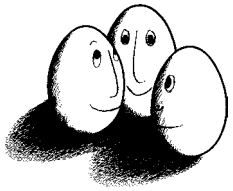
Diplomarbeit

Inkrementelles Clustering von Newsgroupartikeln

Sascha Hennig



Diplomarbeit
am Fachbereich Informatik
der Universität Dortmund

Freitag, 30. September 2005

Betreuer:

Prof. Dr. Katharina Morik
Dipl.-Inform. Michael Wurst

Danksagung

Ein herzliches Dankeschön an alle, die mir bei der Erstellung der vorliegenden Arbeit helfend zur Seite gestanden haben. Mein besonderer Dank gilt meinen beiden Betreuern Prof. Dr. Katharina Morik und Dipl.-Inform. Michael Wurst.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Seit Beginn der 90er Jahre hat das Internet einen zunehmenden Einfluss auf das Leben in unserer modernen Gesellschaft gewonnen. Aber nicht nur das Internet an sich hat einen festen und weiter wachsenden Stellenwert in der vor allem wirtschaftlichen Entwicklung der Gesellschaft. Auch die damit einhergehenden technisch geprägten Veränderungen, wie beispielsweise E-Mail-Kommunikation oder die Verwaltung großer Datenmengen mittels Datenbanken, verändern unsere Arbeits- und Lebensweise. Unabhängig und doch auch parallel zu diesen Entwicklungen wurde ein Begriff geprägt und in vielen Zusammenhängen verwendet: der Begriff der *Informationsgesellschaft*. Er soll die zunehmende Bedeutung von Information auf alle Lebensbereiche, besonders auf den Bereich der Arbeit, ausdrücken. Information wird in wirtschaftlicher Hinsicht häufig als der vierte Produktionsfaktor neben Arbeit, Boden und Kapital bezeichnet. Zuweilen ist von *Information* auch als ein immaterielles Gut oder Kapital die Rede. Der Begriff der Informationsgesellschaft und die technische Entwicklung sind eng miteinander verknüpft, und dies nicht nur aufgrund der Tatsache, dass beide Phänomene in der gleichen Zeit aufgetaucht sind. Vielmehr ist die Bedeutung des Gutes Information für die Gesellschaft erst durch die zunehmenden Möglichkeiten der Verfügbarkeit und der Verarbeitung, welche die technische Entwicklung bereitgestellt hat, in dem Maße gestiegen, dass die Charakterisierung als Informationsgesellschaft eine logische Konsequenz war.

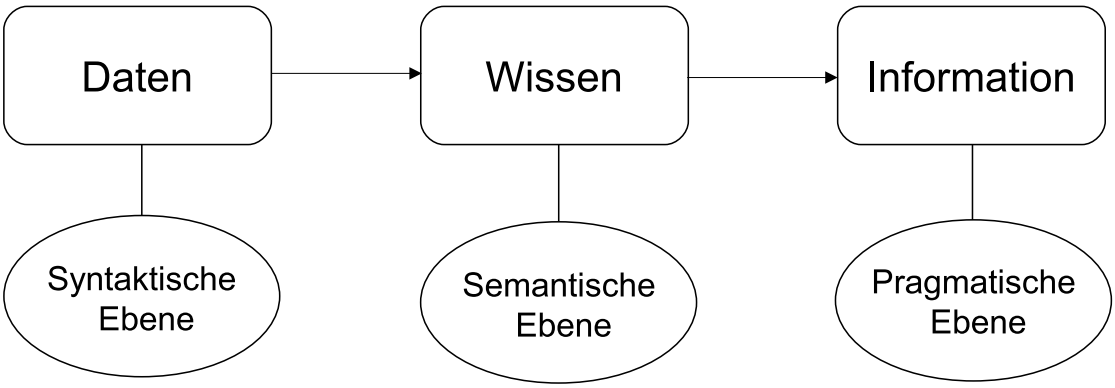


Abbildung 1.1: Daten - Wissen - Information

Doch was ist Information eigentlich? In der Informationswissenschaft wird der Begriff der Information mit zwei weiteren Begriffen verbunden: *Daten* und *Wissen*. Die Abbildung 1.1 veranschaulicht die Verbindung zwischen den drei Begriffen (siehe [Fuh03]).

Daten sind die symbolischen Repräsentationen von Sachverhalten. So ist beispielsweise der Anzeigewert von 25° Celsius auf einem digitalen Thermometer ein Datum[1]. Daten sind auf einer syntaktischen Ebene angesiedelt, wie in der Abbildung 1.1 zu sehen ist. In diesem Sinne wäre eine Datenbasis eine nackte Sammlung von Werten ohne Semantik.

[1] Der Begriff Daten ist der Plural von Datum (lat. Singular *datum*,Plural *data* für gegebenes).

Wird zwischen den einzelnen Daten eine Beziehung hergestellt, also Semantik hinzugefügt, so erhält man Wissen. Die Verknüpfung zwischen dem Wert *25° Celsius* mit der Stadt Dortmund und dem (Zeit-)Datum *09.09.2005 um 14 Uhr* ist beispielsweise Wissen. Information wiederum wird auf der pragmatischen Ebene definiert. Kuhlen definiert Information in [Kuh90] so: „Information ist die Teilmenge von Wissen, die von jemandem in einer konkreten Situation zur Lösung von Problemen benötigt wird." Anders ausgedrückt, wenn Wissen angewendet wird, so wird es in Information transformiert. Will nun jemand wissen, wie die Temperatur in Dortmund am 09.09.2005 um 14 Uhr war, so ist *25° Celsius* für ihn eine Information. Die Beziehung zwischen Wissen und Information lässt sich schlagwortartig auch wie folgt ausdrücken: *Information ist Wissen in Aktion.* Hier wird auch ein weiterer wesentlicher Unterschied zwischen Wissen und Information angedeutet: Information ist flüchtig und Wissen beständig. Am besten lässt sich dies anhand von Datenbanksystemen verdeutlichen. In einem Datenbanksystem werden nicht nur die Daten gehalten, es wird auch teilweise die Semantik des betreffenden Gebietes im System modelliert. Das heißt, ein Datenbanksystem enthält Wissen. Stellt nun ein Anwender eine Anfrage an das Datenbanksystem, um einen bestimmten Sachverhalt zu klären, so liefert ihm die Antwort des Systems eine Information. Ist der Sachverhalt für den Anwender nicht mehr relevant, so ist das Wissen, das er von dem Datenbanksystem erhalten hat, keine Information mehr, da er es nicht weiter anwenden kann.

Im Internet hat sich eine spezielle Form des Informations- und Meinungsaustausches herausgebildet. Webforen und Newsgroups sind eine moderne Form der *schwarzen Bretter*. Allerdings bezeichnen die beiden Begriffe *Webforum* und *Newsgroup*, technisch unterschiedliche Ausprägungen derselben inhaltlichen Zielsetzung. Ein Webforum, häufig auch einfach als Forum bezeichnet, ist ein Diskussionsforum auf einer *Website*. Der Zugriff auf ein Webforum erfolgt demnach über einen Internetbrowser. Eine Newsgroup ist ebenfalls ein virtuelles Diskussionsforum. Für den Zugriff auf eine Newsgroup benötigt der Anwender jedoch ein spezielles Programm: einen sogenannten *Newsreader*. Beides sind Diskussionsforen im Internet[2]. In ihnen werden zum Beispiel Fragen zu technischen Problemen erörtert, politische Themen diskutiert oder Neuigkeiten zu verschiedenen Sportarten ausgetauscht. Dabei ist ihre Funktionsweise zeitlich linear strukturiert. Der Besucher einer Newsgroup hat die Möglichkeit, einen eigenen Beitrag an das digitale schwarze Brett zu hängen oder sich die bereits befestigten Beiträge anzuschauen. Dabei gibt es für die Beiträge innerhalb eines Forums so gut wie keine Ordnung, wenn man einmal von der zeitlichen Ordnung absieht. Es gibt häufig Newsgroups, die sich einem großen Themengebiet widmen und sich gleichzeitig in kleinere Untergebiete oder Facettierungen des Oberthemas verzweigen. Diese Ordnung ist in der Regel nur sehr grob. Des Weiteren werden Beiträge nur zeitlich befristet in einer Newsgroup gehalten und sind nach einer bestimmten Zeit für Besucher des Forums nicht mehr zu erreichen.

Um die Abläufe in einem Forum oder einer Newsgroup besser beschreiben zu können, werden dieser Stelle einige Begriffe definiert.

Definition 1.0.1 (**Eröffnungsbeitrag** und **Kommentar**)**.** Das Befestigen eines neuen Beitrages, der nicht Bezug auf einen anderen Beitrag in einem Forum nimmt, ist das *Eröffnen einer neuen Diskussion* und wird als *Eröffnungsbeitrag* bezeichnet. Ein *Kommentar* ist der Antwortbeitrag auf einen bestehenden Beitrag. Ein Kommentar kann als

[2] Die beiden Begriffe *Forum* und *Newsgroup*, werden in dieser Arbeit analog verwendet

Antwortbeitrag auf einen Eröffnungsbeitrag oder auch als Kommentar auf einen anderen Kommentar in das Forum gestellt werden.

Definition 1.0.2 (**Artikel**). Ein *Artikel* in einem Forum bezeichnet eine Menge von Beiträgen, die sich direkt oder indirekt auf einen Eröffnungsbeitrag beziehen. *Direkt* auf einen Beitrag bezogen ist ein Kommentar genau zu diesem Beitrag. Indirekt auf einen Beitrag A bezogen ist ein Kommentar, der über einen weiteren Kommentar oder eine Reihe von Kommentaren mit dem Beitrag A verbunden ist.

Wie stellen sich nun die Beiträge und Artikel eines Forums in dem oben gezeigten Schema *Daten - Wissen - Information* dar? Angenommen sei ein Besucher eines technischen Forums, der seit längerer Zeit über einem Problem brütet. Er beschließt, das Problem in dem Forum zu beschreiben und eröffnet eine Diskussion. Seine Frage ist eine, die schon häufiger in verschiedenen Variation aufgekommen ist, und so erhält er schnell Antwort in Form eines Kommentars. Angenommen, die Antwort ist auch die Lösung seines Problems, so kann er dies nutzen und endlich mit seiner Arbeit fortfahren.

Betrachtet man den eben geschilderten Sachverhalt aus der Sicht der Informationswissenschaft, so wurden Daten in Form von Worten an einem zentralen Ort, dem Forum, abgelegt. Die Aneinanderreihung der Worte zu Sätzen, die das Problem beschreiben, stellen zuallererst Wissen dar. Mit dem Lesen des Beitrages wurde er zu der Information über das Problem, mit deren Hilfe eine Lösung formuliert werden konnte. Der Beitrag, der die Lösung enthält, ist auch in erster Linie Wissen und erst die Anwendung dieses Wissens macht eine Information daraus. Ist nun der Austausch abgeschlossen, handelt es sich bei dem Artikel auch weiterhin um Wissen. Und wie am Beginn des Beispieles erwähnt, handelt es sich bei dem Problem des Besuchers um ein sehr verbreitetes. Der Beitrag stellt also auch für andere Besucher des Forums eine Information dar. Doch wenn sie nicht wissen, wo sie nach dieser Information suchen sollen, ist die Existenz des Beitrages für sie wertlos und das Wissen bleibt ungenutzt.

Die meisten Newsgroups und Foren stellen keine weiteren Zugriffsmöglichkeiten[3] auf die Artikel zur Verfügung. Zuweilen werden FAQs (*Frequently Ask Questions*) bereitgestellt. Aber selbst diese sind häufig nur in Form von Frage-Antwort-Listen aufgebaut und bieten keine Verbindung zu den eigentlichen Artikeln. Daher ist es schwierig, interessante und eventuell informative Artikel zu finden. Ein Besucher, der nur über bestimmte für ihn wichtige Themen des Forums auf dem Laufenden bleiben will, hat es ebenso schwer wie ein konkret Hilfesuchender, welcher nicht die Zeit für das Warten auf eine Antwort hat.

Wie kann also das Wissen innerhalb von Foren weiter genutzt werden? Wie kann die Information, welche die Artikel beinhalten, weiter zugreifbar bleiben? Eine mögliche Lösung sind Klassifikationen. Eine Klassifikation dient der Strukturierung eines Wissensgebietes nach einem formalen Schema. Dieses formale Schema besteht aus Klassen, welche hierarchisch angeordnet sind und in die dann die Objekte, wie beispielsweise die Artikel einer Newsgroup, eingeordnet werden. Eines der bekanntesten Beispiele für eine hierarchische Klassifikation ist der Web-Katalog von *Yahoo!*. Wenn nun die Artikel eines Forums in einer solchen Klassifikation abgelegt sind, so haben Besucher eine zusätzliche Zugriffsstruktur. Sie können die Hierarchie dazu nutzen, durch die Menge der Artikel zu browsen

[3] Artikel einer Newsgroup können nur über einen Newsreader geladen und angesehen werden. Bei Foren werden die Artikel direkt auf der Homepage im Internet mittels eines Browsers gelesen.

und sich die Untermenge, die Klasse, anzuschauen, die sie interessiert. Mithilfe einer solchen Zugriffsstruktur können zwei Informationsbedürfnisse befriedigt werden. Zum einen können Besucher, die sich generell zu einem Gebiet informieren wollen, durch die Klasseneinteilung erkennen, wie sich eine Domäne strukturiert und können sich die Artikel in den sie interessierenden Teilbereichen durchlesen. Zum anderen können Besucher, welche ein konkretes Informationsbedürfnis haben, recht schnell Artikeln finden, in denen sie die gesuchte Information erhalten.

Klassifikationen haben allerdings einen hohen Erstellungs- und Pflegeaufwand. Die Klassen müssen erzeugt werden und jeder Artikel muss in den verschiedenen Ebenen der Hierarchie in einer oder mehreren Klassen abgelegt werden. Weiterhin kommen jeden Tag neue Artikel hinzu und bereits bestehende werden durch Kommentare erweitert. Die neuen Artikel müssen nun wiederum zugeordnet werden und die erweiterten müssen dahingehend überprüft werden, ob sie nicht eventuell weiteren oder sogar komplett anderen Klassen zugeordnet werden. Wegen diesem hohen Aufwand für die Erstellung und die Pflege einer Klassifikation wird sehr selten eine solche für eine Newsgroup oder ein Forum erstellt.

