mit dem Termhäufigkeits-Verfahren extrahieren wir die Top 88 häufigsten Terme aus den Dokumentmengen des Evaluations-Korpus. Für n wurde der Wert 88 gewählt, damit eine Vergleichbarkeit mit dem Centroid-Verfahren gegeben ist. Unter diesen Basis-Anfragen befinden sich 87% der Terme der ursprünglichen Anfragen.

Das Centroid-Verfahren sowie das Termhäufigkeits-Verfahren liefern also für die Bestimmung der Basisanfragen aus einer Dokumentmenge ein ähnliches Ergebnis.

3.3 Rekonstruktion mit Reverted Index

Die Rekonstruktion einer Suchanfrage mit Hilfe des Reverted Index lässt sich in drei Phasen einteilen: der Konstruktion des Reverted Index, dem Finden der Suchtermkandidaten und dem Zusammensetzen der Suchanfrage (Compositing).

Die sicherste Methode, die perfekte Suchanfrage zu rekonstruieren, ist das Probieren aller Kombinationen aller Suchtermkandidaten. Da hierfür sehr viele zeitaufwendige Anfragen an die Suchmaschine nötig sind, schlagen wir eine andere Vorgehensweise vor. Wir gewichten die Suchtermkandidaten danach, wie viele Dokumente der Menge der Ursprungsdokumente jeder Term findet. Anschließend können wir mit der geordneten Liste der Suchtermkandidaten schrittweise, Term für Term, die Suchanfrage zusammenstellen.

Konstruktion des Reverted Index

In Abbildung 3.1 ist die Konstruktion des Reverted Index schematisch zusammengefasst. Aus der gegebenen Menge der Ursprungsdokumente wird der Centroid-Vektor bestimmt. Dieser enthält die Terme der Dokumentmenge, die durchschnittlich mindestens einmal pro Dokument vorkommen. Hierbei schließen wir Stoppworte aus. Jeder Term des Centroid-Vektors wird als Basisanfrage an eine Suchmaschine gestellt. So wird für jede Basisanfrage eine Ergebnisliste von relevanten Dokumenten bestimmt. Für jedes Dokument der Ergebnislisten wird ein Eintrag im Reverted Index angelegt. Jedes Dokument wird dabei auf die Menge seiner Basisanfragen abgebildet.

Finden der Suchtermkandidaten

Den Algorithmus zum Finden der Suchtermkandidaten ist in Listing 3.1 in Pseudocode beschrieben. Beim Finden der Suchtermkandidaten kommt der vorher erstellte Reverted Index zum Einsatz. Für jedes Dokument d aus der Ursprungsdokumentmenge D wird die passende Menge von Termen W_d aus dem Reverted Index entnommen (siehe Listing 3.1, Zeile 5). Diese Terme repräsentieren die besten Suchworte, um das jeweilige Dokument mit einer Suchmaschine zu finden. All diese Terme werden in einer Map zwischengespeichert (Zeilen 3-7). Zusätzlich wird in dieser Map mitgezählt, für wie viele Dokumente sich jeder Term w als Suchterm eignet (Zeile 7). Um das Compositing der Suchtermkandidaten zu vereinfachen, berechnen wir für jeden Term ein Gewicht. Dieses Gewicht setzt sich zusammen aus der Anzahl $\#d$ der Dokumente aus D, die mit dem jeweiligen Term gefunden werden können, geteilt durch die Anzahl aller Dokumente in D. Die Anzahl $\#d$ entnehmen wir der Map (Zeile 10). Jeden Term w aus der Map fügen wir zusammen mit seinem Gewicht der Liste der Suchtermkandidaten $W_{Kandidat}$ hinzu (Zeile 12) und ordnen diese abschließend absteigend nach Gewicht. Das Ergebnis dieses Rekonstruktionschritts ist eine Liste von Suchtermen, die anhand ihres Gewichts geordnet wurden.

Compositing

Beim Compositing werden die Terme der Suchtermkandidaten-Liste zu einer Suchanfrage zusammengesetzt, welche gestellt an eine Suchmaschine möglichst alle Dokumente der Ursprungsdokumentmenge findet. Es handelt sich hierbei um ein Mengenüberdeckungsproblem. Hinter jedem Term der Suchtermkandidaten verbirgt sich eine Menge von Dokumenten, die zusammen genommen die Menge der Ursprungsdokumente abdecken soll. Hierfür setzen wir schrittweise aus Suchtermkandidaten eine Anfrage Q zusammen, stellen diese an eine

Listing 3.1: Algorithmus zum Bestimmen der Suchtermkandidaten

```
1  Eingabe: Ursprungsdokumente D, RevertedIndex
2  Ausgabe: Suchtermkandidaten W_Kandidat
3  Map ← ∅
4  for all d ∈ D do:
5      W_d ← RevertedIndex(d)
6      for all w ∈ W_d do:
7          Map(w) ← Map(w) + 1
8  W_Kandidat ← ∅
9  for all w ∈ Map do:
10     #d ← Map(w)
11     gewicht ← #d : |D|
12     W_Kandidat ← W_Kandidat ∪ {{w, gewicht}}
13 Sortiere W_Kandidat absteigend nach gewicht
14 Ausgabe W_Kandidat
```

Suchmaschine und vergleichen die Top n Dokumente der Ergebnisliste mit der Menge D der Ursprungsdokumente.

Listing 3.2 zeigt die genaue Vorgehensweise in Pseudocode implementiert. Gegeben ist die geordnete Liste der Suchtermkandidaten $W_{Kandidat}$, die Menge der Ursprungsdokumente D und ein Faktor n. Letzterer gibt an, bis zu welchen Rang der Ergebnisliste sich Dokumente aus D befinden dürfen. Es ist deshalb sinnvoll ein $n \geq |D|$ zu wählen. Die Liste $W_{Kandidat}$ wird Term für Term durchgegangen (siehe Listing 3.2, Zeile 5). Dabei wird mit dem Term mit dem größten Gewicht begonnen. In jedem Schritt wird dabei der Term w zu der Suchanfrage Q hinzugefügt (Zeile 6) und Q als Anfrage an eine Suchmaschine gestellt. Die Dokumentmenge $D_{Top\text{-}n}$ besteht aus den Top n Dokumenten der Ergebnisliste dieser Suchanfrage. Anschließend wird ein Vergleichswert v berechnet, der ausdrückt, wie viele Dokumente aus D in der Ergebnisliste $D_{Top\text{-}n}$ enthalten sind (Zeile 8). Dieser Wert v wird verwendet, um zu überprüfen, ob das Hinzufügen von w zu einer Verbesserung der Suchanfrage führt. Ist dieser größer oder gleich des Vergleichwerts der letzten Suchanfrage ohne w (Zeile 9), so befinden sich dank w mehr Dokumente aus D in der Menge $D_{Top\text{-}n}$ und das Compositing wird mit dem nächsten Term fortgesetzt. Ist der Vergleichswert kleiner, so führt w zur Verschlechterung des Ergebnis. Das Compositing wird an dieser Stelle abgebrochen, da wir davon ausgehen, dass die Suchtermkandidaten mit kleinerem Gewicht das Ergebnis weiter verschlechtern. Der *schwache* Term w wird aus Q entfernt und die fertige Suchanfrage Q als Ergebnis ausgegeben.

Ergebnis des Compositing-Schritts ist eine Menge von Suchtermen, die als Suchanfrage an eine Suchmaschine viele Dokumente der Ursprungsdokumente wiederfindet.

Listing 3.2: Algorithmus zum Compositing der Suchtermkandidaten

```
1 Eingabe: Suchtermkandidaten W_Kandidat, Ursprungsdokumente D, n
2 Ausgabe: Suchanfrage Q
3 v ← 0
4 Q ← ∅
5 for all w ∈ W_Kandidat do:
6     Q ← Q ∪ {w}
7     D_Top-n ← Suche(Q, n)
8     v' ← |D_Top-n ∩ D_u| : |D_u|
9     if v' ≥ v:
10        v ← v'
11    else:
12        Q ← Q \ {w}
13        break
14 Ausgabe Q
```

4 Clusterlabeling

Das im letzten Kapitel vorgestellte Verfahren zur Rekonstruktion einer Suchanfrage aus einer gegebenen Menge von Dokumenten soll nun für das Clusterlabeling angewendet werden. Hierfür sind Anpassungen des Verfahrens notwendig, da durch das neue Nutzungsszenario zusätzliche Anforderungen an die rekonstruierte Suchanfrage hinzu kommen. Abschließend wird im nächsten Kapitel unser neuer Ansatz für Clusterlabeling mit etablierten Clusterlabelingverfahren verglichen und evaluiert.

4.1 Clusterlabeling im Allgemeinen

Das Clustern von Dokumenten ist eine Möglichkeit, Dokumentmengen zu strukturieren. Hierbei werden inhaltlich ähnliche Dokumente zu einem so genannten Cluster zusammengefasst. Man kann Clustering beispielsweise mit einer Menge von Dokumenten zu einem mehrdeutigen Begriff deutlich machen. So kann man etwa dem Begriff „Aida“ Dokumente über die Oper von Guiseppe Verdi, über das gleichnamige Musical oder über das Akronym aus dem Bereich des Marketing (Attention Interest Desire Action) zuordnen. Diese Unterthemen bilden je ein Cluster zu dem Überthema Aida. Die Gesamtheit aller Cluster zu einem Überthema wird als *Clustering* bezeichnet. Ausgehend von einem Cluster in einem Clustering werden die anderen Cluster des Clusterings als *benachbarte Cluster* bezeichnet.

Unter dem Begriff des *Clusterlabelings* fasst man Methoden zusammen, die für solche Cluster kurze Titel (Labels) generieren. Diese helfen dem Nutzer, einen Überblick über den Inhalt des Clusters zu bekommen und machen eine Interaktion mit den Clustern möglich.

Im Allgemeinen gibt es zwei große Gruppen von Clusterlabeling-Verfahren: das *Differential Clusterlabeling* und das *Cluster-Internal Labeling* [MRS08, Abschnitt 17.7]. Beim Differential Clusterlabeling wird die Verteilung der Terme innerhalb eines Clusters mit den Verteilungen der benachbarten Cluster verglichen und ermittelt, ob eine (statistische) Abhängigkeit von einem Term und dessen Auftreten in einem Cluster nachweisbar ist. Hierfür können Techniken der *Feature Selection* verwendet werden. Neben der Mutual Information sind hierfür die Jensen-Shannon Divergence und der χ^2-*Test* geeignet. Da diese ähn-

liche Ergebnisse beim Clusterlabeling liefern (siehe [MRS08, Abschnitt 13.5.2] und [MKG10]), benutzen wir bei der späteren Evaluation der Clusterlabeling-Verfahren den χ^2-Test als Stellvertreter für Differential Clusterlabeling (siehe Abschnitt 4.1.1).