Verfahren der Clusteranalyse bieten die Möglichkeit, Klassifikationen automatisch zu erstellen. Aufgrund bestimmter Attribute, bei Texten sind das zum Beispiel die Worthäufigkeiten, wird versucht, diese in vorher nicht näher spezifizierte Klassen, hier Cluster (Klumpen oder Haufen) genannt, einzuteilen. Es müssen keine Beispielsklassen vom Anwender angelegt und mit Datensätzen befüllt werden, wie das bei einer automatisierten Klassifikation der Fall ist. Die Clusteranalyse ist eine Methode aus dem Bereich des maschinellen Lernens. Es handelt sich um sogenanntes unüberwachtes Lernen. Im Folgenden soll die entstehende Struktur eine Clusteranalyse als *Clustering* bezeichnet werden, um sie von der Klassifikation abzugrenzen.

Mit Hilfe von Clusteranalyseverfahren kann eine automatische Strukturierung von Dokumentenkollektionen vorgenommen werden; diese kann flach oder hierarchisch sein. Besonders interessant für die Strukturierung der Artikel einer Newsgroup sind Verfahren die eine hierarchische Struktur erstellen[4]: eine Cluster-Hierarchie. Denn Hierarchien erlauben das Entdecken eines Wissensgebietes vom Allgemeinen zum Speziellen. Wenn eine solche hierarchische Zugriffsstruktur vorhanden ist, kann sich der Besucher einer Newsgroup die Teilmenge von Artikeln selektieren, die das Wissen enthält, dass ihn interessiert. Hierfür muss er keine genaue Frageformulierung im Kopf haben, sondern kann sich durch die Hierarchie *hangeln*, bis er in der Klasse angelangt ist, welche thematisch zu seinem Informationsbedürfnis passt.

Ein wesentliches Problem bei der Clusteranalyse von Newsgroupartikeln ist die Dynamik, die Newsgroups inhärent ist. Diese umfasst zum einen den oben bereits erwähnten Vorgang, dass täglich neue Beiträge hinzukommen und damit neue Artikel eröffnet oder bereits bestehende erweitert werden. Zum anderen werden durch diese neuen Beiträge teilweise auch inhaltliche Veränderungen ausgelöst. Beispielsweise kann in einer Software-Newsgroup das Erscheinen einer neuen Applikation dazu führen, dass sich viele der neu eröffneten Artikeln mit dieser Applikation beschäftigen. In dem für diese Newsgroup erstelltem Clustering gibt es aber noch keinen Cluster, in den diese Artikel eingeordnet werden könnten. Folglich muss ein neuer Cluster erstellt werden. Ähnlich kann eine veraltete Software mit der Zeit nicht mehr Gegenstand von neuen Beiträgen sein. Der

[4] Wie bei einer Klassifikation.

Cluster, welcher die Artikel über diese Software enthält, sollte gelöscht werden, damit das Clustering mit der Zeit nicht unübersichtlich wird. Die Dynamik einer Newsgroup äußert sich also in dem Zuwachs an Artikeln und der daraus resultierenden inhaltlichen Fluktuation.

Eine Clusteranalyse für Newsgroups muss dieser Dynamik Rechnung tragen. Neue Artikel müssen den entsprechenden Clustern in der Cluster-Hierarchie zugeordnet werden. Neue Cluster müssen erstellt und alte gelöscht werden. Das Verfahren muss die Hierarchie inkrementell erweitern können. Eine Prämisse für diese Inkrementierung ist jedoch, dass die Cluster-Hierarchie sich so wenig wie möglich ändern soll. Der Grund für diese Prämisse liegt darin, dass der Besucher einer Newsgroup sich an eine bestimmte Anordnung - eine Cluster-Hierarchie - gewöhnt hat. Er weiß, in welchem Bereich die Themen zu finden sind, die ihn interessieren. Dementsprechend ist eine Änderung der Anordnung für ihn mit dem Umstand verbunden, dass er sich neu orientieren muss. Also sollte sich die Cluster-Hierarchie so wenig wie möglich ändern. Andererseits leidet die Güte der Hierarchie, wenn sie nicht reorganisiert wird. Die Güte einer Cluster-Hierarchie verringert sich beispielsweise, wenn ein Cluster nicht erstellt wird, obwohl viele Artikel in diesen neuen Cluster eingeordnet werden könnten oder Cluster mit nicht mehr diskutierten Themen die Hierarchie unübersichtlich machen. Hier existiert also ein Zielkonflikt. Zum einen soll die Clusterhierarchie so wenig wie möglich verändert werden, zum anderen soll aber eine größtmögliche Güte garantiert werden.

1.1 Zielsetzung und Gliederung

Im Folgenden wird die Zielsetzung und die Gliederung der Arbeit erläutert.

In dieser Diplomarbeit wird ein Verfahren zur Clusteranalyse vorgestellt, das die oben geschilderten Anforderungen der Clusteranalyse für die Artikel einer Newsgroup berücksichtigt. Das Verfahren soll inkrementell sein und es soll die Möglichkeit bieten, mit vorgegebenen Strukturänderungen während der Inkrementierung eine hohe Güte der Clusterhierarchie zu gewährleisten.

Im Kapitel 2 werden die Grundlagen des Gebietes der Clusteranalyse erläutert und verschiedene Verfahren vorgestellt. Die Vor- und Nachteile der einzelnen Verfahren werden aufgezeigt.

Danach wird im Kapitel 3 eine detaillierte Anforderungsanalyse für das Clustering der Artikel einer Newsgroup durchgeführt. Mithilfe der aufgestellten Anforderungen soll ein Verfahren ausgewählt und angepasst werden.

Um das initiale Clustering der Artikelkollektion einer Newsgroup durchführen zu können, müssen verschiedene Vorverarbeitungsschritte durchlaufen werden. Diese Vorverarbeitung muss größtenteils auch für die Inkrementierung der Clusterhierarchie gemacht werden. Daher wird sie in Kapitel 4 separat beschrieben. Hierbei wird das Augenmerk auf die Besonderheiten von Newsgroupartikeln gelegt.

Im Anschluss wird in Kapitel 5 erläutert, wie das gewählte Clusteranalyseverfahren modifiziert wird, um einige Schwächen des ursprünglichen Ansatzes zu beheben und ihn an die speziellen Eigenheiten der Domäne anzupassen. Nach diesen Anpassungen kann mithilfe des Verfahrens eine Clusterhierarchie erstellt werden.

Im Kapitel 6 wird eine inkrementelle Erweiterung des Verfahrens erläutert. Diese wird es erlauben, weitere Artikel und Beiträge in die Hierarchie aufzunehmen. Hierbei soll auf

die möglichen Strukturänderungen an der Hierarchie eingegangen werden und wie sich der Grad der Strukturänderungen operationalisieren lässt. Die aus diesen Betrachtungen hervorgehenden Resultate dienen der Inkrementierung zur Behandlung des Zielkonfliktes zwischen Güte und Strukturkonstanz der Clusterhierarchie. Im Kapitel 7 sollen die Ergebnisse der Evaluation des Clusterings aus den beiden Kapiteln 5 und 6 beschrieben werden. Die Evaluation soll zeigen welche qualitativen Unterschiede zwischen dem inkrementellen Clustering einer existierenden Clusterhierarchie, im Vergleich zu dem erneuten initialen Clustering (also einem Reclustering) bestehen, und wie sich verschiedene Arten von Strukturänderungen auf das inkrementelle Clustering auswirken.

Die folgende Abbildung fasst die Zielsetzung und Vorgehensweise der Arbeit stichpunktartig zusammen.

Zielsetzung

Entwicklung eines Verfahrens zum Clustering von Newsgroupartikeln, welches das inkrementelle Erweitern einer bestehenden Clusterstruktur um neue Artikel ermöglicht.

Vorgehensweise

1. Übersicht zum Gebiet der Clusteranalyse (Kapitel 2)
   a) Erläuterung der Grundlagen der Clusteranalyse
   b) Vorstellung verschiedener Verfahren der Clusteranalyse und Erörterung ihrer Eigenschaften
2. Anforderungsanalyse (Kapitel 3)
   a) Anhand eines Szenarios werden die Anforderungen einer Clusteranalyse für Newsgroup Artikel spezifiziert
   b) Auswahl eines Verfahrens zur Clusteranalyse aus den in Kapitel 2 vorgestellten Verfahren
3. Anpassung und Erweiterung des gewählten Verfahrens
   a) Beschreibung der speziellen Vorverarbeitung der Artikel einer Newsgroup (Kapitel 4)
   b) Verbesserung des Verfahrens (Kapitel 5) zur Durchführung des initialen Clusterings
   c) Inkrementelle Erweiterung des Verfahrens (Kapitel 6)
4. Evaluation des Verfahrens (Kapitel 7)

2 Clustering

Ziel der Clusteranalyse ist, eine Menge von Objekten nach dem Grad ihrer Ähnlichkeit zu gruppieren, anders ausgedrückt, soll in einer vorgegebenen Objektmenge eine inhärent vorhandene Struktur *sichtbar* gemacht werden. Die Clusteranalyse ist somit ein Verfahren der multivariaten Datenanalyse und gehört dort zu den Q-Techniken (siehe [HE89]). Die Clusteranalyse hat eine Vielzahl von Einsatzgebieten und wird von unterschiedlichen Disziplinen eingesetzt. So wird sie beispielsweise in der Archäologie zur Altersbestimmung von Fundstücken verwandt. Es ist bekannt, dass die Muster von Tonscherben der gleichen geschichtlichen Epoche sich mehr ähneln, als die Muster aus unterschiedlichen Epochen. Auf Grundlage dieser Erkenntnis kann nun eine Menge von Tonscherben mittels der Clusteranalyse gruppiert und somit aufgezeigt werden, welche Tonscherben aus der gleichen Epoche stammen. Ein weiteres Einsatzgebiet der Clusteranalyse ist das Finden von Mustern und Zusammenhängen auf den Daten einer Datenbank. Das Schlagwort, welches diesen Vorgang beschreibt, ist *Knowledge Discovery in Databases (KDD)*. Die Disziplin, die sich mit dieser Aufgabe beschäftigt, nennt sich *Data Mining*. In Datenbanken ist oft eine große Anzahl von gleichartigen Objekten in Tabellen gespeichert. Jedes Objekt verfügt über dieselben Attribute, die es beschreiben. Anhand dieser Attribute können Ähnlichkeiten zwischen den einzelnen Objekten ausgemacht werden. Auf diesen Umstand kann zurückgegriffen werden und mittels der Clusteranalyse können Gruppen, genannt Cluster, von ähnlichen Objekten zusammengefasst und eine Struktur in der Menge der Daten erkannt werden, um die Analyse zu ermöglichen. Als letztes Beispiel für den Einsatz der Clusteranalyse sei noch das Clustering von Texten genannt. Hier wird die Clusteranalyse als Verfahren der Informationsextraktion verwandt. Ziel ist das Erkennen einer thematischen Struktur, um beispielsweise das Erstellen von Klassifikationshierarchien oder das Browsing durch die Dokumentenkollektion zu erleichtern.

2.1 Grundlagen des Clusterings

Die Grundlage für die Clusteranalyse ist die Definition der Ähnlichkeit zwischen den Objekten, die gruppiert werden sollen. Sie ist von zentraler Bedeutung für die Qualität des Ergebnisses der Clusteranalyse. In [HTF01] vergleicht Hastic die Wahl des Ähnlichkeitsmaßes mit der Wahl der Kostenfunktion beim Klassifikationsproblem (*überwachtes Lernen*). Wie bereits erwähnt, ist es das Ziel der Clusteranalyse, eine Menge von Objekten in Gruppen (Cluster) ähnlicher Objekte zu gruppieren. Eine Bedingung, die sich direkt aus dieser Aussage herleiten lässt, ist, dass Objekte innerhalb des selben Clusters sich ähnlicher sind als Objekte in verschiedenen Clustern. Die Ähnlichkeit oder auch Unähnlichkeit zwischen zwei Objekten wird durch die Summe der Abweichungen ihrer Attribute bestimmt.

2.1.1 Objekte und Attribute

In der Statistik [HEK89] spricht man statt von Objekten, auch von Untersuchungseinheiten oder Beobachtungen und von Merkmalen, über die diese Beobachtungen verfügen, anstelle von Attributen. Der Wert, den ein bestimmtes Merkmal annimmt, bezeichnet man als Merkmalsausprägung. Merkmalsausprägungen können sehr unterschiedlicher Natur sein. Es kann sich hierbei beispielsweise um Zahlen, Zustände oder einfache Bezeichnungen handeln. Man unterscheidet zwischen *quantitativen* und *qualitativen* Merkmalen. *Nominale* Merkmale sind qualitativer Natur. Die Ausprägungen nominaler Merkmale haben keine Ordnung. Sie haben also keine implizite Ordnung. Beispiele hierfür sind Haar- oder Augenfarbe, Beruf oder Nationalität. Als Spezialfall von nominalen Merkmalen seien Merkmale mit binärer Ausprägung erwähnt, beispielsweise Ja/Nein, Frau/Mann oder vorhanden/nicht vorhanden. Die nächste Kategorie ist die der *ordinalen* Merkmale. Ihre Ausprägungen unterliegen einer Rangfolge. Schulnoten sind ein Beispiel. Allerdings zählen diese Merkmale noch zu den qualitativen, da sich zwar entscheiden lässt, ob eine Ausprägung *besser* ist als eine anderen, aber nicht um *wie viel* dies der Fall ist. Die Abstufung zwischen den einzelnen Stufen eines ordinalen Merkmals ist mitunter sehr subjektiv. Teilweise werden ordinale Merkmale aber quantitativ aufgefasst.

Für statistische Zwecke, wie zum Beispiel im Falle der Clusteranalyse, am besten geeignet sind die *metrischen* Merkmale. Größe, Gewicht, Anzahl und Zeitdauer sind beispielsweise metrische Merkmale. Die Ausprägungen dieser Merkmale unterliegen nicht nur einer Reihenfolge, sondern auch die Abstände zwischen ihnen sind interpretierbar (siehe [HEK89]). 5 Jahre sind 2 Jahre mehr als 3 Jahre. Bei metrischen Merkmalen kann man noch zwischen stetigen und diskreten Merkmalen unterscheiden. Diskrete Merkmale können nur endlich viele oder abzählbar unendlich viele Werte enthalten. Stetige dagegen überabzählbar viele in einem beliebigen Intervall.

Im Weiteren soll für den Ausdruck *Beobachtungen* wieder das Wort *Objekte* und für Merkmale das Wort *Attribute* verwendet werden. Und anstelle von Ausprägungen wird von Werten die Rede sein. In den folgenden Abschnitten wird es um die Berechnung von Distanzen zwischen Objekten mittels ihrer Attribute gehen.

2.1.2 Klassifikationstypen

Häufig werden in Abhandlungen über die Clusteranalyse als Ergebnistypen oder Arten des Clusterings nur zwei grundlegend verschiedene Arten erwähnt. Das partitionierende und das hierarchische Clustering. Diese Klassifikation soll hier noch etwas detaillierter ausgeführt werden. Hartung nennt in [HE89] vier Klassifikationstypen: einmal die *Überdeckung* und die *Partitionierung* und zum anderen die *Hierarchie* und die *Quasihierarchie*. Die beiden Erstgenannten erstellen eine flache Struktur und die anderen beiden hierarchische Baumstrukturen.

2.1.2.1 Überdeckung

Diese flache Struktur besteht aus Clustern, die sich überschneiden können. Anders gesagt, ein Objekt kann in mehreren Clustern vorkommen. Allerdings darf kein Cluster einen anderen vollständig enthalten.

Sei C eine Überdeckung und seien C_i und C_j Cluster dieser Überdeckung,

mit $i \neq j$, dann muss gelten:

$C_i \cap C_j \notin \{C_i, C_j\}$

2.1.2.2 Partition

Eine Partition ist eine Überdeckung mit der Einschränkung, dass die Cluster disjunkt sind, also kein Objekt in mehr als einem Cluster vorkommt.

Sei C eine Partition mit $C_i, C_j \in C$, so muss gelten:

$C_i \cap C_j = \emptyset$ mit $C_i, C_j \in C$ und $i \neq j$

2.1.2.3 Quasihierarchie

Eine Quasihierarchie ist eine Folge von Überdeckungen. Jeder Cluster einer Überdeckung wird wiederum durch eine Überdeckung repräsentiert. Eine Quasihierarchie ist also nicht flach, sondern besteht aus mehreren Stufen. Jede Stufe für sich ist eine Überdeckung der Gesamtmenge aller Objekte. Auf einer Stufe darf kein Cluster einen anderen vollkommen enthalten. Über die Stufen hinweg ist dies aber sehr wohl der Fall.

Sei C_i der Cluster einer Quasihierarchie. So ist die Vereinigung aller echten Teilmengen dieses Clusters gerade genau das Cluster selbst oder die leere Menge.

$$\bigcup_{C_j \subset C_i} C_j \in \{\emptyset, C_i\}$$

2.1.2.4 Hierarchie

Eine Hierarchie ist eine Folge von Partitionen. Sie ist also ein Spezialfall der Quasihierarchie, für den zusätzlich als Bedingung gilt

$C_i \cap C_j \in \{C_i, C_j, \emptyset\}$

2.1.3 Exhaustive Clusteranalyse

Ein letztes Unterscheidungsmerkmal für ein Clustering ist die Abdeckung, die erreicht wird, genauer gesagt, ob es sich um ein exhaustives (erschöpfendes) oder um ein nicht-exhaustives Clustering handelt. Ein exhaustives Verfahren deckt alle Objekte ab und fügt sie mindestens einem Cluster hinzu. Bei einem nichtexhaustivem Clustering gibt es Objekte die keinem Cluster zugeordnet sind. (siehe [HE89])

2.1.4 Formalisierung

Bisher wurde von einer Objektmenge und ihren Attributen gesprochen. Im Folgenden wird diese allgemeine Beschreibung formalisiert, um sie besser handhabbar zu machen.

Gegeben sei eine Menge von gleichartigen Objekten S. Jedes dieser Objekte x_i verfügt über einen Vektor von Attributen. Dieser Attributvektor ist bei allen Objekten der Menge

S gleich. Bei den Objekten kann es sich um Texte in einer Dokumentenkollektion mit ihren Wortvektoren handeln oder um die Ergebnisse einer statistischen Erhebung mit ihren Ergebnisvektoren. Formal:

S: Objektmenge mit $(|S| = n \in \aleph)$

$\vec{x}_i$: i-tes Objekt aus S

$x_{i,j}$: j-tes Attribut des i-ten Objektes

$$\begin{array}{l} \vec{x_1} = \\ \vec{x_2} = \\ \vdots \\ \vec{x_n} = \end{array} \begin{pmatrix} x_{1,1} & \cdots & x_{1,d} \\ x_{2,1} & \cdots & x_{2,d} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,d} \end{pmatrix}$$

Hierbei hat jeder Attributvektor eine Dimensionalität von $d \in D$. Aus den Objekten mit ihren Attributvektoren ergibt sich eine $n \times d$-Matrix, die sogenannte Datenmatrix. Im Folgenden soll es sich bei den Attributen um quantitative Attribute handeln. Dies ist insbesondere eine explizite Voraussetzung bei den meisten Distanzmaßen, die im folgenden Abschnitt dargestellt werden.

2.1.5 Ähnlichkeits- und Distanzmaße

Viele der traditionellen, kombinatorischen Verfahren der Clusteranalyse verwenden so genannte Distanzmaße. Diese geben den aus den Attributvektoren berechneten Abstand zwischen den einzelnen Objekten wieder. Der Abstand repräsentiert die Ähnlichkeit oder Unähnlichkeit zwischen zwei Objekten. Je geringer die Distanz ist, desto höher ist die Ähnlichkeit. Ein erstes Maß zur Messung des Abstandes zwischen zwei Objekten ist die *Hamming Distanz*.

$$dist(\vec{x}, \vec{y}) = \frac{1}{|D|} \sum_{i=1}^{D} \delta_i, \delta_i = \begin{cases} 0, & falls\ x_i = y_i \\ 1, & sonst \end{cases} \tag{2.1}$$

Hier wird auf die Gleichheit der Attributausprägungen der Attributwerte abgestellt. Es wird aufsummiert in welchen Attributen die Objekte nicht übereinstimmen.

Distanzfunktionen müssen die folgenden drei Bedingungen erfüllen:

- $\forall \vec{x}, \vec{y} \in S : dist(\vec{x}, \vec{y}) \in \Re^{\geq 0}$
- $dist(\vec{x}, \vec{y}) = 0 \Leftrightarrow \vec{x} = \vec{y}$
- $dist(\vec{x}, \vec{y}) = dist(\vec{x}, \vec{y})$

Wenn es sich bei den Attributwerten um rein quantitative Daten handelt, wovon im Weiteren ausgegangen wird, wird häufig die *Euklidische Distanz* verwendet.

$$dist(\vec{x}, \vec{y}) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_d - y_d)^2} = \sqrt{\sum_{i=d}^{d} (x_i - y_i)^2} \tag{2.2}$$

Die *Euklidische Distanz* erfüllt noch eine weitere Bedingung: die sogenannte Dreiecksungleichung.

$$\forall \vec{x}, \vec{y}, \vec{z} \in S : dist(\vec{x}, \vec{z}) \geq dist(\vec{x}, \vec{y}) + dist(\vec{y}, \vec{z})$$

Distanzfunktionen die zusätzlich die Dreiecksungleichung erfüllen werden auch *Metriken* genannt. Die *Euklidische Distanz* ist also eine solche Metrik. Sie ist ein Spezialfall der L_r-Metriken ($r \geq 0$)(siehe [HE89]). Die allgemeine Formel für L_r-Metriken, die auch *Minkowski* Distanz genannt wird, (siehe [Han81]) lautet:

$$dist(\vec{x}, \vec{y}) = \sqrt[r]{\sum_{i=d}^{d} |x_i - y_i|^r} \tag{2.3}$$

Ein Problem bei der direkten Anwendung der L_r-Metriken ist, dass die Attribute mit den numerisch größten Ausprägungen die anderen Attribute dominieren und so das Ergebnis überdurchschnittlich stark beeinflussen (siehe [JMF99]). Eine Möglichkeit, dieser Tendenz entgegen zu wirken, ist die Verwendung von Normierungen und Attributgewichtungen.

Zu den metrischen Distanzmaßen zählen auch die *Manhattan-Distanz* (siehe Formel 2.4) und die *Maximumsmetrik*(siehe Formel 2.5).

$$dist(\vec{x}, \vec{y}) = |x_1 - y_2| + \cdots + |x_1 - y_2| = \sum_{i=d}^{d} |x_i - y_i| \tag{2.4}$$

$$dist(\vec{x}, \vec{y}) = max(|x_1 - y_2|, \cdots, |x_1 - y_2|) \tag{2.5}$$

Mit den bisher genannten Maßen lässt sich die Distanz oder Unähnlichkeit zwischen zwei Objekten ausdrücken[1]. Das *Cosinus-Maß* (Formel 2.6) misst im Gegensatz dazu die Ähnlichkeit zwischen zwei Objekten.

$$sim(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{d} x_i * y_i}{\sqrt{\sum_{i=1}^{d} (x_i)^2 * \sum_{i=1}^{d} (x_i) y_i}} \tag{2.6}$$

Je größer der berechnete Wert für das *Cosinus-Maß* ist, desto größer ist die Ähnlichkeit.

Mittels der genannten Distanz- und Ähnlichkeitsmaße kann nun die (Un-)Ähnlichkeit zwischen den Objekten gemessen werden. Vielfach wird nun aus der Datenmatrix (siehe Abschnitt 2.1.4) so eine Distanz- oder auch Ähnlichkeitsmatrix berechnet. Dies ist eine $n \times n$-Matrix, welche die Distanzen zwischen den einzelnen Objekten enthält.

$$D = \begin{pmatrix} 0 & dist(x_1, x_2) & \cdots & dist(x_1, x_n) \\ dist(x_2, x_1) & 0 & \cdots & dist(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ dist(x_n, x_1) & d(x_n, x_2) & \cdots & 0 \end{pmatrix}$$

[1] Je größer der Wert ist den das Maß annimmt, um so größer ist die Unähnlichkeit zwischen zwei Objekten.

Für einige Verfahren zur Clusterkonstruktion kann die Distanzmatrix verwendet werden. Für andere muss die Datenmatrix herangezogen werden.

2.1.6 Generelle Probleme der Clusteranalyse

Neben speziellen problematischen Auswirkungen, die bestimmte Clusteralgorithmen mit sich bringen (siehe Abschnitt 2.2), gibt es bereits im Vorfeld einige wichtige Überlegungen, die für ein erfolgreiche Clustering notwendig sind.

Welche Attribute eines Objektes sollen für die Analyse herangezogen werden? Soll beispielsweise das Kaufverhalten von Konsumenten im Bereich von Lebensmitteln analysiert werden, ist eine Klassifikation der Konsumenten, welche die Augenfarbe mit einbezieht nicht sehr aussagekräftig. Es muss also nachgesehen werden, welche Attribute wirklich für die Domäne relevant sind. Eine Selektion der Attribute ist in diesen Fällen nur von Personen mit entsprechendem Fachwissen sinnvoll durchführbar.

Aber nicht nur von diesem inhaltlichen Standpunkt aus ist die Wahl einer Untermenge der Gesamtattributmenge angezeigt. Denn auch auf die Performance hat die Anzahl der Attribute, die verwendet werden, große Auswirkungen. Also ist hier die Frage, welche Attribute weggelassen oder zusammengefasst werden können, um qualitativ ein (gleich-)gutes Ergebnis zu erzielen, wie mit ihnen. Dieser Prozess wird im Allgemeinen Dimensionsreduktion genannt. Zwei Techniken werden hierzu verwendet. Zum einen die Attributselektion (*feature selection*), mit der entschieden wird, welche Attribute ausgelassen werden können, zum anderen die Attributextraktion (*feature extraction*), mithilfe derer verschiedene Attribute zusammengefasst werden können. (siehe hierzu [JMF99] und [Han81])

Bei räumlich orientierten Clusteransätzen (Clusterverfahren) ist es wichtig, wie mit dem sogenannten *Rauschen*[2] umgegangen wird. Ist damit zu rechnen, dass Ausreißer in der Datenmenge enthalten sind, so sollte man ein Verfahren verwenden, welches diesem Umstand Rechnung trägt. Solche Verfahren sind zum Beispiel *DBSCAN*, welches in Abschnitt 2.2.2 erläutert wird, und *WaveCluster* (siehe [SCZ98]).

2.1.7 Gütemaße und Clusterkriterium

Hartung nennt in [HE89] zwei Maße zur Bewertung der Güte eines Clusterings:

1. *Intra Clusterhomogenität*
2. *Inter Clusterheterogenität*

Die *intra Clusterhomogenität* misst die Güte eines Clusterings anhand der *Gleichartigkeit* der Objekte innerhalb der einzelnen Cluster. Je geringer (größer) die Distanz (Ähnlichkeit) der Objekte in den Clustern ist, desto höher ist die Güte des Clusterings. Die *inter Clusterheterogenität* wiederum bewertet die Güte anhand der *Verschiedenartigkeit* der Objekte in unterschiedlichen Clustern. Je größer (geringer) die Distanz (Ähnlichkeit) der Objekte ist, die sich in verschiedenen Clustern befinden, desto höher ist die Güte des Clusterings.

[2] Mit Rauschen (engl. noise) sind Objekte gemeint, welche außerhalb der restlichen Objekte angesiedelt sind - also Ausreißer.

Abwandlungen dieser Bewertungsmaße für die Güte eines Clusterings werden häufig als Clusterkriterium in den Verfahren der Clusteranalyse eingesetzt. So soll beispielsweise bei dem *k-means* Verfahren (siehe Abschnitt 2.2.1) die Summe der quadratischen Abweichungen zwischen den Objekten innerhalb der Cluster minimiert werden. Es soll also die Homogenität der Cluster maximiert werden.

2.2 Verfahren der Clusteranalyse

In diesem Abschnitt werden nun einige gängige Verfahren zur Clusteranalyse beschrieben. Auch diese Verfahren zur Konstruktion von Clusterings können in unterschiedliche Arten eingeteilt werden. Diese Einordnung soll jeweils bei den einzelnen Verfahren näher erläutert werden. Alle unterschiedlichen Formen, Arten und Ausprägungen von Verfahren hier aufzuzählen und zu diskutieren, würde allerdings den Rahmen dieser Arbeit sprengen. Dementsprechend werden nur einige ausgewählte Verfahren erläutert.