Das *Cluster-Internal Labeling* bezieht sich beim Erstellen des Labels ausschließlich auf die Dokumente eines Clusters und lässt benachbarte Cluster unberücksichtigt. Hierbei spielt häufig das Centroid-Dokument eine große Rolle. Das Centroid-Dokument stellt den arithmetischen Mittelpunkt im Vektorraum des Dokument-Clusters dar. Als Stellvertreter für die Verfahren des Cluster-Internal Labelings wird in dieser Arbeit die *Weighted Centroid Coverage* verwendet (siehe Abschnitt 4.1.2).

4.1.1 χ^2-Clusterlabeling

Das χ^2-Clusterlabeling, als Vertreter des Differential Clusterlabelings, verwendet den χ^2-Test, um herauszufinden, ob eine statistische Abhängigkeit zwischen einem Term und einem Cluster besteht. Der χ^2-Test ist in der Statistik eine Methode, um zu bestimmen, ob das Auftreten zweier Ereignisse stochastisch unabhängig ist. Hierbei wird ein Wert berechnet, der umso größer ist, je höher die Wahrscheinlichkeit ist, dass eine Abhängigkeit zwischen den zwei Ereignissen besteht. Beim Clusterlabeling sind die Ereignisse „Term tritt im Clustering auf“ und „Term befindet sich im Cluster“ von Interesse. Ist die Abhängigkeit zwischen diesen zwei Ereignissen statistisch signifikant nachweisbar, so eignet sich der Term als Clusterlabel. Für die Berechnung des χ^2-Wertes benötigt man jeweils die Verteilung des zu überprüfenden Terms in und außerhalb des Clusters. Tabelle 4.1 zeigt an einem Beispiel, wie so eine Verteilung für ein Clustering von 500 Dokumenten aussehen kann. Hierbei wird ein Term untersucht, der in 70 Dokumenten eines Clusters D_c (mit einer Größe von 100 Dokumenten) und in 100 Dokumenten der anderen Cluster vorkommt. Die Abhängigkeit zwischen Term und Cluster lässt sich wie folgt berechnen (Ergebnis mit Werten aus Tabelle 4.1):

$$\begin{aligned}\chi^2 &= \frac{(N_{1,1}+N_{1,0}+N_{0,0})\times(N_{1,1}N_{0,0}-N_{1,0}N_{0,1})^2}{(N_{1,1}+N_{0,1})\times(N_{1,1}+N_{1,0})\times(N_{1,0}+N_{0,0})\times(N_{0,1}+N_{0,0})}\\ &= 72{,}193.\end{aligned}$$

In der χ^2-Verteilung kann mit Hilfe diesen Werts nun abgelesen werden, mit welcher Wahrscheinlichkeit eine Abhängikeit zwischen Term und Cluster herrscht. Ist der Wert für χ^2 größer als 10,83, so kann eine Abhängigkeit zwischen dem untersuchtem Term und dem Cluster (mit einer signifikanten Wahrscheinlichkeit) angenommen werden [MRS08, Abschnitt 13.5.2]. Ist die Abhängigkeit nachgewiesen, so wird der Term als ein möglicher Labelterm ausgewählt.

	$d \in D_c$	$d \notin D_c$
$w \in d$	$N_{1,1} = 70$	$N_{1,0} = 100$
$w \notin d$	$N_{0,1} = 30$	$N_{0,0} = 300$

Tabelle 4.1: Beispielverteilung eines Terms w in einer geclusterten Dokumentmenge D_c, bestehend aus den Dokumenten d.

Der Term aus dem Zahlenbeispiel wäre somit für das Clusterlabel geeignet (da 72,193 > 10,83). Dieser Test wird für alle Terme des Clusters durchgeführt. Lässt man dabei Terme mit niedriger Häufigkeit aus, erhält man ein besseres Ergebnis [MRS08, Abschnitt 17.7]. Für das Clusterlabel werden dann die k Terme mit den höchsten χ^2-Werten ausgewählt. Wie groß dabei k gewählt wird, hängt davon ab, wie viele Wörter das Clusterlabel enthalten soll. Die Wahl dieses Parameters muss also manuell geschehen, was das automatische Clusterlabeling erschwert.

4.1.2 Weighted Centroid Covering

Als Stellvertreter für das Cluster-Internal Labeling wählen wir für diese Arbeit das Weighted Centroid Covering [SM04]. Dabei betrachtet man die häufigsten Terme des gesamten Clusterings und ordnet diese nach ihrer Häufigkeit absteigend in einer Liste. Diese wird *top-down*, also von dem Term mit der höchsten bis zur niedrigsten Häufigkeit in allen Clustern, durchgegangen und dabei die Terme jeweils einem Cluster zugeordnet. Dabei wird das Cluster ausgewählt, das den Term am häufigsten in seinem Centroid-Dokument enthält. Dies wird so lange gemacht, bis jedem der Cluster eine bestimmte Anzahl von Termen k zugeordnet wurde. Diese k Terme werden jeweils als Clusterlabel verwendet. Wie beim χ^2-Verfahren muss auch bei der Weighted Centroid Covering k manuell bestimmt werden, was das automatische Clusterlabeling erschwert.

4.2 Clusterlabeling im Retrievalkontext

Eine zentrale Frage dieser Arbeit ist, ob eine Suchanfrage, welche aus einer Dokumentmenge rekonstruiert wurde, auch als Clusterlabel einsetzbar ist. Die zugrundeliegende Idee ist insofern interessant, da eine Suchanfrage auch als eine Art Überschrift für die erhaltene Ergebnismenge gesehen werden kann. Diese besteht nämlich nur aus Dokumenten, die relevant für die Suchanfrage sind. Das in Kapitel 3 vorgestellte Verfahren zur Rekonstruktion von Suchanfragen mit Hilfe des Reverted Index wird als Grundlage genommen und erweitert.

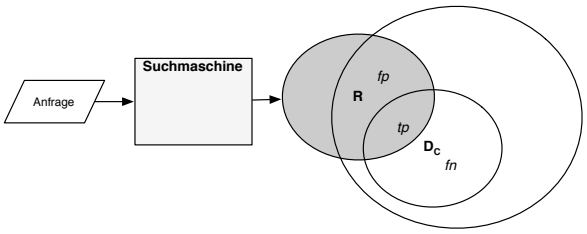


Abbildung 4.1: Darstellung der true positives tp, false positives fp und false negatives fn in der Ergebnisliste R und in dem Cluster D_c

Wir verwenden das Clusterlabeling mit Hilfe der Anfragen-Rekonstruktion als Differential Clusterlabeling. Wir betrachten also nicht nur die Dokumente des zu labelnden Clusters, sondern auch die Dokumente der benachbarten Cluster. Am Ende entspricht das Clusterlabel einer Suchanfrage, deren Ergebnismenge möglichst viele Dokumente des zu labelnden Clusters und möglichst wenige Dokumente der im Clustering benachbarten Cluster enthält. Dafür muss in der Rekonstruktions-Pipeline der Schritt des Finden der Suchtermkandidaten angepasst werden. In diesem Schritt werden die Suchtermkandidaten nach einer Gewichtsformel gewichtet. Die nach Gewicht sortierte Liste wird dann im Compositing-Schritt top-down abgearbeitet und schrittweise Terme zur finalen Suchanfrage hinzugefügt. Dabei wird überprüft, ob das Hinzufügen des Terms zu einer Verbesserung der Suchanfrage führt, indem immer wieder die Ergebnisliste der aktuellen Suchanfrage mit der Ursprungsdokumentmenge verglichen wird (vgl. Abschnitt 3.3). Die verwendete Gewichtsformel drückt bei der einfachen Anfragen-Rekonstruktion aus, wie viele Dokumente der Ursprungsmenge mit einem Term gefunden werden können. Beim Differential Clusterlabeling wird jedoch gleichzeitig beachtet, dass möglichst viele clusterfremde Dokumente ausgeschlossen werden. Für diese Bedingung muss die Gewichtung der Suchtermkandidaten erweitert werden.

Termgewichtung mit F-Measure

Für das Clusterlabeling betrachten wir jeden Suchtermkandidaten, der mit Hilfe des Reverted Index ermittelt wurde, als einen Klassifikator, der aus dem Dokumentkorpus Dokumente eines Clusters auswählen und Dokumente der anderen Cluster ausschließen kann. Wir schlagen für das Gewicht jedes Suchtermkandidaten die Bewertung des Klassifikators mit Hilfe des *F-Measures* vor. Dieses setzt sich zusammen aus *Recall*, also die Fähigkeit des Klassifikators Clusterdokumente zu finden, und *Precision*, der Fähigkeit des Klassifikators clusterfremde Dokumente auszuschließen.

Für die Berechnung des F-Measures sind folgende Teilmengen des Clusterings gegeben: die *true positives* (tp) sind die Dokumente, die sich im Cluster befinden und von dem Suchwort gefunden werden. Die *false positives* (fp) sind die Dokumente, die vom Suchwort gefunden werden, aber zu anderen Clustern gehören. Die *false negatives* (fn) sind die Dokumente, die sich im Cluster befinden, aber nicht von dem Suchwort gefunden werden (vgl. Abbildung 4.1). Recall entspricht dem Anteil der true positives in den Clusterdokumenten ($C = tp \cup fn$). Precision entspricht dem Anteil der true positives in der Ergebnisliste ($R = tp \cup fp$). Das F-Measure berechnet sich wie folgt:

$$F = \frac{(1 + \beta^2) \times |tp|}{(1 + \beta^2) \times |tp| + \beta^2 \times |fn| + |fp|}.$$

Mit Hilfe des Faktors β kann das Verhältnis zwischen Precision und Recall angepasst werden. Bei einem $\beta < 1$ wird Precision höher gewichtet als Recall, also der mehr Fokus darauf gelegt, dass clusterfremde Dokumente ausgeschlossen werden (false negatives minimieren). Bei einem $\beta > 1$ wird Recall höher gewichtet als Precision, also der der Fokus mehr auf die Fähigkeit des Klassifikators gelegt, Clusterdokumente zu finden (true positives zu maximieren).