2.2.1 k-means

k-means ist ein in der Literatur sehr häufig erwähntes Verfahren der Clusteranalyse. Es erzeugt in seiner ursprünglichen Form eine flaches, partitionierendes und exhaustives Clustering. *k-means* beginnt mit einem initialen Clustering und versucht danach, dieses iterativ zu optimieren. Hierfür verwendet das Verfahren ein globales Cluster-Kriterium. Genauer wird die Optimierung mit Hilfe einer Kostenfunktion vorgenommen, die über das gesamte Clustering definiert ist. Und zwar mit der Summe der quadratischen Abweichungen.

$$c(C_i) = \sum_{r=1}^{|C_i|} \sum_{s=1}^{|C_i|} (dist(x_r^i, x_s^i))^2 \tag{2.7}$$

x_r^i ist das r-te Element des Clusters C_i

Diese Kostenfunktion definiert ein Maß für die Homogenität eines Clusters. Sie kann auch ausgedrückt werden durch die Summe der quadrierten Abweichungen aller Objekte eines Clusters zum Mittelpunkt μ des Clusters.

$$c(C_i) = \sum_{r=1}^{|C_i|} (dist(\mu^i, x_r^i))^2 \tag{2.8}$$

Der grundlegende Ablauf des Verfahrens kann wie folgt beschrieben werden:

1. Wähle zufällig k Mittelpunkte für die Cluster.
2. Weise jedes Objekt $x \in S$ dem Cluster zu, dessen Mittelpunkt es am nächsten ist.
3. Berechne für jeden Cluster C_i seine Kosten über die Funktion 2.8.
4. Berechne für jeden Cluster C_i seinen neuen Mittelpunkt μ_i mit

$$\mu_i = \frac{1}{|C_i|} \sum_{r=1}^{|C_i|} x_r^i$$

5. Wiederhole die Schritte 2 - 4 solange, bis sich die $c(C_i)(\forall i)$ oder die Cluster selbst nicht mehr ändern.

Als Eingabeparameter wird die Anzahl an Clustern übergeben, die das Verfahren produzieren soll. Hierin ist einer der größten Nachteile von *k-means* zu sehen. Der Anwender wird in den wenigsten Fällen wissen, wie viele Cluster der Struktur inhärent sind. Also ist die notwendige Angabe dieses Parameters in der Regel ein *Schuss ins Blaue*.

Eine weiterer Nachteil von *k-means* ist die Neigung zu sphärischen (kugelförmigen) Clustern. Andere Formen werden kaum erkannt und oft in kleinere Cluster zerlegt. Auch weist der Algorithmus jedes Objekt einem Cluster zu, auch wenn das Objekt sehr weit von allen Clustern entfernt ist. Er ist also nicht in der Lage, Ausreißer in der Objektmenge zu erkennen. Als letzter Nachteil sei noch erwähnt, dass *k-means* sensitiv gegenüber der Auswahl der initialen Clustermittelpunkte ist. Anders gesagt, der Algorithmus findet nicht mit Sicherheit ein globales Optimum, sondern eher ein lokales Optimum das der Startkonstellation nahe ist.

Ein positiver Aspekt an *k-means* ist seine geringe Zeitkomplexität - $O(k * n)$.

Eine Abwandlung dieses ursprünglichen *k-means* Algorithmus ist *Bisecting k-means* (siehe [SKK00]).

1. Wähle zufällig ein Cluster der geteilt werden soll.
2. Erstelle zwei Subcluster unter Verwendung des ursprünglichen *k-means* Algorithmus.
3. Wiederhole den Schritt 2, eine vorgegebene Anzahl von Durchgängen und nimm den Durchgang als Ergebnis, in dem die höchste Homogenität der beiden Cluster erreicht wurde.
4. Wiederhole Schritt 1 und 2 so oft, bis die gewünschte Anzahl an Clustern erreicht wurde.

Mit der Homogenität eines Clusters ist gemeint, dass die Kostenfunktion des Clusters invertiert wird. Genauer $1/c(C_i)$. Je höher dieser Wert ist, desto homogener ist ein Cluster. *Bisecting k-means* ist nicht so anfällig gegenüber der Wahl der initialen Mittelpunkte und hat daher auch höhere Chancen, ein globales Optimum für das Clustering zu finden. Auch kann mit *Bisecting k-means* einerseits eine flache Partitionierung und andererseits eine Hierarchie erstellt werden.

2.2.2 DBSCAN

Eine andere Form des flachen Clusterings, welches nicht auf der Sichtweise beruht, dass Cluster sich durch Objekte bilden, die möglichst nah an einem Mittelpunkt liegen, ist das dichte-basierte Verfahren *DBSCAN* (*Density Based Spatial Clustering of Applications with Noise*) (siehe [EKS96]). Bei diesem Verfahren werden die Cluster anhand eines lokalen Clusterkriteriums bestimmt, welches sich der Definition der räumlichen Dichte bedient. Um dies näher zu erläutern, müssen als erstes einige Begriffe eingeführt werden.

- Sei $\epsilon \in \Re^{\geq 0}, x \in D$. Dann ist $N_\epsilon(x) = \{y \in D : dist(x, y) \leq \epsilon\}$ die ϵ-Umgebung von x.

- Sei $MinPTS \in N.x \in D$ ist ein *Kernobjekt*, falls $|N_\epsilon(x)| \geq MinPTS$.
- $x \in D$ ist *direkt dichte-erreichbar* von $y \in D$, falls y Kernobjekt ist und $x \in N_\epsilon(y)$.
- $x \in D$ ist *dichte-erreichbar* von $y \in D$, falls $\exists p_1, \ldots, p_n \in D : p_1 = y, p_n = xund \forall i \in \{1, \ldots, n-1\} : p_{i+1}$ ist *direkt dichte-erreichbar* von p_i.
- $x, y \in D$ sind *dichte-verbunden*, falls $\exists z \in D : x$ und y sind *dichte-erreichbar* von z.
- x ist ein *Randobjekt*, wenn es kein Kernobjekt ist und in der ϵ-Umgebung eines Kernobjektes liegt.
- x ist *Rauschen*, wenn es kein Kern- oder Randobjekt ist.

In Abbildung 2.1 sollen diese Definitionen veranschaulicht werden. Die ϵ-Umgebung der Objekte A, B und C ist durch die großen Kreise angegeben. Wenn nun ein $MinPTS$ von drei vorausgesetzt wird, sind diese Objekte offensichtlich Kernobjekte. Die Objekte D und E sind Randobjekte für jedes der erstgenannten drei Objekte. Alle Objekte A-E können einem Cluster zugeordnet werden. Das Objekt F ist Rauschen.

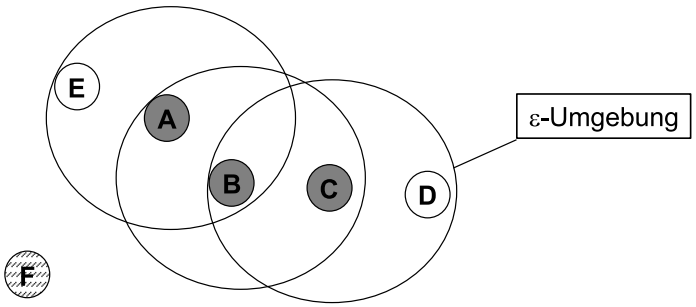


Abbildung 2.1: ϵ-Umgebung, Dichte-Erreichbarkeit, Randobjekt und Rauschen

Mithilfe dieser Definitionen lassen sich sehr einfach Cluster erzeugen. Man nehme ein beliebiges Kernobjekt als *Ur-Punkt* des ersten Clusters und ordne dann alle von diesem Objekt aus *dichte-erreichbaren* Objekte demselben Cluster zu. Danach nehme man das nächste Kernobjekt, welches noch keinem Cluster zugeordnet ist und verfährt ebenso. Dies wiederholt man bis keine Kernobjekte mehr unzugeordnet. So erhält man eine Überdeckung der Objektmenge (ein Randobjekt kann ja mehreren Clustern zugeordnet werden), in der alle Objekte die keinem Cluster zugewiesen sind, als Rauschen betrachtet werden können.

Die Abbildung 2.2 zeigt den Ablauf des DBSCAN-Algorithmus in einer Pseudocode-Notation. Da diese Notation auch für die Veranschaulichung kommender algorithmischer Abläufe eingesetzt wird, soll an dieser Stelle auf den DBSCAN-Algorithmus im Speziellen, aber auch auf die Notation im Allgemeinen eingegangen werden.

In den ersten drei Zeilen vor der eigentlichen Prozedur werden globale Objekte (im sinne objektorientierter Programmierung) definiert, welche in der Prozedur **doDBSCAN()** benutzt werden. Die *ObjectList* ist eine Liste mit allen Objekten in D. Die *ClusterList* ist eine leere Liste, in der die konstruierten Cluster gesammelt werden. Und *MinPTS* ist

```
ObjectList          := All Objects
ClusterList         := Empty
MinPTS              := Minimal required number of objects in ε-neighborhood

doDBSCAN()
1   For each Object ∈ ObjectList
2     If |ε-N(Object)| ≥ MinPTS AND
3      NOT (∃ Cluster ∈ ClusterList with Object ∈ Cluster) Then
4       NewCluster := ∪ {Object}
5       ProofQueue := ε-N(Object)
6       Loop (ProofQueue Not Empty)
7         ProofObject := First from ProofQueue
8         If NOT (ProofObject ∈ NewCluster) Then
9           NewCluster ∪ ProofObject
10          If (|ε-N(Object)| ≥ MinPTS) Then
11            ProofQueue ∪ ε-N(ProofObject)
12          End If
13        End If
14      End Loop
15      ClusterList ∪ NewCluster
16    End If
17  End For
```

Abbildung 2.2: DBSCAN Algorithmus

ein Zahlobjekt, das die Anzahl der minimal notwendigen Objekte in der ϵ-Umgebung definiert, welche notwendig sind, damit ein Objekt ein *Kernobjekt* ist.

In der Prozedur **doDBSCAN()** wird dann in einer Schleife (Zeile 1) für jedes Objekt in der *ObjectList* überprüft, ob es ein Kernobjekt ist (Zeile 2) und ob es noch keinem Cluster zugeordnet ist[3] (Zeile 3). Wenn beide Bedingungen erfüllt sind, wird ein neuer Cluster mit dem Kernobjekt angelegt und die Warteschlange *ProofQueue* mit der Menge der Objekte in der ϵ-Umgebung des Objektes gefüllt. Die Zeilen 4 und 5 können von daher als das Anlegen neuer Instanzen angesehen werden. Dies soll durch die Notation „:=“ verdeutlicht werden. Die Warteschlange *ProofQueue* enthält zu dem Zeitpunkt ihrer Initiierung in Zeile 5 also alle von *Object* aus *direkt dichte-erreichbaren* Objekte. In der Schleife zwischen den Zeilen 6 und 14 wird nun jedes Objekt der Warteschlange als erstes darauf überprüft, ob es noch nicht in dem in Zeile 4 neu angelegten Cluster enthalten ist. Wenn dies der Fall ist, darf das Objekt hinzugefügt werden. Als zweites wird in Zeile 10 überprüft, ob *ProofObject* ein Kernobjekt ist. Wenn ja, wird die Warteschlange an ihrem Ende um die Objektmenge in der ϵ-Umgebung von *ProofObject* erweitert. Dieses Vorgehen hat zur Folge, dass alle von dem initialen Kernobjekt aus *dichte-erreichbaren* Objekte in den neuen Cluster (*NewCluster*) eingeordnet werden. Wurde dies nun für ein Kernobjekt gemacht, wird nach dem nächsten Kernobjekt gesucht, welches noch keinem Cluster zugeordnet ist.

[3] Die Methode ϵ-**N**(*Object*) gibt die Menge aller Objekte in der ϵ-Umgebung von *Object* zurück (ohne *Object* selbst). Der Bereich der ϵ-Umgebung, also das ϵ, wird mit der Methode ϵ-**N** implizit als gegeben angesehen.

Aus der Betrachtung des Algorithmus ergeben sich einige Vorzüge von DBSCAN. Mit Hilfe dieses Verfahrens können Cluster beliebiger Form gefunden werden. Auch ist es im Gegensatz zu *k-means* deterministisch. Denn egal mit welchem Kernobjekt die Prozedur gestartet wird oder besser gesagt, wie die Liste aller Objekte auch sortiert sein mag, es werden immer dieselben Cluster gefunden. Des Weiteren kann Rauschen erkannt werden. Alle Objekte, die keinem Cluster zugeordnet sind, die also von keinem Objekt aus *dichte-erreichbaren* sind, können eben als Rauschen eingestuft werden. Ein letztes Merkmal von DBSCAN ist, dass Objekte in mehrere Cluster eingeordnet werden können. Es wird also eine Überdeckung produziert.

Allerdings hat auch dieser Algorithmus seine Schwächen. Auch er benötigt die Angabe von Eingabeparametern, die das Ergebnis signifikant bestimmen. Als erstes muss eine ϵ-Umgebung vom Anwender festgelegt werden. Welches ist aber die richtige Größe für diesen Parameter? Außerdem muss entschieden werden, ab welcher Objektanzahl in der ϵ-Umgebung eines Objektes dieses ein Kernobjekt ist. Das Finden dieser beiden Parameter unterliegt in den meisten Fällen einer *try-and-error*-Methode. Eventuell müssen viele Versuche gemacht und die Parameter immer wieder angepasst werden, ehe ein befriedigendes Ergebnis erzielt wird. Außerdem kann es auch Gebiete mit unterschiedlicher Dichte geben, so dass eher ein hierarchisches Clustering angezeigt ist. Eine Erweiterung des Verfahrens, das diesen Nachteil von DBSCAN behebt, wird in [ABK99] beschrieben. *OPTICS* arbeitet zwar noch mit einem festem MinPTS, aber der Wert für die ϵ-Umgebung wird unterhalb eines festen Schwellwertes ϵ variiert. So wird eine hierarchische Clusterstruktur aufgebaut, die der unterschiedlichen Dichte von Gebieten Rechnung trägt. Der Anwender ist nicht mehr gezwungen viele, Werte für ϵ auszuprobieren.

Es ist anzumerken, dass DBSCAN und auch OPTICS eine Laufzeit von $O(N^2)$ in der Anzahl der Objekte haben. Der Grund dafür ist, dass vor der eigentlichen Clusteranalyse die Distanzen zwischen den einzelnen Objekten berechnet werden müssen, also eine Distanzmatrix aufgebaut werden muss. Bei *k-means* ist die Datenmatrix ausreichend.

2.2.3 Agglomerative Clusteranalyse

In diesem Abschnitt soll nun als Letztes kein einzelnes Verfahren vorgestellt werden, sondern eine Klasse von Verfahren. Dabei wird nur recht oberflächlich auf die wichtigsten Merkmale eingegangen. Der gemeinsame Oberbegriff, welcher die hier gemeinten Verfahren zusammenfasst, ist: agglomeratives Clustering. Dieses Gruppe von Clusteringverfahren erzeugt *bottom-up* eine hierarchische Struktur. Das Vorgehen vereinfacht dargestellt, ist hier Folgendes:

1. Bilde für jedes Objekt x_i der Eingabemenge D einem eigenen Cluster C_i.
2. Suche aus der Menge aller Cluster C die zwei Cluster mit der geringsten Distanz $dist(C_i, C_j)$ zueinander und vereinige diese zu einem neuen Cluster $C_k = C_i \cup C_j$.
3. Führe Schritt 2 solange durch, bis nur noch ein Cluster mit allen Objekten aus D vorhanden ist.

Entscheidend bei diesen Verfahren ist das Kriterium, welches die Cluster selektiert, die zusammengefasst werden sollen. Dieses Kriterium ist der Abstand zwischen den Clustern. Dieser Abstand wiederum ist am Beginn des Verfahrens der Abstand zwischen den

einzelnen Objekten. Der Abstand zwischen den einzelnen Objekten kann mit den Maßen aus Abschnitt 2.1.5 gemessen werden. Nun muss das gewählte Abstandsmaß auf ganze Cluster ausgeweitet werden. Hierfür gibt es drei verschiedene Definitionen:

- **Single-Link**:
$$dist(C_i, C_j) = min_{x \in C_i, y \in C_j} dist(x, y) \tag{2.9}$$
- **Complete-Link**:
$$dist(C_i, C_j) = max_{x \in C_i, y \in C_j} dist(x, y) \tag{2.10}$$
- **Average-Link**:
$$dist(C_i, C_j) = \frac{1}{|C_i|\,|C_j|} \sum_{x \in C_i, y \in C_j} dist(x, y) \tag{2.11}$$

Verfahren, die mit **Single-Link** arbeiten, nehmen als Abstand zwischen zwei Clustern immer die Entfernung der beiden einander nächsten Objekte in diesen Clustern. **Complete-Link** Verfahren arbeiten hingegen immer mit den Objekten mit der größten Entfernung zwischen zwei Clustern. **Average-Link** Verfahren als Letztes errechnen die durchschnittliche Entfernung zwischen allen Objekten zweier Cluster und verwenden diesen Schnitt als Cluster-Entfernung.

Zu den Vorteilen des agglomerativen Clusterings zählen die hierarchische Struktur und das Fehlen von Parametern, die der Benutzer vor Beginn des Clusterings festlegen muss.

Allerdings haben alle Verfahren, je nachdem welche Cluster-Entfernungsdefinition sie anwenden, auch Nachteile. Ein Nachteil von **Single-Link** Verfahren ist, dass sie sehr längliche Cluster produzieren. Das bedeutet, dass einem Cluster immer weitere Cluster zugeordnet werden. Verfahren, die mit den beiden anderen Definitionen von Cluster-Entfernung arbeiten, produzieren im Allgemeinen nur sphärische Cluster (wie *k-means*). Ein weiterer Nachteil aller Verfahren des agglomerativen Clusterings ist ihre hohe Laufzeit ($O(N^2)$). Auch hier ist die Notwendigkeit der Berechnung der Distanzmatrix der Grund für die hohe Laufzeit.

2.3 Textclustering

Bisher wurde die Clusteranalyse im Allgemeinen behandelt. In diesem Abschnitt sollen einige Besonderheiten des Text- oder Dokumentclustering erläutert werden. In diesem Zusammenhang werden zwei Verfahren zur Durchführung einer Clusteranalyse vorgestellt, die sich von den bisher beschriebenen abstandsbasierten Verfahren gravierend unterscheiden.

Textclustering bezeichnet die Clusteranalyse, angewandt auf eine Kollektion von Dokumenten. Das Ziel ist es, die Dokumente inhaltlich oder thematisch zu gruppieren. Seinen Ursprung hat das Textclustering in der Informations-Extraktion oder genauer im *Information Retrieval*. Im Information Retrieval wird aus einer Dokumentenkollektion auf die Anfrage eines Anwenders hin eine Teilmenge von Dokumenten als Antwort ausgegeben. Die Dokumente in der Antwortmenge sind anhand ihrer Ähnlichkeit zur Anfrage geordnet. Die Ähnlichkeit zwischen Anfrage und Dokument wird auf die gleiche Art berechnet wie die Ähnlichkeit zwischen Objekten in der Clusteranalyse. Als formale Grundlage dient

hier das *Vektor-Raum-Modell* (siehe [vR79]). In diesem werden Dokumente und Anfragen als Termvektoren repräsentiert.

- Sei D die Dokumentenkollektion und T die Menge aller betrachteten Terme in der Kollektion.
- $d_i \in D$ das i-te Dokument.
- $t_{i,j}$ ist die Repräsentation des j-ten Terms aus der Menge aller Wörter T im i-ten Dokument.
- $w(t_{i,j})$ eine Gewichtungsfunktion.
- $\vec{d_i} = (w(t_{i,1}), \ldots, w(t_{i,K}))$ mit $K = |T|$.

Ein Dokument wird also als Vektor von Termen aufgefasst, wobei die Terme mit einer Gewichtungsfunktion versehen sind. Jeder Dokumentenvektor und jeder Anfragevektor hat die gleiche Dimensionalität $K = |T|$ [4]. Als Gewichtungsfunktion für einen Term in einem Dokument kann zum Beispiel die Termhäufigkeit herangezogen werden oder auch eine normalisierte Termhäufigkeit wie *tfxidf* (siehe 2.3.1).

Dokumentclustering wurde anfänglich als Methode genutzt, um das Information Retrieval zu verbessern (siehe [CKP92]). Hierfür wurde für die Dokumente beispielsweise eine hierarchische Clusteranalyse durchgeführt und als Antwort auf eine Anwenderanfrage die Cluster zurückgegeben, welche die höchste Ähnlichkeit aufwiesen. Die Strategien zum Auffinden dieser Cluster beruhten, wie das *Information Retrieval* an sich, auf der Suche nach dem *nächsten Nachbarn*. Die Retrievalqualität in Verbindung mit der Clusteranalyse war allerdings nicht besser als die direkte Dokumentensuche und so wurde keine sehr intensive wissenschaftliche Untersuchung dieses Gebietes für sinnvoll erachtet.

Erst eine andere Auffassung der Zielsetzung des Dokumentenclusterings führte zu einer intensiveren Forschung auf diesem Gebiet. In [CKP92] wird ausgeführt, dass die Zielsetzung von *Information Retrieval* das Finden eines Dokumentes auf ein spezielles Informationsbedürfnis eines Anwenders hin ist und das Clustering eher einer explorativen Suche gerecht wird. Als bestes Beispiel ist ein Anwender zu nennen, der keine spezielle Frage im Kopf hat, sondern sich generell zu einem Themengebiet informieren möchte. Dieser Anwender kann hierfür beispielsweise das Ergebnis eines hierarchischen Clusterings verwenden. Angefangen bei allgemeinen Dokumenten, kann er sich entlang der Hierarchie zu den spezielleren Ausführungen durcharbeiten und so ein Themengebiet erschließen. Nach dieser Neudefinition der Zielsetzung des Dokumentclusterings begann eine intensivere Erforschung dieses Gebietes. In den nächsten Abschnitten sollen ein paar Besonderheiten herausgegriffen und näher erläutert werden.

2.3.1 tfxidf Gewichtung

Da die Term-Gewichtungsformel *tfxidf* bei der Termselektion (siehe Abschnitt 4.2.5) für das in dieser Arbeit beschriebene Verfahren der Clusteranalyse eingesetzt wird, soll sie an dieser Stelle beschrieben werden. Die Formel für die *tfxidf*-Gewichtung (aus [vR79]

[4] An der Stelle j eines jeden Vektors steht die Gewichtung ein und desselben Terms. Sollte der Term in einem Dokument nicht vorhanden sein, so wird dort 0 als Wert angenommen.

entnommen) besteht aus zwei Komponenten: zum einen aus der inversen Dokumenthäufigkeit (idf) und zum anderen aus der normalisierten Vorkommenshäufigkeit (ntf). Um diese beiden Komponenten formalisieren zu können, sind weitere Definitionen notwendig. Sei

- l_i Dokumentlänge (Anzahl der Terme) des Dokumentes i,
- al Durchschnittliche Länge der Dokumente in D,
- $tf_{i,j}$ Vorkommenshäufigkeit von Term j in Dokument i und
- n_j Anzahl der Dokumente, in denen der Term j vorkommt.

Mithilfe dieser Zusatzdefinitionen lässt sich zunächst die inverse Dokumenthäufigkeit wie folgt formalisieren:

$$idf_j = \frac{\log \frac{|D|}{n_j}}{|D| + 1} \tag{2.12}$$

Die Formalisierung der normalisierten Vorkommenshäufigkeit lautet:

$$ntf_{i,j} = \frac{tf_{i,j}}{tf_{i,j} + 0,5 + 1,5\frac{l_i}{al}} \tag{2.13}$$

Die inverse Dokumenthäufigkeit gewichtet also einen Term umso höher, je weniger er in der gesamten Dokumentenkollektion vorkommt. Die normalisierte Vorkommenshäufigkeit eines Terms soll ihn entsprechend seiner Vorkommenshäufigkeit innerhalb des Dokumentes gewichten. Um unterschiedlich große Dokumentlängen auszugleichen, geht die jeweilige Dokumentlänge in die Berechnung mit ein, und zwar ins Verhältnis gesetzt zur durchschnittlichen Dokumentlänge der Kollektion. Aus diesen beiden Komponenten wird nun die *tfxidf*-Gewichtung eines Terms j in einem Dokument i berechnet.

$$w(t_{i,j}) = idf_j * ntf_{i,j} \tag{2.14}$$

Die Termvektoren setzen sich aus den Termgewichtungen zusammen, wie sie durch die Formel 2.14 berechnet werden.

2.3.2 Vorverarbeitung

Bevor eine Clusteranalyse auf einer Dokumentenkollektion durchgeführt werden kann, muss eine spezielle Vorverarbeitung stattfinden. Die Dokumentrepräsentation ist, wie oben ausgeführt, ein (gewichteter) Termvektor. Die Dokumente müssen in diese Repräsentationsform transformiert werden. Der erste Schritt dieser Transformation ist das sogenannte *Tokenizing*, im Deutschen auch Wortgrenzenerkennung genannt. Hiermit ist gemeint, dass ein Dokumente in seine einzelnen Worte oder Tokens (engl. für Zeichen) zerlegt wird, damit auf diese zugegriffen werden kann. Danach werden alle sogenannten *Stoppworte* entfernt. Stoppworte sind nicht bedeutungstragende Worte, wie zum Beispiel Artikel, Adjektive oder Adverbien. Diese Worte sind meist in Listen zusammengefasst und die Worte eines Dokumentes werden dann mit diesen Wortlisten verglichen und gegebenenfalls entfernt. Der letzte Schritt ist die Stammformreduktion oder englisch *stemming*. Die Worte werden hierbei auf ihre Stammform reduziert. Ziel dieses Verfahrens ist

es, die verschiedenen Flexions- und Derivationsformen eines Wortes zusammenzuführen (siehe [vR79]). So werden beispielsweise die Worte „computer“, „compute“, „computation“ und „computerization“ alle auf ihre Stammform „comput“ abgebildet. Die so reduzierten Worte sollen im Weiteren als Terme bezeichnet werden. Nachdem diese drei Vorverarbeitungsschritte abgeschlossen sind, ist die allgemeine Vorverarbeitung, wie sie für die meisten Verfahren der Clusteranalyse verwendet wird, beendet.

2.3.3 Probleme des Textclustering

Bevor im folgenden Abschnitt weitere Algorithmen zur Clusteranalyse erörtert werden, soll noch auf die Probleme eingegangen werden die speziell beim Text- oder Dokumentclustering auftreten und die die bisher (in 2.2) beschriebenen Algorithmen, beim Dokumentclustering aufweisen. Dabei soll auf ein allgemeines Dokumentclustering abgestellt werden. Die Besonderheiten beim Clustering von Artikeln einer Newsgroup werden im Kapitel 3 erläutert.

Ein grundlegendes Problem, mit dem sich jedes automatische System zur Textanalyse auseinandersetzen muss, ist die Komplexität und die Vielschichtigkeit der natürlichen Sprache. Damit sind Dinge wie hypotaktische Satzstruktur, Komposita [5], die Auflösung von Co-Referenzen [6] oder das Erkennen von Polysemen oder Homographen gemeint.

Bei der Clusteranalyse, so wie sie sich derzeit darstellt, wird diese Problemvielfalt zumindest eingeschränkt. Ein Text wird, wie oben beschrieben, als ein Vektor oder manchmal auch als eine Menge von Termen angesehen. Die Ähnlichkeit zweier Dokumente ergibt sich hier aus der Berechnung der Distanz zwischen ihren Termvektoren. Somit wird hier eine grammatikalische oder syntaktische Analyse außen vor gelassen. Die Information, die aus diesen Sprachbestandteilen gewonnen werden können, gehen dadurch aber verloren und einige Probleme bleiben auch nach wie vor bestehen. So können Synonyme oder auch Abkürzungen dazu führen, dass Texte die das gleiche oder zumindest ein ähnliches Thema behandeln, als unterschiedlich eingestuft werden. Polyseme und Homographen haben den gegenteiligen Effekt. Besonders bei recht kurzen Dokumenten können diese Phänomene sehr starke Bedeutung erlangen, unter anderem auch, weil sie zu sehr spärlich besetzten Termvektoren für Dokumente führen.