In Listing 4.1 ist der für Clusterlabeling angepasste Algorithmus zur Bestimmung der Suchtermkandidaten in Pseudocode beschrieben. Im Gegensatz zum Algorithmus der einfachen Anfragen-Rekonstruktion (vgl. Abschnitt 3.3, Listing 3.1) betrachten wir neben den Dokumenten, die in der Ergebnisliste der rekonstruierten Anfrage vorkommen sollen (im Fall des Clusterlabelings die Menge der Clusterdokumente D_c), auch die Menge der Dokumente, die ausgeschlossen werden sollen (die Menge der clusterfremden Dokumente D_f). Für beide Dokumentmengen wird mit Hilfe des Reverted Index bestimmt, welche Terme jeweils wie viele true positives ($d \in D_c$) und false positives ($d \in D_f$) finden. Diese Zuordnung wird jeweils in einer Map für die true positives Map_{tp} und einer Map für die false positives Map_{fp} zwischengespeichert(siehe Listing 4.1, Zeile 3 und 4). Alle Terme in Map_{tp} werden anschließend mit dem F-Measure gewichtet und zur Menge der Suchtermkandidaten hinzugefügt (Zeile 10 und 11). Die Werte für tp und fp werden für den Term w aus der jeweiligen Map entnommen (Zeile 7 und 8). Der Wert für fn, also die Anzahl der false negatives, berechnet sich aus der Differenz des Betrags von D_c und tp (Zeile 9). Abschließend wird die Liste $W_{Kandidat}$ absteigend nach Gewicht sortiert und Ausgegeben (siehe Zeile 12 und 13)

Wurden alle Suchtermkandidaten ermittelt und gewichtet, kann das Compositing, wie in Abschnitt 3.3 beschrieben, fortgesetzt werden. Die fertig zusammengestellte Suchanfrage des Clusters wird als Clusterlabel verwendet.

Listing 4.1: Algorithmus zum Bestimmen der Suchtermkandidaten für Clusterlabeling

```
1 Eingabe: Clusterdokumente D_c, clusterfremde Dokumente D_f, RevertedIndex
2 Ausgabe: Suchtermkandidaten W_Kandidat

3 Map_tp ←konstruiereMap(Dokumentmenge D_c, RevertedIndex)
4 Map_fp ←konstruiereMap(Dokumentmenge D_f, RevertedIndex)
5 W_Kandidat ← ∅
6 for all w ∈ Map_tp do:
7     tp ← Map_tp(w)
8     fp ← Map_fp(w)
9     fn ← |D_c| − tp
10    gewicht ← fscore(tp, fp, fn)
11    W_Kandidat ← W_Kandidat ∪ {{w, gewicht}}
12 Sortiere W_Kandidat absteigend nach gewicht
13 Ausgabe W_Kandidat

14 Funktion konstruiereMap(Dokumentmenge D, RevertedIndex)
15 Map ← ∅
16 for all d ∈ D do:
17    W_d ← RevertedIndex(d)
18    for all w ∈ W_d do:
19        Map(w) += 1
20 return Map
```

5 Evaluation

In diesem Teil der Arbeit evaluieren wir das neu entwickelte Clusterlabeling-Verfahren und vergleichen dieses mit bekannten Verfahren. Hierfür wenden wir zwei Arten der Evaluation an. Mit Hilfe von maschinellen Methoden zum Ähnlichkeitsvergleich zweier Textstrings, werden generierte Clusterlabels mit dem passenden Referenzlabel verglichen. Das Referenzlabel wird hierbei als das bestmögliche Label für ein Cluster angenommen. Je ähnlicher ein generiertes Label dem Referenzlabel ist, umso besser wird das Clusterlabeling-Verfahren bewertet. Dafür verwenden wir neben den klassischen Ähnlichkeitsmaßen Jaccard-Index und Kosinus-Ähnlichkeit auch das F-Measure und ein weiteres Ähnlichkeitsmaß, das die Kosinus-Ähnlichkeit durch semantische Merkmale erweitert. Diese *Explicit Semantic Analysis* wurde bisher noch nicht für die Evaluation von Clusterlabels genutzt.

Die zweite Art der Evaluation der Clusterlabeling-Verfahren ist eine Nutzerstudie, bei der Nutzer die generierten Clusterlabels bewerten.

Anhand der Ergebnisse können wir beurteilen, wie sich die Qualität der mit dem Rekonstruktions-Verfahren erstellten Clusterlabels im Vergleich zu anderen Clusterlabeling-Verfahren unterscheidet. Durch eine detaillierte Analyse werden außerdem Schwachstellen und Stärken der einzelnen Clusterlabeling-Verfahren deutlich.

5.1 Der Cluster-Korpus

Für die Evaluierung der Clusterlabeling-Verfahren ist ein Korpus von Clusterings verschiedener Dokumentmengen notwendig. Dieser wird dazu verwendet, Clusterlabels zu erstellen.

Das Ambient Dataset[1] (Abkürzung für AMBIgous ENTries) bildet die Grundlage für den Experiment-Korpus. Es wurde bereits von verscheidenen Forschungsgruppen für Experimente von Clustering- und Clusterlabeling-Verfahren verwendet [NC10, SGH11, TC11]. Das Ambient Dataset gliedert sich in 44 Themen zu mehrdeutigen Begriffen. Mit Hilfe der *Disambiguation Pages*[2]

[1] http://credo.fub.it/ambient/

[2] http://en.wikipedia.org/wiki/Wikipedia:Links_to_%28disambiguation%29_pages

von Wikipedia wurden diese in insgesamt 791 Unterthemen unterteilt. Die kurzen Beschreibungen dieser Unterthemen in den Disambiguation Pages werden als die *Referenzlabel* der Cluster verwendet.

Für einige der Unterthemen wurden dem Ambient Dataset durch *Relevance Judgements* Dokumente zugeordnet. Dafür wurde jedes der Überthemen als Anfrage an eine Suchmaschine gestellt und die Top 100 Dokumente der Ergebnisliste von einer Nutzergruppe in die jeweiligen Unterthemen eingeordnet. Da für jedes der 44 Themen bis zu 37 Unterthemen existieren, bekommen viele der Unterthemen (also der eigentlichen Cluster) wenige oder gar keine Dokumente zugeordnet. Um beim Clusterlabeling gute und eindeutige Ergebnisse zu erhalten, ist es von Vorteil, wenn wir mit Clustern arbeiten können, die viele Dokumente enthalten. Deshalb wird im Rahmen dieser Arbeit ein selbst zusammengestellter Dokumentkorpus benutzt, welcher 50 Dokumente für jedes der Unterthemen des Ambient Datasets enthält. Hierfür wurde jedes Referenzlabel der Unterthemen als Anfrage an die Suchmaschine Bing gestellt und die Top 50 Dokumente zu einem Cluster zusammengefasst.

Um das Suchanfragen-Rekonstruktions-Verfahren anzuwenden, benötigen wir eine Suchmaschine, die alle Dokumente des Cluster-Korpus im indizierten Dokumentkorpus enthält. Dafür indizieren wir den Cluster-Korpus mit Hilfe des Retrieval-Modells Bm25f und erweitern den Index der ChatNoir-Suchmaschine.

5.2 Maschinelle Evaluationsverfahren

Für das Evaluieren der Clusterlabeling-Verfahren werden generierte Clusterlabels mit Referenzlabels verglichen. Dies kann zum einen mit maschinell bestimmten Ähnlichkeitswerten zweier Labelstrings bewertet werden. Je höher dabei der Ähnlichkeitswert ist, umso besser bewerten wir das Clusterlabeling-Verfahren.

5.2.1 Klassische Evaluationsverfahren

Für einen ausführliche Vergleich der Verfahren χ^2-Test, Weighted Centroid Covering und Suchanfragen-Rekonstruktion werden mit Hilfe von F-Measure, Jaccard Index und Kosinuähnlichkeit maschinell Ähnlichkeitswerte bestimmt. Wir bezeichnen den Ähnlichkeitsvergleich mit diesen Verfahren als klassische Evaluationsverfahren.

5.2.1.1 F-Measure

Bei dem F-Measure handelt es sich um ein Maß, bei dem die Clusterlabeling-Verfahren als ein klassifizierendes Verfahren betrachtet werden. Das heißt ein Verfahren wird gut bewertet, wenn es in der Lage ist, ein Label zu generieren, das viele Terme des passenden Referenzlabels enthält und gleichzeitig nur wenige Terme beinhaltet, die sich nicht im Referenzlabel befinden. Dabei werden die beiden Labels als Mengen von Termen betrachtet und verglichen. Der Wert des F-Measure ist das harmonische Mittel aus Recall und Precision. Recall entspricht dem Anteil der Terme des generierten Labels, die auch in dem Referenzlabel vorkommen.

$$Recall = \frac{|Label \cap Referenz|}{|Label|}$$

Precision ist der Anteil der Terme des Referenzlabels, die mit Hilfe des Clusterlabeling-Verfahrens gefunden wurden.

$$Precision = \frac{|Label \cap Referenz|}{|Referenz|}$$

Mit anderen Worten beschreibt Recall die Fähigkeit des Clusterlabeling-Verfahrens, die richtigen Terme dem Clusterlabel zuzuordnen und Precision die Fähigkeit fremde Terme auszuschließen. Ob ein Term *richtig* oder *fremd* ist, hängt davon ab, ob er in dem Referenzlabel vorkommt oder nicht. Der Wert F berechnet sich aus dem harmonischen Mittel aus Precision und Recall.

$$F = 2 \cdot \frac{Recall \cdot Precision}{Precision + Recall}$$

In folgendem Beispiel wird das generierte Label „Gabriel Physiker Daniel Danzig“ mit dem passenden Referenzlabel „Gabriel Fahrenheit, ein deutscher Physiker“ mit Hilfe des F-Measure verglichen. Zuerst werden beide Strings in Einzelterme zerlegt, diese auf den Wortstamm reduziert und in Kleinschreibung umgewandelt. Bei der Berechnung der Ähnlichkeit werden die Terme des Überthemas, in diesem Fall *Fahrenheit*, nicht beachtet. Die beiden Labels ergeben also folgende Termmengen:

$$\begin{aligned} Label &= \{\mathbf{gabriel}, \mathbf{physik}, daniel, danzig\} \\ Referenz &= \{\mathbf{gabriel}, deutsch, \mathbf{physik}\} \end{aligned}$$

In der Mengendarstellung wurden diejenigen Terme fett gedruckt, die in beiden Mengen vor kommen. Zwei der Terme des Referenzlabels wurden auch mit dem Clusterlabeling-Verfahren dem generierten Label zugeordnet. Das bedeutet,

dass die Hälfte der Terme des Labels richtige Terme sind - also einen Recall von 0,5. Insgesamt umfasst das Referenzlabel drei verschiedene Terme. Davon wurden zwei mit Hilfe des Clusterlabeling-Verfahrens gefunden. Das bedeutet eine Precision von 0,66. Aus dem harmonischen Mittel zwischen Precision und Recall ergibt sich ein F von 0,57.