Neben diesen Problemen, die sich mit der Interpretation der inhaltlichen Ähnlichkeit von Dokumenten beschäftigen, ergibt sich aber auch ein Effizienzproblem. Der Termvektor eines Dokumentes kann unter Umständen sehr lang sein. Diese Dimensionalität wird zwar durch die im Abschnitt 2.3.2 beschriebenen Vorverarbeitungsschritte wie Stoppwortentfernung und Stemming reduziert, doch ist verbleibende Termanzahl der Dokumentenkollektion immer noch beträchtlich. Daraus folgt ein hoher Berechnungsaufwand für die Distanz zwischen zwei Dokumenten.

2.3.4 Weitere Verfahren der Clusteranalyse

Die bisher präsentierten Verfahren zur Clusteranalyse beruhen auf der Sichtweise, dass eine Menge von Objekten anhand der Ähnlichkeit ihrer Attributvektoren gruppiert wer-

[5] Diese mehrgliedrigen Ausdrücke wie „Donaudampfschifffahrtsgesellschaftbetreiber“ sind besonders in der deutschen Sprache sehr verbreitet.

[6] Co-Referenzen sind unterschiedliche Benennungen für ein Objekt. Zum Beispiel „Microsoft“ und MS oder „Gerhard Schröder“ und der „Bundeskanzler“

den soll. Im Sinne des Dokumentclusterings heißt das, eine Menge von Dokumenten zu gruppieren, in dem man ihre Termvektoren als Basis zur Ähnlichkeitsberechnung heranzieht. Diese Ansätze sind als *dokumentenorientiert* zu bezeichnen, was die verwendeten Clusterkriterien betrifft. Denn die Cluster ergeben sich mehr oder weniger direkt aus dem Verhältnis (Abstand- oder Ähnlichkeit) zwischen den einzelnen Dokumenten. Am klarsten ist dies bei „DBSCAN“ (siehe Abschnitt 2.2.2). Ein Cluster wird hier gebildet, indem alle zueinander *dichte-erreichbaren* Dokumente ausgewählt werden.

Die beiden folgenden Verfahren zur Clusteranalyse gehen auch von der Menge der Terme einer Dokumentenkollektion aus, jedoch ist ihr Clusterkriterium eher als *clusterorientiert* zu bezeichnen. Beide verwenden keine Termvektoren und daraus hergeleitete Distanzen zwischen den Dokumenten, um Cluster zu erstellen. Allerdings sind die Grenzen zwischen Cluster- und Dokumentorientiertheit fließend. Auf diesen Umstand wird an entsprechender Stelle hingewiesen. Dennoch soll dieses Merkmal für die Unterscheidung zwischen den bisher beschriebenen Verfahren und den folgenden verwendet werden. Auch sind die beiden Verfahren, insbesondere das zweite, hinführend zu dem in dieser Arbeit gewählten Ansatz.

2.3.4.1 Clustering mit Large Items

In ihrem Artikel „Clustering Transactions using Large Items“[WXL99] führen die Autoren aus, dass die herkömmlichen Clusteranalysemethoden bei spärlich besetzten Attributvektoren ihrer Ansicht nach nicht gut funktionieren, da paarweise Ähnlichkeit in diesem Fall nicht aussagekräftig sei.

Im Folgenden soll der alternative Ansatz der Autoren in [WXL99] beschrieben werden. Ihr Clusterkriterium ist weniger dokumentenorientiert, wie das bei den bisherigen Ansätzen der Fall war, sondern eher clusterorientiert. Sie gehen dabei von einem allgemeinen Clusterszenario aus. Das heißt, sie haben Objekte, die sie *transactions* nennen und Attribute die sie *items* nennen. Wir wollen diese Begriffe der Vollständigkeit halber gleich auf Dokumente und Worte bzw. Terme abbilden. Ein Dokument (transaction) ist dann eine Menge von Termen (items). Der Algorithmus, der das Clustering später durchführt, beruht auf einigen speziellen Begriffsdefinitionen.

- *MinSup*: Eine Zahl im Intervall $]0, 1]$, die den Mindestanteil von Dokumenten für ein *large term* festlegt.
- ***large term***: Ein Term, der in einer Anzahl von Dokumenten eines Clusters vorkommt, die größer (gleich) der festgelegten Mindestanzahl ist.
- $Large_i$ ist die Menge der *large terms* in Cluster C_i.
- ***small term***: Ein Term, der in einer Anzahl von Dokumenten eines Clusters vorkommt, die kleiner der festgelegten Mindestanzahl ist.
- $Small_i$ ist die Menge der *„small terms“* in Cluster C_i.

Ein *large term* in einem Cluster C_i ist also ein Term, der mindestens in k Dokumenten dieses Clusters vorkommt, wobei die Schranke k für einen Cluster C_i definiert ist durch: $k = MinSup * |C_i|$. Jeder andere Term, welcher weniger häufig in dem Cluster vorkommt,

ist ein *small term*. Als Clusterkriterium wird eine Kostenfunktion verwendet, die sich wie folgt definiert:

$$Intra(C) = \left| \bigcup_{j=1}^{k} Small_j \right| \tag{2.15}$$

$$Inter(C) = \Sigma_{j=1}^{k} |Large_j| - \left| \bigcup j = 1^k Large_j \right| \tag{2.16}$$

$$Cost(C) = w * Intra(C) + Inter(C) \tag{2.17}$$

In Formel 2.15 bezeichnet $Intra(C)$ also die Kardinalität der Vereinigungsmenge aller *small terms* in allen Clustern einer Partition C. Hiermit wird das Vorkommen von Termen bestraft, die in nicht ausreichender Kardinalität in den Clustern vorkommen. $Inter(C)$ (Formel 2.16) berechnet wiederum ein Maß für die Überlappungen von *large terms* zwischen den Clustern. Mit $Inter(C)$ wird also das Vorkommen von *large terms* in unterschiedlichen Clustern bestraft. Der Gewichtungsfaktor w in der Kostenfunktion in Formel 2.17 gibt an, welcher der beiden Kostenfaktoren stärker gewichtet werden soll.

Die Formel 2.17 wird als Clusterkriterium verwendet. Mit ihrer Hilfe werden die Dokumente der Kollektion Clustern zugewiesen. Das Clusterverfahren schließlich besteht aus zwei Phasen: einer Allokationsphase und einer Verfeinerungsphase. In der Allokationsphase wird die Dokumentenkollektion einmalig durchlaufen und jedes Dokument einem Cluster zugeordnet. Ein Dokument kann hierbei einem bestehendem Cluster zugeordnet werden oder einen Cluster neuen eröffnen - je nachdem bei welcher Aktion die Kosten $Cost(C)$ am geringsten sind. In der Verfeinerungsphase werden die Dokumente in umgekehrter Reihenfolge durchlaufen und für jedes Dokument wird überprüft, ob es in einem anderen Cluster geringere Kosten verursacht. Wenn dies der Fall ist, wird das jeweilige Dokument dem kostengünstigeren Cluster zugewiesen. Dieser Vorgang wird so oft wiederholt, bis in einem Durchlauf kein Dokument mehr neu zugewiesen wird.

Die beiden Phasen sind in den Abbildungen 2.3 und 2.4 in Pseudocode beschrieben. In der Allokationsphase wird, wie oben bereits ausgeführt, über alle Dokumente iteriert (Zeile 1) und jeweils der beste Cluster gesucht. Dies kann entweder ein bereits bestehender Cluster (Zeilen 3-8) oder ein eigens für dieses Dokument neu angelegter (Zeilen 9-12) Cluster sein. Die Menge der Cluster (*ClusterList*) ist zu Beginn der Phase leer. Für das erste Dokument wird dementsprechend in jedem Fall ein neuer Cluster angelegt.

Danach wird nur ein neuer Cluster für ein Dokument angelegt, wenn das Dokument, in diesen neuen Cluster verbracht, das kostenminimale Ergebnis erzielt. Ansonsten wird das Dokument einem der bereits bestehenden Cluster zugeordnet, in dem die Kosten minimal sind. Die Kostenfunktion wird in Formel 2.17 gezeigt, wobei im Pseudocode der Buchstabe **C** für die bestehende Partition (die Menge aller bestehenden Cluster mit ihren bereits zugewiesenen Dokumenten) steht.

Dieses Verfahren produziert eine flache Partitionierung der Dokumentenkollektion. Das Clusterkriterium ist clusterorientiert, da die zu minimierende Kostenfunktion über die Menge aller Cluster und die in ihnen enthaltenen *large* bzw. *small terms* definiert ist. Allerdings ist auch eine gewisse Ähnlichkeit mit dem *k-means* (siehe Abschnitt 2.2.1) Verfahren vorhanden, welches als dokumentenorientiert eingestuft wurde. So wird bei

```
DocumentList        := All Documents
ClusterList         := Empty

doAllocation()
1    For each Document ∈ DocumentList
2        BestResult := Infinity
3        For each Cluster ∈ ClusterList
4            If Cost(C with Document → Cluster) < BestResult then
5                BestResult := Cost(C with Document → Cluster)
6                BestCluster := Cluster
7            End If
8        End For
9        If Cost(C with Document → NewCluster) < BestResult then
10           BestCluster := NewCluster
11           ClusterList ∪ NewCluster
12       End If
13       Document → BestCluster
14   End For
```

Abbildung 2.3: Allokationsphase

```
DocumentList        := All Documents
ClusterList         := All in Allocation created Clusters

doRefinement()
1    BestResult := Cost(C)
2    Changed := true
3    Loop Changed
4        Changed := false
5        For each Document ∈ DocumentList
6            OldCluster := GetOldClusterFrom(Document)
7            NewCluster := GetBestClusterFor(Document)
8            If Cost(C with Document → NewCluster) < BestResult then
9                BestResult := Cost(C with Document → NewCluster)
10               OldCluster - Document
11               Document → NewCluster
12               Changed := true
13           End If
14       End For
15   End Loop
```

Abbildung 2.4: Verfeinerungsphase

k-means versucht, den durchschnittlichen Abstand der Dokumente in den Clustern zueinander zu minimieren. Über die *large* bzw. *small terms* wiederum wird im Grunde auch eine (gröbere) Distanzfunktion zwischen Dokumenten definiert, die genutzt wird, um Dokumente bestimmten Clustern zuzuweisen. Der Unterschied ist in den formalen Definitionen der Clusterkriterien zu sehen. Bei k-means wird die Formel 2.7 der quadratischen Abweichungen direkt über die Termvektoren, also die Repräsentation der Dokumente definiert. Bei Clustering via *large terms* ist das Clusterkriterium die Formel 2.17

über die Terme und ihre Häufigkeit in den Clustern definiert, also nur mittelbar über Dokumente.

Der Algorithmus beginnt mit einem *greedy-Clustering* der Dokumentenkollektion und versucht dieses später zu optimieren. Nachteil bei diesem Vorgehen ist allerdings, dass der Algorithmus in lokalen Optima konvergiert und abhängig ist von der Reihenfolge, in der die Dokumente eingelesen werden.

Ein Vorteil des Verfahrens ist, dass für jeden produzierten Cluster auch eine Labelung existiert, die direkt aus dem Verfahren resultiert, und zwar in Form der *large terms*.

2.3.4.2 Frequent Terms

Ein neuer Ansatz wählt einen ähnlichen Ausgangspunkt, wie der im vorherigen Abschnitt beschriebene. Allerdings nutzt dieser neue Ansatz das Konstrukt der häufigen Termmengen auf eine direktere Weise. Termmengen werden hier als Clusterkandidaten betrachtet und nicht als reine Kostenfaktoren. In zwei Papern (siehe [FWE03] und [BEX02])werden Clusteranalyseverfahren beschrieben, die diesen Ansatz nutzen. Jedoch soll nur das Verfahren „Frequent Term-Based Text Clustering“ aus [BEX02] beschrieben werden, da dieses Verfahren als Ausgangspunkt für das in dieser Arbeit dargestellte Verfahren verwendet wird.

Grundannahme des Ansatzes, den beide Verfahren anwenden, ist, dass es in Dokumentkollektionen Terme gibt, die eine größere inhaltliche Relevanz für diese Kollektion haben als andere und dass mithilfe dieser Terme eine alternative Clusteranalyse möglich ist. Diese inhaltlich aussagekräftigeren Terme und auch Termmengen, sind gerade die Term(-mengen), welche in der Dokumentenkollektion häufig vertreten sind. Für die Beschreibung des Verfahrens werden zuerst wieder ein paar Begriffe mit ihren Definitionen eingeführt:

- $D = \{D_1, \ldots, D_n\}$ sei die Menge aller Dokumente.
- $T = \{t_1, \ldots, t_k\}$ sei die Menge aller Terme.
- $D_j \subseteq T$ sei die Repräsentation eines Dokumentes durch die Menge seiner Terme.
- $MinSuport \in [0, 1]$ definiert die minimal notwendige Menge an Dokumenten, in denen eine Termmenge enthalten sein muss, um häufig zu sein - um ein *frequent term set* zu definieren.
- $cov(T' \subset T) = \{D_i \in D | T' \subseteq D_i\}$.
- $F = \{F_j \subseteq T | \, |cov(F_j)| \geq MinSuport * |D|\}$.

Die Definition $cov(T' \subset T)$ bezeichnet die Menge an Dokumenten, in denen eine bestimmte Termmenge enthalten ist. Eine Termmenge wiederum ist häufig, wenn sie in einer minimalen Anzahl an Dokumenten vorkommt. Eine solche häufige Termmenge, welche also den $MinSupport$ überschreitet, wird als *frequent term set* (abgekürzt fts) bezeichnet. Und F ist die Menge aller *frequent term set*. Zusätzlich soll ein aus genau k Termen bestehendes fts ein k-fts sein.