5.2.1.2 Jaccard Index

Der Jaccard Index wird in der Statistik dazu genutzt, Ähnlichkeiten zwischen zwei Mengen zu erkennen. Er berechnet sich aus dem Betrag der Schnittmenge der beiden Mengen, geteilt durch den Betrag der Vereinigungsmenge.

$$Jaccard = \frac{|Label \cap Referenz|}{|Label \cup Referenz|}$$

Will man für das Beispiel aus Abschnitt 5.2.1.1 den Jaccard-Index berechnen, betrachtet man folgende Mengen:

$$\begin{aligned} Label \cap Referenz &= \{gabriel, physik\} \\ Label \cup Referenz &= \{gabriel, physik, deutsch, daniel, danzig\} \end{aligned}$$

Wie man in dem Beispiel sieht, sind zwei der insgesamt fünf verschiedenen Terme in beiden Labels vorhanden. Der Jaccard Index beträgt also 0,4.

5.2.1.3 Kosinus-Ähnlichkeit

Ein weiteres Verfahren für die Berechnung der Ähnlichkeit zwischen einem generierten Label und dem passenden Referenzlabel, ist die Kosinus-Ähnlichkeit [ZM98]. Hierfür müssen die zu vergleichenden Labels in das Vektorraum-Modell überführt werden. Wie im Bereich des Information Retrievals üblich, werden dafür beide Labels als mehrdimensionale Vektoren durch ihre Termhäufigkeiten repräsentiert. Die Kosinus-Ähnlichkeit berechnet sich aus dem Kosinus des Winkels zwischen den beiden Vektoren.

$$\cos\theta = \frac{label_1 \cdot label_2}{||l_1||||l_2||}$$

Tabelle 5.1 zeigt den Vektorraum für die zwei aus den letzten Abschnitten bekannten Label „Gabriel Physiker Daniel Danzig“ und „Gabriel Fahrenheit, ein deutscher Physiker“. Jede Zeile in der Tabelle entspricht einem Vektor. Der Kosinus des Winkels zwischen diesen beiden Vektoren und somit auch die Kosinus-Ähnlichkeit zwischen den beiden Labels beträgt 0,55.

	gabriel	physik	deutsch	daniel	danzig
Referenz	1	1	1	0	0
Label	1	1	0	1	1

Tabelle 5.1: Vektorraum der Labels „Gabriel Physiker Daniel Danzig“ und „Gabriel Fahrenheit, ein deutscher Physiker“. Die Terme des Überthemas (in diesem Beispiel: *Fahrenheit*) wurden nicht betrachtet.

5.2.2 Semantische Evaluation

Für die Evaluation der Clusterlabeling-Verfahren wurde im letzten Abschnitt verschiedene klassische Maße vorgestellt. Alle Verfahren vergleichen, welche Terme in dem generierten Label sowie in dem passenden Referenzlabel vorkommen und berechnen anhand verschiedener Betrachtungsweisen einen Ähnlichkeitswert. Je höher dieser ist, umso höher ist die Ähnlichkeit und umso besser bewerten wir das Clusterlabeling-Verfahren. Ein Problem dabei ist, dass die klassischen Evaluationsverfahren nicht alle guten Clusterlabels auch gut bewerten. Nehmen wir beispielsweise ein generiertes Label, das den Inhalt des Referenzlabels mit anderen Worten ausdrückt. Mit den klassischen Evaluationsverfahren würde dieses schlecht bewertet werden, da generiertes Label und Referenzlabel wenig gleiche Terme enthalten.

Deutlich wird dies etwa in dem Beispiel in Tabelle 5.2. Hier wurden Clusterlabels für ein Cluster des Überthemas *Fahrenheit* erstellt. Das gelabelte Cluster enhält Dokumente über den gleichnamigen deutschen Physiker. In der Tabelle sind alle Clusterlabels sowie das Referenzlabel dargestellt. Fett gedruckte Wörter sind Terme des Referenzlabels, kursiv dargestellte Wörter sind entweder Stoppwörter oder im Überthema enthalten und werden bei der Berechnung der Evaluationsmaße ausgelassen. Die generierten Labels haben unterschiedliche Länge und enthalten jeweils zwei Terme des Referenzlabels. Das Referenzlabel enthält Iden vollständigen Namen des Physikers, seine Herkunft und seinen Beruf. Label 2 und Label 3 enthalten nur zwei der drei Informationen des Referenzlabels. Label 1 enthält alle Informationen des Referenzlabels, mit dem Unterschied, dass anstatt des Terms *Physiker* der allgemeinere Begriff *Wissenschaftler* zugeordnet wurde. Deshalb kann Label 1 als das beste der drei generierten Labels eingestuft werden, Label 2 als das zweitbeste, weil es sehr kurz ist und zwei der Informationen wiedergibt, und Label 3 als das schlechteste, weil es zusätzliche Terme enthält, die nicht den Inhalt des Clusters beschreiben.

Tabelle 5.3 zeigt die Evaluations-Maße der Labels aus Tabelle 5.2 verglichen mit dem Referenzlabel. Betrachtet man nur die klassischen Verfahren Jaccard-Index, F-Measure und Kosinus-Ähnlichkeit, so sieht man, dass das subjektiv

Label	Labeltext
Referenz	Gabriel *Fahrenheit*, *ein* deutscher Physiker
Label1	**Gabriel** *Fahrenheit* **deutscher** Wissenschaftler
Label2	**Gabriel** *Fahrenheit* **Physiker**
Label3	**Gabriel Physiker** Daniel Danzig

Tabelle 5.2: Beispiel für generierte Labels und Referenzlabels zu dem Überthema Fahrenheit. Die *kursiven* Terme werden in den Berechnungen der Evaluation ausgelassen, da diese entweder einem Stoppwort oder einem Term des Überthemas entsprechen. Die **fett gedruckten** Terme der generierten Labels (Label1-Label3) sind diejenigen, die auch im Referenzlabel enthalten sind.

favorisierte Label 1 nicht als das beste Label bewertet wurde. Diesen Platz nimmt hierbei Label 2 ein, da dieses sehr kurz ist und ausschließlich aus Termen des Referenzlabels besteht. Der Term *Wissenschaftler* in Label 1 ist zwar inhaltlich passend für die Dokumente des Clusters, fällt jedoch beim Mengenvergleich negativ ins Gewicht. Um auch Synonyme oder Oberbegriffe von Termen des Referenzlabels positiv zu bewerten, verwenden wir ein Verfahren, das die Ähnlichkeits-Analyse semantisch betrachtet.

Ein großes Problem bei den klassischen Verfahren ist die geringe Anzahl an Termen. Dadurch sind nur wenige semantisch zusammenhängende Terme enthalten, wie sie beispielsweise bei einem langen Text vorhanden sind. Der Vektorraum ist zu klein, um eine gute Grundlage für den Ähnlichkeitsvergleich zu haben. An dieser Stelle knüpft die *Explicit Semantic Analysis* [GM07] an. Hierbei wird ein semantischer Interpreter eingesetzt, der den Vektorraum der zu vergleichenden Labels mit Hilfe von Wikipedia-Artikeln semantisch erweitert.

In Abbildung 5.1 ist der Ablauf der Explicit Semantic Analysis schematisch dargestellt. Ausgehend von 10.000 zufällig gewählten Wikipedia-Artikeln wird ein Inverted Index gebaut, der jeden Term w auf eine Liste von so genannten *Konzepten* c abbildet. Unter Konzepten werden hier Text-Bausteine

	Mengenvergleichende Verfahren			
	F-Measure	Jaccard-Index	Cos-Ähnlichkeit	ESA erweitert
Label1	0,66	0,5	0,66	0,79
Label2	0,66	0,8	0,81	0,66
Label3	0,57	0,4	0,55	0,42

Tabelle 5.3: Evaluations-Werte der Labels aus Tabelle 5.2

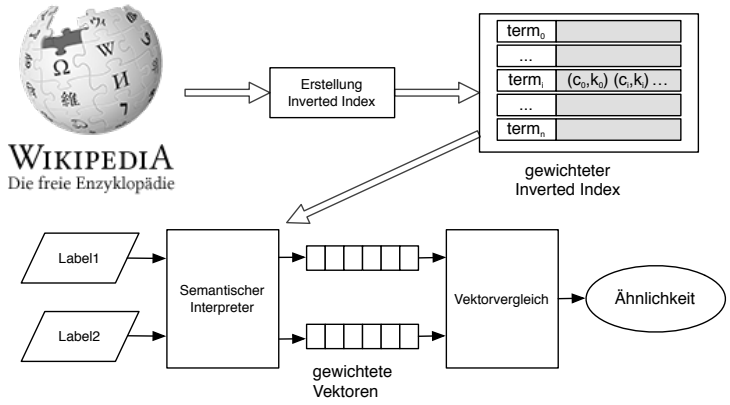


Abbildung 5.1: Ablaufschema der *Explicit Semantic Analysis*. Der Inverted Index von zufälligen Wikipedia-Artikel bildet die Grundlage für einen semantischen Interpreter. (Vgl. Abbildung 1 in [GM07])

verstanden, die Wissen organisieren [GM07, Abschnitt 1]. Wir verwenden bei der Explicit Semantic Analysis für ein Konzept einen Wikipedia-Artikel. Die Konzepte sind nach ihrer *Zugehörigkeit* k bezüglich des Terms w geordnet. Bekommt der semantische Interpreter ein Label T, so wandelt dieser zuerst dieses in $tf \cdot idf$-Vektorrepräsentation um, sodass für jeden Term w ein $tf \cdot idf$-Gewicht v vorhanden ist. Für jeden Term w aus T wird aus dem Inverted Index eine Liste von Konzepten c mit ihren dazugehörigen Gewicht k entnommen. Als Output liefert der semantische Interpreter einen Vektor, der für jedes Konzept ein kombiniertes Gewicht enthält: den semantischen Interpretationsvektor. Das kombinierte Gewicht eines Konzepts berechnet sich aus der Summe aller $tf \cdot idf$-Gewichte v von jedem enthaltenen Term w aus T, multipliziert mit dem Zugehörigkeits-Gewicht k:

$$Gewicht(c_i) = \sum_{w_j \in T} v_j \cdot k_i.$$

Die $tf \cdot idf$-Repräsentationen der zu vergleichenden Labels werden auf diese Weise semantisch erweitert. Statt Ähnlichkeitswerte kurzer Labels zu vergleichen, wird mit Hilfe der erhaltenen Interpretationsvektoren die Kosinus-Ähnlichkiet berechnet.

Die letzte Spalte der Tabelle 5.3 zeigt die Kosinus-Ähnlichkeit, welche mit Hilfe von Explicit Semantic Analysis erweitert wurde. Man sieht, dass das subjektiv favorisierte Label 1 den höchsten Ähnlichkeitswert erreicht hat, da hier der Term Wissenschaftler nun positiv in das Gewicht gefallen ist. Die zusätzlich enthaltenen Terme von Label 3 haben keinen semantischen Bezug zu

den Termen im Referenzlabel. Der Ähnlichkeitswert dieses Labels ist dadurch geringer als der der anderen Labels. Die semantische Erweiterung ist also ein nützliches Werkzeug, um den Textvergleich der Labels für die Evaluation zu verbessern.

5.3 Nutzer-Evaluation

Neben der maschinellen Evaluation mit Hilfe der klassischen Verfahren und semantischen Erweiterung führen wir zusätzlich eine Nutzer-Evaluation für die Bewertung der Clusterlabeling-Verfahren durch. Hierfür werden einer Gruppe von Nutzern jeweils gleichzeitig die generierten Labels, sowie das Referenzlabel für ein Cluster präsentiert. Die Aufgabe des Nutzers ist es aus ‚den generierten Labels das beste auszuwählen. Dies wird für die Cluster von 100 Unterthemen wiederholt. Das Verfahren, dessen Clusterlabels häufig ausgewählt werden, wird in der Auswertung der Nutzerstudie gut bewertet.

Abbildung 5.2 zeigt die Nutzeroberfläche, die der Proband in der Evaluation bedienen soll. Unter dem Referenzlabel (in Abbildung 5.2, Nutzerelement 1) werden drei generierte Labels als Auswahl (Nutzerelement 2) dargestellt. Diese wurden mit jeweils mit dem χ^2-Verfahren, der Weighted Centroid Covering, sowie dem Suchanfragen-Rekonstruktions-Verfahren erstellt. Mit Hilfe von Nutzerelement 3 in Abblidung 5.2 kann der Proband seinen Fortschritt verfolgen.

Um eine bessere Lesbarkeit zu garantieren, werden die generierten Labels einem Postprocessing unterzogen. Dabei werden zuerst Wiederholungen von Termen mit gleichem Wortstamm entfernt. Eine solche Wiederholung, beispielsweise das Auftreten eines Terms in Singular- und Pluralform, führt zu einer Verlängerung des Clusterlabels, ohne dessen Qualität zu verbessern. Das Herausfiltern solcher Terme macht das Label kürzer, klarer und für den Probanden besser lesbar. Ein weiteres Problem, das die Lesbarkeit erschwert, ist die Reihenfolge der Labelterme. Nachdem das Clusterlabeling-Verfahren ein Clusterlabel generiert hat, liegen die Worte meist nach ihrem Gewicht geordnet vor. Zusammengehörige Terme können dabei in falscher Reihenfolge, oder gar durch andere Terme getrennt, auftreten. Generiert beispielsweise ein Clusterlabeling-Verfahren das Clusterlabel „york city highway new“, so steht der Term *york* an erster Stelle, weil diesem das höchste Termgewicht zugeordnet wurde. Herauszufinden, dass das Label wohl Dokumente über den „new york city highway“ betitelt, ist zwar nicht schwer, kostet aber dem Nutzer trotzdem eine zusätzliche Anstrengung. Um die Lesbarkeit zu erhöhen, werden alle Terme mit Hilfe der *Netspeak-Wortsuchmaschine*[3] zu einer sinnvollen Phrase geordnet. Hierbei werden alle Terme an diesen Webservice übermittelt, mit dem Befehl die am

[3] http://www.netspeak.org/

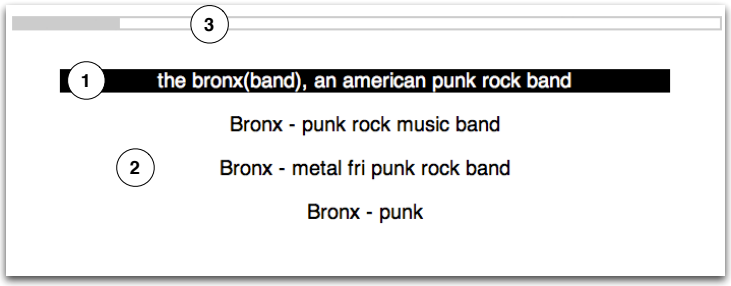


Abbildung 5.2: Interface der Nutzerstudie: Der Proband bekommt zu jedem Referenzlabel (1) drei generierte Labels zur Auswahl (2). Der Fortschrittsbalken (3) zeigt dem Probanden, wie weit er in der Nutzerstudie fortgeschritten ist.

häufigsten verwendete Reihenfolge dieser Terme zu ermitteln. Netspeak vergleicht anhand des Google-N-Gram Korpus verschiedene Schreibweisen und gibt Statistiken über diese zurück. Die am häufigsten verwendete Reihenfolge der Labelterme wird dann für das Clusterlabel verwendet. Findet Netspeak für die Terme des Labels keine Reihenfolge, so wird wird mit einer Teilmenge der Labelterme eine Reihenfolge gesucht. Die ausgelassenen Terme werden dann an die geordneten Terme angehängt.

5.4 Ergebnisdiskussion

Für die vergleichende Evaluation der Clusterlabeling-Verfahren werden jeweils mit Hilfe der Suchanfragen-Rekonstruktion (Abschnitt 4.2), χ^2-Clusterlabeling (Abschnitt 4.1.1) und Weighted Centroid Covering (Abschnitt 4.1.2) Clusterlabels für die 791 Cluster der 44 Überthemen des auf dem Ambient Dataset basierenden Dokumentcluster-Korpus (siehe Abschnitt 5.1) erstellt. Für die Konstante k, die bei den Labeling-Verfahren χ^2 und Weighted Centroid Covering angibt, wie lang jedes generierte Clusterlabel sein soll, wurde die durchschnittliche Länge der mit dem Suchanfragen-Rekonstruktions-Verfahren erstellten Label gewählt. Das bedeutet eine Label-Länge von $k = 5$.

Die generierten Labels werden jeweils mit den Referenzlabels der Cluster verglichen. Hierbei ermitteln wir Ähnlichkeitswerte einerseits mit Hilfe des F-Measure (Abschnitt 5.2.1.1), des Jaccard-Index (Abschnitt 5.2.1.2), der Kosinus-Ähnlichkeit (Abschnitt 5.2.1.3) und der semantisch erweiterten Kosinus-Ähnlichkeit (Abschnitt 5.2.2) und zusätzlich mit Hilfe von Nutzerbewertungen (Abschnitt 5.3). Generiert ein Clusterlabeling-Verfahren Labels mit hohen Ähnlichkeitswerten, so handelt es sich um ein gutes Labeling-Verfahren.

	Rekonst.	χ^2	WCC
F-Measure	0,103	0,137	0,056
Jaccard-Index	0,051	0,068	0,028
Cos-Ähnlichkeit	0,367	0,352	0,188
ESA erweitert	0,443	0,434	0,311

Tabelle 5.4: Durchschnittliche Ergebniswerte der maschinellen Evaluation

5.4.1 Ergebnis maschineller Verfahren

Tabelle 5.4 zeigt die durchschnittlichen Ähnlichkeitswerte der durch Rekonsturktionsverfahren, χ^2-Labeling und Weighted Centroid Covering (WCC) generierten Clusterlabels. In Tabelle 5.5 werden die Ähnlichkeitswerte der Clusterlabeling-Verfahren direkt gegenüber gestellt. Ist der Ähnlichkeitswert von Verfahren 1 in der ersten Spalte der Tabelle größer als der von Verfahren 2 in der zweiten Spalte, so wird dies durch einen Pfeil nach oben (↑) dargestellt. Ein Doppelpfeil symbolisiert dabei, ob der Unterschied statistisch signifikant ist. Für die Bestimmung der statistischen Signifikanz verwenden wir den *Student's T-Test*, da es sich bei den Ähnlichkeitswerten um nicht normalverteilte Daten handelt und die Standard-Abweichung nicht bekannt ist. Smucker et al. bewerten den Students's T-Test als geeignet für Evaluationen im Bereich des Information Retrieval [SAC07].

Wie man in diesen Darstellungen sieht, besitzt das Weighted Centroid Covering Verfahren mit statistischer Signifikanz bei allen Ähnlichkeitsmaßen die niedrigsten Ähnlichkeitswerte aller Clusterlabeling-Verfahren. Da dieses Verfahren nur clusterintern Labels generiert und dabei keine differenzierenden Terme ermittelt werden (vgl. Abschnitt 4.1.2), war ein solches Ergebnis zu erwarten. Interessanter ist eher der Vergleich des in dieser Arbeit vorgestellten Suchanfragen-Rekonstruktions-Verfahren (abgekürzt: *Rekonst.*) und des χ^2-Labelings. Die Ähnlichkeitswerte dieser beiden Verfahren zeigen keinen deutlichen Unterschied. Bei der Kosinus-Ähnlichkeit sowie der Kosinus-Ähnlichkeit mit semantisch erweiternden Vektoren hat das Rekonstruktions-Labeling durchschnittlich höhere Werte als das χ^2-Labeling, wobei der Unterschied nicht signifikant ist. Für die Ähnlichkeitsmaße F-Measure und Jaccard-Index besitzt das χ^2-Labeling signifikant höhere Werte als das Rekonstruktions-Verfahren.

Hinter beiden Verfahren steckt grundsätzlich die gleiche Idee, um Clusterlabels zu generieren. Es werden alle Terme der Dokumente eines Clusters darauf geprüft, ob diese gut das Cluster von den benachbarten Clustern des Clusterings differenzieren. Es ist deshalb nachvollziehbar, dass beide Verfahren

Verfahren1	Verfahren2	F-Measure	Jaccard	Cos-Ähn.	ESA erw.
Rekonst.	χ^2	$\Downarrow$	$\Downarrow$	$\uparrow$	$\uparrow$
	WCC	$\Uparrow$	$\Uparrow$	$\Uparrow$	$\Uparrow$
χ^2	Rekonst.	$\Uparrow$	$\Uparrow$	$\downarrow$	$\downarrow$
	WCC	$\Uparrow$	$\Uparrow$	$\Uparrow$	$\Uparrow$
WCC	χ^2	$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$
	Rekonst	$\Downarrow$	$\Downarrow$	$\Downarrow$	$\Downarrow$

Tabelle 5.5: Vergleich der Verfahren untereinander. $\uparrow$ bedeutet, dass Verfahren 1 einen höheren Ähnlichkeitswert als Verfahren 2 hat. $\Downarrow$ bedeutet, dass Verfahren 1 einen *signifikant* niedrigeren Ähnlichkeitswert als Verfahren 2 hat.

ähnliche Labels generieren. Um Stärken und Schwächen beider Verfahren zu analysieren, betrachten wir die Ähnlichkeitswerte im Detail.