Die Konstruktion der fts kann mittels des so genannten *Apriori Algorithmus* durchgeführt werden. Dieser Algorithmus wurde ursprünglich für das Finden häufig vorkommender *itemsets* in Transaktionen verwendet. Mithilfe dieser *itemsets* können dann unter

anderem Assoziationsregeln hergeleitet werden (siehe [CLA04]). Der Algorithmus benutzt die Apriori-Eigenschaft, die besagt, dass jede nicht-leere Untermenge eines fts ebenfalls ein fts, also häufig ist (siehe [HK01]). Die Konstruktion der fts erfolgt ebenenweise. Die (k-1)-fts werden verwendet, um die k-fts zu erstellen, in dem aus den (k-1)-fts alle möglichen Kandidaten mit der Kardinalität k erstellt und diese dann daraufhin überprüft, ob eine beliebige Untermenge eines Kandidaten bereits nicht häufig ist. Wenn dies der Fall ist, kann dieser Kandidat von weiteren Betrachtungen ausgeschlossen werden. Für alle am Ende dieser Prozedur verbliebenen Kandidaten, wird dann ein kompletter Scan der Daten vorgenommen, um festzustellen, ob sie auch wirklich häufig sind. Der Algorithmus verwendet also eine *generate-and-test* Methode, um die Menge aller fts zu erzeugen. Kritikpunkt ist hier, dass er aber alle möglichen Kandidaten erst einmal erzeugen muss und dies sind unter Umständen recht viele. Ein Algorithmus, der dies umgeht, ist *FP-growth* (*Frequent-pattern growth* - siehe [HPY00]). Dieser Algorithmus benutzt einen *divide-and-conquer* Ansatz unter Verwendung einer kompakten Prefix-Baum Struktur. Besonders bei großen Datenkollektionen kann durch den Einsatz des FP-growth Algorithmus ein Effizienzvorteil erzielt werden.

Es gibt zwei interessante Eigenschaften der oben genannten Definition von *frequent term sets*. Die fts bilden mit der $\subseteq$-Funktion für Mengen einen vollständigen Verband. Das heißt für je 2 fts gibt es genau ein kleinstes gemeinsames Oberelement und ein größtes gemeinsames Unterelement. Auch gilt die Monotonie-Eigenschaft: Jede Teilmenge eines fts ist auch wiederum ein fts.

In [BEX02] werden nun Clustering Verfahren beschrieben, die auf der Menge der fts operieren: zum einen ein Verfahren, das ein flaches Clustering erzeugt. Aus diesem Verfahren, wird ein weiteres hergeleitet, welches eine Clusterhierarchie erstellt. Beide Verfahren können sowohl mit, als auch ohne Überlappungen in den erstellten Strukturen arbeiten. Ein Dokument kann also, je nach Wahl der Verfahrensvariante, sowohl in genau einem Cluster[7], als auch in mehreren Clustern vorhanden sein. Da in dieser Arbeit ein Verfahren eingeführt werden soll, das eine hierarchische Clusteranalyse mit Überlappungen durchführt, wird zur Beschreibung des Clusterverfahrens in [BEX02] auch diese Variante ausgewählt, um spätere Vergleiche zu vereinfachen.

Ausgangspunkt für das Clustering einer Dokumentenkollektion ist die Menge der *frequent term sets*. Jedes fts definiert einen Clusterkandidaten. Durch die Kardinalität der Termmenge eines Clusterkandidaten, die ihn definiert, ist auch gleichzeitig seine Tiefe in der Clusterhierarchie vorgegeben. Hat ein Kandidat einen Term, der ihn definiert, so wird er auf der Ebene unter der Wurzel angesiedelt (1.Ebene), hat er zwei Terme, bekommt er einen Platz auf der Ebene darunter (2.Ebene), usw. Die Kandidaten auf der ersten Ebene sind trivialerweise die Kandidaten für die Wurzel. Mit den Kandidaten auf den weiteren Ebenen, verhält es sich etwas anders. Sie sind jeweils die Kandidaten für Cluster(-kandidaten) aus der ersten Ebene, allerdings nur für solche, deren Termmengen Untermengen ihrer eigenen Termmenge sind. Ein Clusterkandidat mit der Termmenge $\{A, B\}$ kann also nur Kandidat für die Cluster(-kandidaten) mit den Termmengen $\{A\}$ und $\{B\}$ sein. Formalisiert lautet die Definition für die Menge der Clusterkandidaten eines Cluster(-kandidaten):

- $N(F_i) = \{F_j \in F | F_j \subset F_i \wedge |F_i - F_j| = 1\}$ mit $F_i \in F$

[7] Bei der hierarchischen Variante jeweils in genau einem Cluster auf einer Hierarchie-Ebene.

Wobei F die Menge aller fts und somit auch die Menge aller Clusterkandidaten bezeichnet. Die Bezeichnung F_i steht also zum einen für eine Termmenge aus der Menge aller Termmengen F und zum anderen für den Clusterkandidaten, den diese Termmenge definiert.

Zunächst soll das Clustering der Dokumentmenge eines Clusters beschrieben werden, da dies für die hierarchische Variante auch als Grundalgorithmus dient.

Das Verfahren zur Auswahl der Cluster aus der Kandidatenmenge hat folgenden Grundgedanken: Die gewählten Cluster sollen minimale Überlappungen (in den Dokumenten) aufweisen. Es soll also eine überlappungsminimale Überdeckung der Dokumentmenge erreicht werden. Der theoretische Hintergedanke ist, dass eine Termmenge, die möglichst wenig Schnitt (in den Dokumenten) mit anderen Termmengen hat, die in ihr enthaltene Dokumentenmenge am besten charakterisiert. Es soll im Folgenden das (flache) Clustering einer Dokumentmenge eines beliebigen Cluster(-kandidaten) betrachtet werden. Um die Beschreibung zu vereinfachen, sei der Cluster mit C und seine Kandidatenmenge mit $N(C) = \{C_1, C_2, ..., C_n\}$ bezeichnet. Die Auswahl aus den Kandidaten findet iterativ statt. Nach jeder Iteration wird der jeweils gewählte Cluster aus der Menge der Kandidaten entfernt, bis alle Dokumente durch mindestens einen der gewählten Cluster abgedeckt sind. Die Menge $N(C)$ verringert sich also mit jeder Iteration um einen Kandidaten.

Um nun den Begriff der Überlappung besser handhabbar zu machen, wird zuerst eine Funktion f für die Dokumente D_i eines Cluster C definiert.

$$f(D_i) = |\{C_i \in N(C) | D_i \in C_i\}| \tag{2.18}$$

Die Funktion f gibt also Kardinalität der Menge der Clusterkandidaten wieder, in denen ein Dokument enthalten ist. Die Überlappung eines Clusterkandidaten mit den anderen Clusterkandidaten ist somit

$\sum_{D_j \in C_i} (f(D_j) - 1)$ mit $C_i \in N(C)$

In [BEX02] definieren die Autoren weiterhin den *Standard Overlapp* (die Standardüberlappung) eines Clusters wie folgt:

$$SO(C_i) = \frac{\sum_{D_j \in C_i} (f(D_j) - 1)}{|\{D \in C_i\}|} \tag{2.19}$$

Die Standardüberlappung eines Clusterkandidaten (kurz SO) ist also die durchschnittliche Überlappung pro Dokument. Durch die Normierung der absoluten Überlappung eines Clusterkandidaten auf seine Dokumentenanzahl werden kleinere Clusterkandidaten nicht so stark benachteiligt.

Die Abbildung 2.5 zeigt die Prozedur für das flache Clustering in Pseudocode. Die äußere Schleife (Zeile 1) wird solange durchlaufen, bis alle Dokumente abgedeckt sind. Hierbei sind alle Dokumente gemeint, die durch die Clusterkandidaten abgedeckt werden können. Es wird also nicht Bezug auf die Dokumentmenge des Clusters C genommen, da es natürlich sein kann, dass die Kandidaten seine Dokumentmenge nicht vollständig abdecken können. Die Funktion **cov**(*ChosenClusterList*) gibt genau die Anzahl der durch die bisher gewählten Clusterkandidaten abgedeckten Dokumente zurück. In der inneren Schleife (Zeilen 3 - 8) werden alle noch verbliebenen Kandidaten durchlaufen und der

```
ClusterCandidateList   := All candidates N(C) from cluster C
DocumentNumber         := Number of all documents from C which can be covered by the
                          set of candidates N(C)
ChosenClusterList      := Empty

doFlatClustering()
1    Loop cov(ChosenClusterList) ≠ DocumentNumber
2        BestResult := Infinity
3        For each Candidate ∈ ClusterCandidateList
4            If SO(Candidate) < BestResult then
5                BestResult := SO(Candidate)
6                BestCandidate := Candidate
7            End If
8        End For
9        ChosenClusterList ∪ Candidate
10       ClusterCandidateList / Candidate
11   End Loop
```

Abbildung 2.5: Flaches Clusterverfahren

Kandidat mit der geringsten Standardüberlappung ausgewählt. Der so gewählte Kandidat wird der Liste der gewählten Cluster (*ChosenClusterList*) zugewiesen und aus der Liste der Kandidaten (*ClusterCandidateList*) entfernt. Wenn alle Dokumente abgedeckt sind, enthält *ChosenClusterList* die Menge der gewählten Cluster.

Das hierarchische Clusterverfahren geht nun Ebenenweise und *top-down* vor. Zuerst werden aus der Menge der Clusterkandidaten auf der ersten Ebene (also aus allen 1-fts), mittels des flachen Clusterverfahrens die *besten* Clusterkandidaten selektiert. Danach wird für jeden dieser gewählten Cluster, dieses Verfahren auch auf seine Clusterkandidaten angewandt. Und so wird weiter verfahren, bis die Cluster erreicht sind, welche selbst keine Kandidaten mehr haben.

Die so erstellte Hierarchie ist eine Quasihierarchie mit einem weiteren, besonderen Merkmal. Ein Cluster kann auch mehr als einen Eltern-Cluster haben. Dies ist dadurch bedingt, dass ein k-fts aufgrund der Verbandsstruktur der Termmengen und der Monotonieeigenschaft immer $k-1$ Vorgänger hat.

Das Clusterverfahren ist stark clusterorientiert, da aus einer Menge von Clusterkandidaten eine Teilmenge selektiert wird und dies anhand eines Maßes, der Standardüberlappung aus Formel 2.19, das auf dem Verhältnis der Kandidaten untereinander beruht.

Ein Vorteil dieses Clusterverfahrens ist, dass die hohe Dimensionalität und auch die spärliche Besetzung der Termvektoren der Dokumente hier kaum eine Rolle spielen. Die Dimensionalität wird auf die Anzahl der häufigen Terme reduziert. Es wird auch nicht mehr der Abstand zweier Dokumente verwendet, welcher bei spärlicher Besetzung der Termvektoren zu Problemen führen kann. Statt dessen gelangen zwei Dokumente in denselben Cluster, wenn dieser eine aussagekräftige Beschreibung seiner Dokumentmenge liefert (geringe Überlappung mit anderen Clustern) und die beiden Dokumente diese Beschreibung (Termmenge des Clusters) beinhalten. Ein weiterer Vorteil ist die implizite Clusterbeschreibung anhand der *frequent term sets*. Diese bietet einen ersten Eindruck

des Inhaltes der Cluster.

Allerdings sind mit dem Verfahren in der oben beschriebenen Form auch einige Probleme verbunden. So werden alle Terme, die den *MinSupport* überschreiten, für das Clustering verwendet. Doch ist es fraglich, ob ein Term, der einfach nur sehr häufig vorkommt, auch inhaltlich relevant für die Domäne einer Dokumentenkollektion ist. Ein weiterer Nachteil ist die Voraussetzung, dass alle Dokumente abgedeckt werden müssen. In einer Menge von Dokumenten befinden sich häufig thematische Ausreißer. Diese werden mit hoher Wahrscheinlichkeit in nur wenigen fts enthalten sein[8], müssen aber dennoch abgedeckt werden. Dies kann dazu führen, dass Clusterkandidaten mit hoher Überlappung gewählt werden müssen, was das Ergebnis verschlechtert.

Auf die Eigenschaften und Nachteile des hier geschilderten Verfahrens wird in Kapitel 5 weiter eingegangen.

[8] Dieser Effekt ist durchaus wünschenswert, da so in gewissem Maße auch eine Ausreißererkennung möglich ist.

3 Problemstellung

Nach der Einführung in den Bereich der Clusteranalyse im vorherigen Kapitel soll nun die Aufgabenstellung dieser Arbeit konkretisiert werden. Dazu wird zunächst eine kurze Erläuterung zu der Struktur der dort benutzten Dokumente, also den Artikeln einer Newsgroup gegeben. Dann soll auf die speziellen Probleme und Anforderungen eingegangen werden, die diese Struktur und die generellen Verfahrensweisen in einer Newsgroup an eine Clusteranalyse stellen.

3.1 Thread - Artikel - Dokument

In der Einleitung wurde von den *Artikeln* einer Newsgroup gesprochen. Beim Textclustering ist im Allgemeinen von Dokumenten die Rede. Beide Begriffe sind aber in diesem Zusammenhang nicht ganz zutreffend, da sich zum einen der Artikel in einer Newsgroup mit der Zeit verändern kann und da in den meisten Fällen mehrere Autoren in einem interaktiven Prozess an seiner *Erstellung* beteiligt sind[1]. Besser wird man bei einem Newsgroup Artikel von einem Thread sprechen können. Ein Thread (Diskussionsfaden) wird von einem Autoren durch einen Beitrag eröffnet (Eröffnungsbeitrag). Oft ist das Ziel des Autoren, die Antwort auf eine Frage oder die Lösung für ein spezielles Problem zu erhalten. Manchmal ist es aber auch die Eröffnung einer Diskussion zu einem speziellen Thema. Diese Zielsetzungen variieren von Newsgroup zu Newsgroup. Nach dem Eröffnungsbeitrag kann der Thread von anderen Autoren erweitert werden und zwar indem sie auf den Eröffnungsbeitrag oder auf einen späteren Beitrag antworten. Auch kommt es häufiger vor, dass der Autor des Eröffnungsbeitrages seine Frage noch näher spezifiziert und auf den eigenen Eröffnungsbeitrag (oder einen Antwortbeitrag) antwortet. Wie gesagt, ist die Bezeichnung Artikel oder Dokumente für die Threads einer Newsgroup etwas missdeutig. Dennoch sollen diese Begriffe verwendet werden, um die Objekte in einer Newsgroup (Artikel) und die Objekte der Clusteranalyse (Dokumente) besser fassen zu können.

Der Artikel einer Newsgroup besteht aus einem oder mehreren einzelnen Beiträgen. Ein Beitrag stellt die atomare Einheit dar. Ein Thread oder Artikel setzt sich aus einem oder mehreren Beiträgen zusammen, die ein gemeinsames Thema behandeln. Im Kapitel 4 wird unter anderem beschrieben wie aus einem solchen Artikel durch die Vorverarbeitung ein Dokument erstellt wird.