Vergleich des Rekonstruktion-Verfahren mit χ^2-Labeling

Für einen detaillierteren Vergleich des Rekonstruktion-Verfahrens mit dem χ^2-Labeling anhand der maschinell ermittelten Ähnlichkeitswerte, werden diejenigen Labels näher betrachtet, bei denen das eine Label viel besser bewertet wurde, als das andere. In Tabelle 5.6 findet man Beispiele für Labels, bei denen die semantische erweiterte Kosinus-Ähnlichkeitswerte eine hohe Differenz ergibt. Die ersten drei Labels in Tabelle 5.6 sind Beispiele, bei denen das durch χ^2-Labeling erstellte Label einen viel höheren Ähnlichkeitswert aufweist als das durch Rekonstruktions-Labeling erstellte Label. An dieser Stelle sieht man eine große Schwachstelle des Rekonstruktions-Labelings: die Länge des Clusterlabels. Zwar enthalten diese schlecht bewerteten Labels relativ viele Terme, die zu dem Referenzlabel passen (in der Darstellung fett gedruckt), jedoch sind diese Labels zu kurz (etwa 27% der Labels bestehen nur aus ein

	Referenzlabel	χ^2-Labeling	Rekonstruktions-Verfahren
1	The Bronx(band), an American punk rock band	**punk rock** music **band** bands	**punk**
2	Bronx Zoo	city **bronx zoo zoos** show	**zoo bronx** exhibits wildlife tickets conservation animals
3	De Havilland Hornet, an aircraft	**aircraft havilland** de home mosquito	**havilland**
4	Trapped in a Purple Haze, a TV movie	time **trapped** jobeth people carly	**trapped purple haze** jonathan **movies**
5	James Oppenheim (1882–1932), US poet author and editor	world free history art **oppenheim**	**james poet** novelist wikipedia
6	The Wolseley Hornet	**wolseley** world car riley bmc	**wolseley hornet**

Tabelle 5.6: Beispiele für generierte Labels bei denen die Werte der semantisch erweiterten Kosinus-Ähnlichkeit einen großen Unterschied haben.

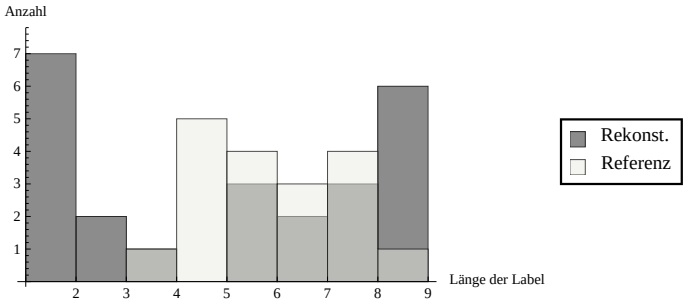


Abbildung 5.3: Histogrammm der Labellängen der durch das Rekonstruktions-Verfahren generierten Labels, welche mit der semantisch erweiterten Kosinus-Ähnlichkeit viel schlechter bewertet wurden als die durch χ^2-Labeling generierten Labels.

oder zwei Wörtern) oder zu lang (etwa ein Drittel der Labels hat eine Länge von 8 und mehr Wörtern). Dies wirkt sich negativ auf die Ähnlichkeitsmaße aus, da die Referenzlabel stoppwortgefiltert meist aus 4-6 Wörtern bestehen.

In dem Histogramm in Abbildung 5.3 sind die Häufigkeiten der Wortlängen der Clusterlabels dargestellt, bei denen die durch das Rekonstruktions-Verfahren erstellten Labels viel schlechter bewertet wurden als die Labels, die durch χ^2-Labeling erstellten wurden. Hier wird bestätigt, was beim Betrachten der Beispiele 1-3 in Tabelle 5.6 vermutet werden kann: es befinden sich bei den schlecht bewerteten Labels des Rekonstruktions-Verfahrens wenige in dem Bereich der Wortanzahl 4-6, also der häufigsten Längen der Referenzlabels.

Die Beispiele 4-6 in der Tabelle 5.5 zeigen Labels, bei denen das Rekonstruktions-Verfahren deutlich besser als das χ^2-Labeling bewertet wurde. Hier ist kein großer Unterschied in der Länge der Labels auffällig. Die χ^2-Label wurden anscheinend deshalb schlechter bewertet, weil diese nur wenige Terme enthalten, die im Referenzlabel vorkommen oder die semantisch zum Referenzlabel passen.

An diesen Beispielen wird deutlich, dass das in dieser Arbeit entwickelte Suchanfragen-Rekonstruktions-Verfahren viele *gute* Terme den Clusterlabels zuordnet, diese jedoch häufig sehr kurz oder sehr lang sind. Zur Verbesserung des Clusterlabeling-Verfahrens in zukünftiger Arbeit könnte diese Schwachstelle beim Compositing der Clusterlabels näher analysiert und verbessert werden.

5.4.2 Ergebnis der Nutzerstudie

Die Nutzerstudie umfasst die Clusterlabel-Bewertungen von 29 verschiedenen Probanden. Für die 100 Unterthemen, deren generierte Clusterlabels dem Nutzer präsentiert werden (vgl. Abschnitt 5.3), wurden die Themen ausgewählt, für deren generierte Labels durchschnittlich die höchsten Ähnlichkeitswerte

Verfahren	Klicks (relativ)	Gewinner
χ^2-Labeling	1084 (0,45)	53
Rekonst.	936 (0,39)	36
WCC	380 (0,16)	11
Gesamt	2400	100

Tabelle 5.7: Das Ergebnis der Nutzerstudie zeigt, wie häufig ein Verfahren insgesamt gewählt wurde (eine Wahl entspricht einem Klick) und wie oft das Verfahren das häufigste für ein Thema war (das am häufigsten gewählte Thema entspricht dem Gewinner)

bestimmt wurden. Für einen Durchlauf benötigten die Probanden ungefähr 15-30 Minuten.

Aufgrund des einfachen Versuchaufbaus war es möglich, die Nutzerstudie mit Hilfe einer Web-Anwendung umzusetzen. So konnten die Probanden den Versuch online, ohne Aufsicht und anonym, durchführen. Diejenigen Probanden, die nicht zu allen 100 Clustern eine Bewertung abgaben und die Nutzerstudie abgebrochen haben, werden in der Auswertung nicht beachtet. Von den 29 Probanden haben 23 den Test vollständig durchgeführt.

Tabelle 5.7 zeigt das Ergebnis der Nutzerstudie. Die zweite Spalte zeigt, wie häufig ein Clusterlabeling-Verfahren insgesamt gewählt wurde. Am häufigsten (45% aller Klicks) wurden die durch χ^2-Labeling generierten Clusterlabels gewählt. Labels des Rekonstruktions-Verfahrens wurden am zweithäufigsten angeklickt (bei 39 % aller Klicks). Nur bei den wenigsten Clustern wurden Labels des Weighted Centroid Covering-Verfahrens ausgewählt (16% aller Klicks). Das Verfahren, das am häufigsten für ein Thema gewählt wurde, ist das „gewinnende“ Verfahren, oder der *Gewinner* eines Themas. Die dritte Spalte von Tabelle 5.7 zeigt, wie häufig ein Verfahren Gewinner eines Themas war. Zwar ist die Reihenfolge des Rankings gleich (χ^2-Labeling, das beste Verfahren, das Rekonstruktion-Verfahren das zweitbeste, usw.), jedoch sind die Abstände viel größer. Der Vorsprung des χ^2-Labeling-Verfahrens gegenüber dem Rekonstruktions-Verfahren ist von 6 auf 17 Prozentpunkte angewachsen. Das deutet darauf hin, dass einige „Gewinne“ des χ^2-Labelings nur sehr knapp waren und es anscheinend bei einigen Themen unter den Nutzern keinen eindeutigen Sieger gab.

Eindeutigkeit der Entscheidungen

Anhand der Ergebnisse lassen sich die Themen bestimmen, bei denen ein Clusterlabeling-Verfahren eindeutig als das Beste bewertet wurde. In diesem Fall haben ein Großteil aller Nutzer das Thema eines bestimmten Verfahrens

gewählt, welches dadurch mit großem Vorsprung gewinnt. Wurden für ein Thema die Labels der Verfahren etwa gleich oft gewählt, bzw. hat ein Label mit nur kleinem Abstand gewonnen, so handelt es sich um ein Thema mit unklarem Ergebnis. Die Klassifizierung der Eindeutigkeit der Entscheidung lässt sich anhand der *korrigierten Varianz* der Entscheidung zu einem Cluster bestimmen. Dafür wird für jedes Thema die *Entscheidungsliste* betrachtet. Haben für ein Thema *top* beispielsweise von zehn Probanden fünf das Verfahren 1, zwei Verfahren 2 und drei Verfahren 3 gewählt, so sieht die Entscheidungsliste entsprechend wie folgt aus:

$$top = \{1, 1, 2, 1, 1, 3, 1, 3, 2, 2\}.$$

Aus dieser Entscheidungsliste werden jeweils für jedes Verfahren der Anteil bestimmt. In dem Beispiel wurde Verfahren 1 von 50% der Probanden gewählt und bekommt den Wert 0,5, Verfahren 2 erhält den Wert 0,3 und Verfahren 3 den Wert 0,2. Die korrigierte Varianz dieses Themas wird mit Hilfe folgender Formel berechnet (die eingesetzten Werte sind für das Beispielthema):

$$\begin{aligned} Varianz &= \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2 \\ &= \frac{1}{2}\cdot((0,5-\frac{1}{3})^2 + (0,3-\frac{1}{3})^2 + (0,2-\frac{1}{3})^2) = 0,02333. \end{aligned}$$

Dabei entspricht n der Länge der Entscheidungsliste, x dem Anteil des Verfahrens i und μ dem Mittelwert aller x:

$$\begin{aligned} \mu &= \frac{1}{n}\sum_{i=1}^{n} x_i \\ &= \frac{1}{3}\cdot(0,5+0,3+0,2) = \frac{1}{3}. \end{aligned}$$

Je höher die Varianz eines Themas, umso höher ist die Eindeutigkeit der Entscheidungen.[4] Für die Einteilung der Varianzen in die Klassen „eindeutig“ und „unklar“ werden die Varianzen aller Themen in 3 Teile geclustert. Hierbei werden die Werte zu einer Gruppe zusammengefasst, die eine geringe Differenz zueinander haben. So ergibt sich die Gruppe der niedrigen Varianzen (nicht eindeutige, unklare Themen), der hohen Varianzen (eindeutige Themen) und die Gruppe mit mittleren Varianzen. In Tabelle 5.8 sind die Varianz-Bereiche der Eindeutigkeitsklassen aufgeführt. Wie man hier sieht, haben 41% der Themen einen sehr niedrigen Varianzwert. Bei diesen Themen konnte also kein

[4] Anstatt der korrigierten Varianz kann man diese Untersuchung auch mit der *Entropie* durchführen und erhält ein sehr ähnliches Ergebnis. Hierbei gilt: je höher die Entropie, umso unklarer ist die Entscheidung für ein Thema

Klasse	Varianz-Bereich	#Themen	Anteil χ^2	Anteil Rekonst.	Anteil WCC
eindeutig	0,19 - 0,33	27	0,59 (0,16)	0,41 (0,11)	0,04 (0,01)
unklar	0 - 0,09	41	0,56 (0,23)	0,26 (0,11)	0,17 (0,70)
sonstige	0,09 - 0,19	32	0,43 (0,14)	0,43 (0,14)	0,09 (0,03)
gesamt	0 - 0,333	100	0,53	0,36	0,11
nicht unklar	0,096-0,333	59	0,50	0,42	0,07

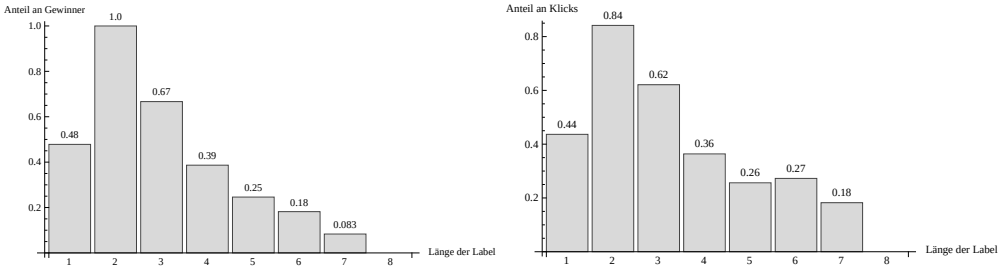
Tabelle 5.8: Einteilung der Themen bezüglich ihrer Eindeutigkeit in der Entscheidung.

klarer Sieger bestimmt werden. In diese Klasse fallen Themen, bei denen die Clusterlabeling-Verfahren ähnlich starke bzw. schwache Labels generiert haben. Bei 27% der Themen sind die Varianzwerte relativ hoch. Bei diesen Themen konnten sich die Nutzer auf einen klaren Favoriten festlegen. Ebenfalls in Tabelle 5.8 ist die Zusammensetzung der gewinnenden Labels dargestellt. 56% der unklar bewerteten Themen haben Labels des χ^2-Verfahrens gewonnen und nur 26 % die Labels des Rekonstruktions-Verfahrens. Ignoriert man die unklar entschiedenen Themen in der Bewertung, so verringert sich der Vorsprung des χ^2-Labeling gegenüber dem Rekonstruktions-Verfahren von 17 Prozentpunkte auf 8 Prozentpunkte.

Einfluss der Labellänge

Bei der Evaluierung mit den maschinell bestimmten Ähnlichkeitsmaßen haben wir festgestellt, dass ein großer Schwachpunkt des Rekonstruktions-Verfahrens die Länge des generierten Labels ist. Mit Hilfe der Nutzerstudie können wir überprüfen, ob sich die Nutzer tatsächlich an sehr langen oder sehr kurzen Labels stören. In dem Diagramm in Abbildung 5.4 ist dargestellt, wie hoch der Anteil an Gewinnerlabels bei den Labels einer bestimmten Länge ist. Man kann diese Labels in 3 Kategorien unterteilen: die sehr kurzen Labels mit einer Wortanzahl von 1 bis 3 Wörtern, die Labels mit einer mittleren Länge von 4 bis 6 Wörtern, die der Durchschnittslänge aller Labels entspricht, und die sehr langen Labels mit einer Wortanzahl von mehr als 6 Wörtern. Vergleicht man diese Kategorien untereinander, so sieht man, dass die kurzen Labels den größten Anteil an Gewinnern haben, gefolgt von den Labels mit mittlerer Länge. Anteilig die wenigsten Gewinner sind unter den sehr langen Labels. So sind beispielsweise nur 2 der 24 Labels mit der Wortanzahl 7 auch Gewinnerlabels.

Diese Darstellung zeigt deutlich, dass wenn Nutzer bei der Evaluation sehr kurze Labels präsentiert bekommen, sie dazu neigen, diese unabhängig von der Länge des Referenzlabels zu favorisieren. Die „erfolgreichste“ Labellänge ist die Wortanzahl 2. Alle Labels dieser Länge sind Gewinnerlabels.



(a) Anteil der Gewinnerlabels (b) Anteil der Klicks

Abbildung 5.4: Beliebtheit bestimmter Labellängen in Abhängigkeit von ihrem Anteil von Gewinnerlabels (a) und Klicks (b).

Die Beliebtheit von kurzen Clusterlabels bei den Nutzern sieht man auch anhand der durchschnittlichen Labellänge der Gewinnerlabels. Diese liegt mit 3,75 Wörtern unter der durchschnittlichen Labellänge von 5 Wörtern.

Vergleich der maschinellen Evaluation mit der Nutzerstudie

In Tabelle 5.9 sind die Ergebnisse der maschinellen Verfahren (Jaccard-Index, F-Measure, Kosinus-Ähnlichkeit und ESA erweiterte Kosinus-Ähnlichkeit) und der Nutzerevaluation gegenübergestellt. Die Zahlen stellen dar, wie oft ein mit dem jeweiligen Verfahren generiertes Label mit dem jeweiligen Evaluationsverfahrens gewonnen hat. Ein Label eines Clusterlabeling-Verfahrens gewinnt bei der maschinellen Evaluation, wenn für dieses ein höherer Ähnlichkeitswert bestimmt wird als für die Labels der anderen Verfahren des gleichen Themas. Ein Label gewinnt bei der Nutzer-Evaluation, wenn dieses häufiger als die anderen generierten Labels für ein Thema gewählt wurde.

Bei der hier betrachteten Stichprobe, handelt es sich um die 100 Themen mit der höchsten durchschnittlichen ESA erweiterten Kosinus-Ähnlichkeit. Man sieht anhand der Werte in Tabelle 5.9, dass diese Auswahl an Themen in Bezug auf die Stärke der Verfahren nicht ganz ausgeglichen ist. So enthält die Stichprobe doppelt so viele Themen, bei denen Labels des χ^2-Labelings gewinnen, als Themen, bei denen durch das Rekonstruktions-Verfahren generierte Labels gewinnen. Diese Verteilung entspricht nicht der Verteilung aller 791 Themen,

Verfahren	Jaccard	F-Measure	Cos-Korr.	ESA erw.	Nutzer
χ^2	53	53	47	60	53
Rekonst.	35	35	42	30	36
WCC	12	12	11	10	11

Tabelle 5.9: Vergleich der Gewinnerzahlen zwischen den maschinellen Evaluationsverfahren und der Nutzerstudie.

Verfahren	gesamt	eindeutig
Jaccard	0,67	0,78
F-Measure	0,67	0,78
Cos-Korr.	0,76	0,96
ESA erw.	0,68	0,89

Tabelle 5.10: Anteil an gleichen Entscheidungen der maschinellen Evaluationsverfahren und Nutzerentscheidungen

bei denen (nach ESA erweiterte Kosinus-Ähnlichkeit) etwa gleich viele Themen von χ^2-Verfahren und Rekonstruktions-Verfahren gewinnen.[5] Das bedeutet, dass dem Nutzer mehr starke durch χ^2-Labeling generierte Labels präsentiert werden, als durch das Rekonstruktions-Verfahren generierte Labels.

Vergleicht man die Werte der verschiedenen Evaluationsverfahren in Tabelle 5.9 so sieht man, dass den Gewinnerzahlen zufolge die Evaluationen mit Hilfe des Jaccard-Index und des F-Measure der Nutzerstudie am ähnlichsten sind. Betrachtet man jedoch, wie häufig die Nutzer und das maschinelle Evaluationsverfahren gleich entschieden haben (siehe Tabelle 5.10), so ist die Kosinus-Ähnlichkeit der Nutzerentscheidung am ähnlichsten. Unterschiede zwischen maschinellen Verfahren liegen wahrscheinlich an der Eindeutigkeit der Themen. Betrachtet man den Anteil an gleichen Entscheidungen nur bei den eindeutigen Themen (siehe zweite Spalte in Tabelle 5.10) so sind die Werte generell höher. Als einen weiteren Grund für unterschiedliche Entscheidungen kann man die Länge des Labels nennen. Die durchschnittliche Länge der Label, bei denen unterschiedlich entschieden wurde, liegt mit 3,7 Worten unter der durchschnittlichen Labellänge der maschinell bestimmten Gewinnerlabels von 4,11.

[5] χ^2-Labeling: 325 Themen, Rekonstruktions-Verfahren: 297 Themen, WCC: 168 Themen

6 Zusammenfassung und Ausblick

Im Zentrum dieser Arbeit steht die Idee, Cluster-Dokumente mit Labels genauso zu verknüpfen, wie Suchmaschinen Anfragen mit Ergebnisdokumenten verknüpfen. Wir betrachten dabei die Ergebnisliste als das Cluster von ähnlichen Dokumenten und die Suchanfrage als passendes Clusterlabel. Das Clusterlabeling entspricht in unserer Betrachtungsweise also einem Verfahren, das aus einer Menge von ähnlichen Dokumenten eine mögliche Suchanfrage rekonstruiert.

Ein nützliches Hilfsmittel, um diese Idee umzusetzen, ist der Reverted Index. Es handelt sich hierbei um eine Datenstruktur, in der abgebildet wird, welche Dokumente sich mit welchen Suchworten finden lassen. So können wir, ausgehend von einer Menge von Dokumenten, einfach mögliche Suchterme bestimmen, diese mit Hilfe einer Funktion gewichten und anhand der geordneten Liste der Suchtermkandidaten eine Suchanfrage zusammenstellen, die alle Dokumente der Ausgangsmenge in der Ergebnisliste enthält.

Die Initialisierung des Reverted Index benötigt einige Zeit, da viele Suchanfragen verarbeitet werden müssen. Es ist jedoch möglich, diese Datenstruktur vorzuberechnen und serialisiert zu speichern. Ein weiterer Vorteil des Reverted Index ist dessen Unabhängigkeit von Retrieval-Modell und indiziertem Dokumentkorpus, da die Suchmaschine selbst nur als ein System genutzt wird, das auf einen bestimmten Input (eine Suchanfrage) einen bestimmten Output (die Ergebnisliste) liefert. Dadurch ist unser Verfahren zur Suchanfragen-Rekonstruktion mit allen modernen Suchmaschinen möglich. Eine tiefere Analyse von verschiedenen Retrieval-Modellen ist deshalb nicht nötig.

Um das Suchanfragen-Rekonstruktions-Verfahren für Clusterlabeling anzuwenden, erweitern wir dieses durch eine andere Auswahl und Gewichtung der Suchtermkandidaten. Das bedeutet, dass wir für ein Clusterlabel eine Suchanfrage rekonstruierten, deren Ergebnisliste alle Dokumente des Clusters und gleichzeitig keine Dokumente der benachbarten Cluster des Clusterings enthält. Beim Hinzufügen eines Terms zu einem Clusterlabel wägt das Verfahren also ab, wie viele Dokumente des zu labelnden Clusters und wie viele Dokumente der benachbarten Cluster in den Top n Positionen der Ergebnisliste auftauchen. Hierfür gewichten wir die Suchtermkandidaten, welche wir mit Hilfe des Reverted Index ermittelt haben, mit dem F-Measure.

Die Vorgehensweise, differenziert Terme aus den Clusterdokumenten zu extrahieren, nennt man *Differential Clusterlabeling*. Ein anderer Vertreter dieser Art ist das χ^2-Labeling. Ein Verfahren, das sich bei der Label-Generierung allein auf die Dokumente des Clusters beschränkt, nennt man *Cluster-Internal Labeling*.

Um zu ermitteln, wie gut unser Verfahren funktioniert und wie die Qualität im Vergleich zu anderen Clusterlabling-Verfahren ist, führen wir eine ausführliche Evaluation durch. Die Basis hierfür bildet ein selbst zusammengestellter, in Clusterings strukturierter Dokumetkorpus, der auf der Hierachie des Ambient-Datasets basiert. Für jedes Cluster des Korpus existiert eine Menge von Dokumenten und ein Referenzlabel.

Für die Evaluation generieren wir für jedes der 791 Cluster des Korpus ein Label mit Hilfe unseres Rekonstruktions-Verfahrens, dem χ^2-Labeling als Vertreter für das Differential Clusterlabeling und mit Hilfe des Weighted Centroid Coverings, das wir so angewendet haben, dass es nur clusterintern Labels generiert. Für jedes generierte Label bestimmen wir die Ähnlichkeit zum passenden Referenzlabel. Je größer dabei die Ähnlichkeit ist, umso besser hat das Verfahren gearbeitet. Den Ähnlichkeitsvergleich führen wir einerseits maschinell und andererseits von Menschenhand bewertet durch.

Bei der maschinellen Evaluation benutzen wir die klassischen Evaluationsverfahren Jaccard-Index, F-Measure und Kosinus-Ähnlichkeit. Zusätzlich verwenden wir zum ersten Mal ein Verfahren für die semantische Erweiterung der Kosinus-Ähnlichket. Bei dieser *Explicit Semantic Analysis* wird anhand von zufällig gewählten Wikipedia-Artikeln der auf Termhäufigkeit basierende Vektorraum semantisch erweitert. Der Ähnlichkeitsvergleich mit Hilfe der ESA erweiterten Kosinus-Ähnlichkeit ist eine neue Methode, Clusterlabels zu evaluieren.

Für die Nutzerstudie haben 23 Probanden für ein Cluster generierte Clusterlabels mit dem passenden Referenzlabel verglichen. Die Aufgabe der Probanden war es, unter den generierten Labels dasjenige auszuwählen, welches am besten zu dem Referenzlabel passt.

Alle Evaluationsverfahren haben das clusterintern arbeitende Weighted Centroid Covering als das schlechteste Verfahren eingestuft. Für die clusterdifferenzierenden Verfahren χ^2-Labeling und Rekonstruktions-Verfahren ermittelten wir höhere durchschnittliche Ähnlichkeitswerte. Hierbei schneidet bei Jaccard-Index, F-Measure und Nutzerstudie das χ^2-Labeling signifikant besser ab. Bei Kosinus-Ähnlichkeit und semantisch erweiterter Kosinus-Ähnlichkeit wurden für das Rekonstruktions-Verfahren höhere Ähnlichkeitswerte bestimmt.

Anhand einer genaueren Analyse der *gewinnenden* und *verlierenden* Clusterlabels, konnten wir Rückschlüsse auf Vor- und Nachteile unseres neuen Clusterlabeling-Verfahrens ziehen. Ein großer Vorteil des Rekonstruktion-Ver-

fahrens ist, dass die Länge des Labels nicht manuell bestimmt werden muss, sondern das Label automatisch so generiert wird, dass es gestellt an eine Suchmaschine alle Dokumente des Clusters in der Ergebnisliste enthält. Ein Problem hierbei ist, dass dadurch teilweise sehr lange Labels (von einer Länge von acht und mehr Worten) und teilweise sehr kurze Einwort-Labels erstellt werden. Gerade die niedrigen Ähnlichkeitswerte der langen Labels verschlechtern das Ergebnis der Evaluation für das Rekonstruktions-Verfahren. Die kurzen Labels bekommen bei der maschinellen Evaluation zwar auch niedrige Ähnlichkeitswerte, sind bei den Nutzern jedoch stark favorisiert.

Um das Rekonstruktions-Verfahren zu verbessern, ist es notwendig das Problem der Labellänge zu lösen. Hierfür wäre es möglich, die rekonstruierte Suchanfragen einem Postprocessing zu unterziehen, bei dem unnötige Terme, die die Zusammensetzung der Ergebnisliste nicht beeinflussen, entfernt werden.

Für die zukünftige Arbeit an diesem Thema ist es interessant, wie Clusterlabels in der Evaluation abschneiden, die mit anderen Suchmaschinen, als der in dieser Arbeit verwendeten ChatNoir-Suchmaschine, generiert werden. Die ChatNoir-Suchmaschine ermittelt die relevanten Dokumente anhand einer einfachen UND-Verknüpfung der Suchwörter. Mit einer Suchmaschine, die eine Suchanfrage anders verarbeitet (beispielsweise mit Synonymen sucht), lassen sich auch andere Suchanfragen und somit andere Clusterlabels konstruieren.

Des Weiteren ist der Versuchsaufbau der Nutzerstudie kritisch zu betrachten. Wir gestalten diesen in dieser Arbeit so, dass eine Vergleichbarkeit mit den Werten der maschinellen Evaluationsverfahren möglich ist: die Nutzer führen einen Ähnlichkeitsvergleich von generierten Labels mit Referenzlabels durch. Das Referenzlabel beschreibt zwar gut den Inhalt des Clusters, eignet sich aber aufgrund dessen Länge nicht immer als tatsächliches Clusterlabel. In einem möglichen alternativen Versuchsaufbau könnte ein Nutzer Dokumente eines Clusters betrachten und beurteilen, welches der generierten Labels am geeignetsten für dieses Cluster ist. Ein solches Vorgehen bewertet das Clusterlabeling-Verfahren unabhängig von einem Referenzlabel und wäre wertvoller für die Evaluation.

Zusätzlich sind weitere Anwendungsgebiete für die Suchanfragen-Rekonstruktion denkbar. Das Verfahren könnte beispielsweise dazu genutzt werden, für eine Dokumentmenge ähnliche Dokumente zu finden. So könnte man ausgehend von einer Dokumentmenge, die beispielsweise durch eine Recherche zusammengetragen wurde, mit Hilfe einer rekonstruierten Suchanfrage automatisch weitere Dokumente zu dem jeweiligen Recherche-Thema finden.

Literaturverzeichnis

[BCDG08] Francesco Bonchi, Carlos Castillo, Debora Donato und Aristides Gionis. Topical Query Decomposition. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, Seiten 52–60, New York, NY, USA, 2008. ACM.

[FT04] Bent Fuglede und Flemming Topsøe. Jensen-Shannon Divergence and Hilbert space Embedding. 2004.

[GM07] Evgeniy Gabrilovich und Shaul Markovitch. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, Seiten 1606–1611, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[HS10] Matthias Hagen und Benno Stein. Search Strategies for Keyword-based Queries. In A Min Tjoa und Roland Wagner (Hrsg.), *7th International Workshop on Text-Based Information Retrieval (TIR 10) at DEXA*, Seiten 37–41. IEEE, September 2010.

[JWG06] Chris Jordan, Carolyn Watters und Qigang Gao. Using Controlled Query Generation to Evaluate Blind Relevance Feedback Algorithms. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '06, Seiten 286–295, New York, NY, USA, 2006. ACM.

[MKG10] Markus Muhr, Roman Kern und Michael Granitzer. Analysis of Structural Relationships for Hierarchical Cluster Labeling. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, Seiten 178–185, New York, NY, USA, 2010. ACM.

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

[NC10] Roberto Navigli und Giuseppe Crisafulli. Inducing Word Senses to Improve Web Search Result Clustering. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, Seiten 116–126, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[PCG10] Jeremy Pickens, Matthew Cooper und Gene Golovchinsky. Reverted Indexing for Feedback and Expansion. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, Seiten 1049–1058, New York, NY, USA, 2010. ACM.

[SAC07] Mark D. Smucker, James Allan und Ben Carterette. A Comparison of Statistical Significance Tests for Information Retrieval. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, Seiten 623–632, New York, NY, USA, 2007. ACM.

[SGH11] Benno Stein, Tim Gollub und Dennis Hoppe. Beyond Precision@10: Clustering the Long Tail of Web Search Results. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, Seiten 2141–2144, New York, NY, USA, 2011. ACM.

[SM04] Benno Stein und Sven Meyer zu Eißen. Topic Identification: Framework and Application. In Klaus Tochtermann und Hermann Maurer (Hrsg.), *4th International Conference on Knowledge Management (I-KNOW 04)*, Journal of Universal Computer Science, Seiten 353–360, Graz, Austria, Juli 2004. Know-Center.

[TC11] Anil Turel und Fazli Can. A New Approach to Search Result Clustering and Labeling. In Mohamed Salem, Khaled Shaalan, Farhad Oroumchian, Azadeh Shakery und Halim Khelalfa (Hrsg.), *Information Retrieval Technology*, Band 7097 aus *Lecture Notes in Computer Science*, Seiten 283–292. Springer Berlin / Heidelberg, 2011.

[ZCT+04] Hugo Zaragoza, Nick Craswell, Michael Taylor, Suchi Saria und StephenRRobertson. Microsoft Cambridge at TREC-13: Web and HARD tracks. In *Proceedings of the 13th Text REtrieval Conference*, TREC '04, Gaithersburg, Maryland, USA, November 2004.

[ZM98] Justin Zobel und Alistair Moffat. Exploring the Similarity Space. Band 32, Seiten 18–34, New York, NY, USA, April 1998. ACM.