Definition 3.1.1 (**Artikel** und **Dokument**)**.** Ein *Artikel* ist also eine Aneinanderreihung von Beiträgen, ein unbehandeltes Dokument, das durch die Vorverarbeitung in die gewünschte Form transformiert wird. Der Begriff *Dokument* soll dann für das Resultat der Vorverarbeitung verwendet werden, da es das Objekt bezeichnet, welches in das Clusterverfahren eingeht.

[1] Die Bezeichnung *Artikel* wird häufig auch für einzelne Beiträge verwendet und dann wieder für die Zusammenfassung mehrerer Beiträge zu einem Thread.

Die Beziehung zwischen den Begriffen *Beitrag*, *Artikel* und *Dokument* wird in Abbildung 3.1 veranschaulicht.

Abbildung 3.1: Transformation atomarer Beiträge in Dokumente

3.1.1 Probleme des Clustering von Newsgroup Artikeln

In Abschnitt 2.3.3 wurden die Problematiken des Textclustering im allgemeinen erläutert. Hier sollen jetzt die speziellen Probleme die sich beim Clustering von Newsgroup Artikeln ergeben, beschrieben werden.

Generell kann in den Texten einer Newsgroup mit einer erhöhten Fehlerrate in der Rechtschreibung gerechnet werden, seien es Flüchtigkeitsfehler oder auch die Unkenntnis der richtigen Schreibweise. Die Folgen sind ähnlich denen, die sich aus Synonymen ergeben (siehe Abschnitt 2.3.3). Es werden Unterschiede zwischen Dokumenten erzeugt die gar nicht da sind.

Eine weitere Schwierigkeit kann sich aus der sehr stark variierenden Länge der Artikel ergeben. Ein Artikel kann nur einen Satz oder aber auch mehrere Seiten beinhalten. Daraus folgen sehr unterschiedlich stark besetzte Termvektoren und die Qualität eines inhaltlichen Vergleiches ist dadurch nur schwer zu gewährleisten.

Zwei weitere Eigenarten der Artikel einer Newsgroup sind zum einen die Wiederholung von Vorgängerbeiträgen eines Threads und zum anderen das Verwenden von prosafremden Textfragmenten[2]. Im ersten Fall wird der gesamte Text eines Vorgängerbeitrages oder auch nur Teilauszüge daraus in den eigenen Beitrag kopiert. Dies beeinflusst Clusterverfahren, die mit Wortgewichtungen arbeiten, die auf der Häufigkeit von Worten basieren. Prosafremde Textfragmente blähen die Dimension der Wortvektoren weiter auf, da zum Beispiel die in einer Programmiersprache benutzten Terme (Befehle, Variablen, etc.) nicht durch die üblich Vorverarbeitungsschritte wie Stopwortentfernung (siehe 4.2) herausgefiltert werden können und doch kaum etwas zur thematischen Eingrenzung des Artikelinhaltes beitragen.

Als letztes ist noch zu nennen, dass ein geringer Anteil der Autoren einen großen Anteil an Artikeln schreibt und häufig verwenden diese Autoren einen Standardsatz oder gar Standardtext als Abschluss ihres Beitrages. Da nun diese Autoren in einem verhältnismäßig großen Anteil aller Artikel mindestens einen Beitrag verfasst haben, taucht auch ihr Standardsatz in einem dementsprechenden Anteil aller Artikel auf und erzeugt eine thematische Ähnlichkeit, die vielfach nicht vorhanden ist. Besonders bei einer kleineren Artikelkollektion kann dies zu einer Verfälschung des Ergebnisses führen.

3.2 Szenario und Anforderung an die Clusteranalyse für Newsgroupartikel

Bevor im nächsten Kapitel ein neues Verfahren zur Clusteranalyse von Artikeln einer Newsgroup vorgestellt wird, soll hier ein Szenario beschrieben werden. Von diesem Sze-

[2] Beispielsweise werden in Newsgroups, die Programmierung zum Inhalt haben, Codefragmente in den eigentlichen Text integriert.

nario ausgehend, sollen die daraus resultierenden Anforderungen an das Verfahren der Clusteranalyse erläutert werden.

3.2.1 Szenario

Ausgangspunkt der Betrachtung ist die Tätigkeit des Administratoren einer Newsgroup. Dieser verwaltet und moderiert eine Newsgroup. Das heißt, er sieht sich beispielsweise einzelne Artikel an und überprüft, ob diese thematisch in die Newsgroup passen und löscht oder kommentiert diese gegebenenfalls. Teilweise nimmt er auch selbst an Diskussionen teil oder beantwortet Fragen. Des Weiteren erstellt er FAQs', um Overhead zu vermeiden.

Allerdings werden unter Umständen mehr als hundert Artikel täglich neu eröffnet, so dass die Kontrolle jedes Einzelnen sehr zeitaufwändig ist. Auch das Erstellen von FAQs' ist mit nicht minderem Zeitaufwand behaftet, da sich die Domäne der Newsgroup thematisch mit der Zeit wandelt und neue thematische Schwerpunkte hinzu kommen und alte obsolet werden.

Der Administrator überlegt sich nun, dass er eine weitere Zugriffsstruktur für die Artikel seiner Newsgroup haben möchte. Diese soll zum einem ihm selbst gestatten, einen schnelleren Überblick über neue Artikel zu erlangen. Die Identifikation domänenfremder Artikel sollte beispielsweise erleichtert werden. Und zum anderen soll sie es den Besuchern seiner Newsgroup ermöglichen, schneller Information zu den sie interessierenden Fragen zu erhalten. Als Lösung wählt er eine hierarchisch-thematisch gegliederte Zugriffsstruktur. Allerdings möchte er den Umstand vermeiden, diese selbst erstellen und manuell pflegen zu müssen, da dies unter anderem auch erfordern würde, dass er sich eben jeden einzelnen neuen oder auch geänderten Artikel ansieht.

Da er aber weiß, dass viele Besucher seine Newsgroup regelmäßig frequentieren, möchte der Administrator die Struktur möglichst konstant halten. Damit soll die Orientierung in der Newsgroup erleichtert werden. Ein sich ständig ändernde Zugriffsstruktur könnte die Besucher der Newsgroup verwirren und dazu führen, dass die Struktur nicht angenommen wird.

3.2.2 Anforderungen

Eine hierarchisches Clustering erfüllt die grundlegenden Voraussetzungen für die oben geschilderten Ansprüche des Administratoren. Allerdings sollen die speziellen Anforderungen hier noch einmal aufgelistet werden.

- Erzeugung einer thematischen Hierarchie.
- Cluster-Label: Der Administrator möchte für jedes Cluster, das erstellt wird, zumindest einen groben Hinweis auf den Inhalt des betreffenden Cluster haben.
- Überdeckendes Clustering: Ein Dokument darf in mehreren Kind-Clustern des gleichen Eltern-Clusters vorkommen.
- Inkrementelles Verhalten:
  - Neue Dokumente sollen automatisch eingeordnet werden.
  - Änderungen in Artikeln (neue Beiträge zu einem bestehenden Artikel) sollen berücksichtigt werden.

- Bei Bedarf sollen neue Cluster entstehen und alte wegfallen.
- Die Änderungen an der Struktur sollen minimal gehalten werden.

Die oben angegebenen Punkte stellen nur eine grobe Beschreibung dessen dar, was das Clusterverfahren gewährleisten soll. Der Schwerpunkt dieser Arbeit wird auf das inkrementelle Arbeiten des gewählten Verfahrens gesetzt. Das heißt: Wie können neue Dokumente aufgenommen und den Änderungen an bestehenden Dokumenten Rechnung getragen werden, unter der Prämisse, dass die sich daraus ergebenden strukturellen Änderungen an der Hierarchie abgebildet werden? Allerdings soll dies nicht bedeuten, dass mit jedem Dokument, das neu in die Artikelkollektion aufgenommen wird, die Hierarchie neu erzeugt wird. Dieser Vorgang würde bedeuten, dass der Administrator und die Besucher der Newsgroup sich ständig einer neuen Struktur gegenüber sehen würden. Die inkrementelle Komponente des Verfahrens muss also thematisch strukturelle Änderungen Abbilden können um die Güte der Hierarchie gewährleisten zu können, aber auch der Tatsache Rechnung tragen, dass für den Administrator und die Besucher jede Veränderung an der Hierarchie eine zeitweise Verminderung der Übersichtlichkeit bedeutet.

Für das initiale Clustering bietet sich das in Abschnitt 2.3.4.2 vorgestellte Verfahren des Clusterings mit *frequent term sets* an. Es erstellt eine Quasihierarchie (es ist also hierarchisch und ein Dokument kann in mehreren Clustern, desselben Eltern-Clusters vorkommen). Des Weiteren hat jeder Cluster eine Bezeichnung, in Form der häufigen Terme die diesen Cluster definieren, die einen ersten Eindruck seines Inhaltes vermittelt.

4 Vorverarbeitung

Um das Verfahren evaluieren zu können, ist es in einem Prototyp implementiert. Für diese Implementierung wurden Einschränkungen in Bezug auf die zur Evaluation benutzten Newsgroups getroffen. Diese Einschränkungen werden im folgenden Abschnitt 4.1 beschrieben. In einem weiteren Abschnitt (4.2) wird die allgemeine Vorverarbeitung der Dokumente bzw. Beiträge einer Newsgroup erläutert, wie sie im Prototyp implementiert ist. Die Vorverarbeitung und die Einschränkungen sollen an dieser Stelle beschrieben werden, weil sie sowohl für das initiale Clustering, welches in Kapitel 5 erläutert wird, als auch für die inkrementelle Erweiterung in Kapitel 6 notwendige Voraussetzungen sind.

4.1 Einschränkungen der Implementierung

Für die Implementation des Prototypen werden zwei Einschränkungen gemacht, um die für die eigentliche Clusteranalyse unwesentlichen Schritte der Vorverarbeitung so einfach wie möglich zu halten. So werden die Artikel einer Newsgroup, oder besser die einzelnen Beiträge (siehe 3.1), über das NNTP-Protokoll geladen. Das heißt, es können nur Newsgroups berücksichtigt werden, welche über dieses Protokoll arbeiten. Da aber eine große Auswahl von thematisch unterschiedlichen Newsgroups über dieses einfach zu handhabende Protokoll verfügbar sind, stellt diese Einschränkung keinen Nachteil dar. Prinzipiell ist es möglich, den Prototypen auch auf Foren oder Newsgroups im WorldWideWeb, die also über das HTTP-Protokoll und HTML laufen, anzuwenden. Allerdings müssten für diese jeweils spezifische HTML-Wrapper geschrieben werden, die die Extraktion der einzelnen Beiträge vornehmen würden. Des Weiteren ist das automatisierte Laden einer großen Anzahl von Beiträgen aus diesen Foren nicht gern gesehen, da es zu einer hohen Auslastung des betreffenden Servers führt.

Die zweite getroffene Einschränkung betrifft die Sprache der gewählten Newsgroups. Es werden nur englischsprachige Newsgroups betrachtet, da es für diese gute und effiziente Stemming-Algorithmen gibt. Der Grund hierfür liegt darin, dass die englische Sprache weit weniger flektiert als die deutsche ist und Algorithmen zur Grund- und Stammformbestimmung eine höhere Genauigkeit erreichen (siehe [Fuh03]).

4.2 Allgemeine Vorverarbeitung

Wie in Abschnitt 3.1 beschrieben, bestehen die Artikel einer Newsgroup aus einem oder mehreren einzelnen Beiträgen, die zu einem Thread zusammengefasst werden. Diese Artikel werden dann durch die Vorverarbeitung, die in den kommenden drei Unterabschnitten beschrieben wird, in Dokumente umgewandelt. Diese Dokumente werden dann als Basis-Objekte für die Clusteranalyse verwendet. Allerdings werden die Dokumente nicht direkt als Eingaben für die Clusteranalyse verwendet. Es wird eine Term-Dokument-Darstellung verwendet, in der nicht nur einem Dokument seine Terme die es enthält kennt, sondern auch umgekehrt. Die Term-Dokument-Sicht, also die Sicht der Terme auf ihre Dokumente, ist die wichtigere von beiden. Da das in dieser Arbeit beschriebene Clusterverfahren

auf häufig in der Dokumentenkollektion vorkommenden Termen beruht. Des Weiteren wird in Unterabschnitt 4.2.5 beschrieben wie aus der Menge der Terme, die für die Clusteranalyse und ihre Inkrementierung relevanten Terme selektiert werden.

4.2.1 Entfernung von Beitragswiederholungen

Da es, wie in 3.1.1 erläutert, üblich ist, bei einem Kommentar auf einen Beitrag den Vorgängertext mit aufzuführen und dies zu Verfälschungen bei der Termgewichtung führen kann, werden diese in einem ersten Vorverarbeitungsschritt entfernt[1]. Hierzu wird der Umstand genutzt, dass eine solche Wiederholung zeilenweise von mindestens einem „>“ angeführt wird. Anhand dieses Zeichens können also Wiederholungen erkannt und auch entfernt werden.

4.2.2 Entfernung von Standardtexten

In den Beiträgen von Newsgroups verwenden die Verfasser häufig einen standardisierten Textfuß, welcher eine Grußformel oder auch einen Lieblingsspruch beinhaltet. Beispielsweise sind viele Beiträge einer Newsgroup, die sich der Applikation *MS Access* widmet, von einem speziellen Autoren, welcher als Abschluss immer die Zeile „26 + 6 = 1 It's Irish Math“ seinen Beiträgen anfügt. Und da er verhältnismäßig viele Beiträge verfasst, kommt auch das Wort *Irish* in vielen Artikeln der Newsgroup vor. Da der hier verwendete Ansatz auf der Dokumenthäufigkeit eines Terms innerhalb der Gesamtdokumentenkollektion basiert, führen solche Standardtexte dazu, dass die Resultate, die das Verfahren liefert, verfälscht werden können. Der Grund hierfür ist, dass Terme als häufig eingestuft werden und damit einen Einfluss auf das Verfahren erhalten, welcher ihrer Bedeutung nicht entspricht. So ist der Term *Irish* mit Sicherheit kein thematisch signifikanter Term für eine Access Newsgroup. Der Einfluss solcher Standardtexte muss allerdings nicht so klar erkennbar sein. So gibt der Verfasser auch häufig die Adresse seiner Homepage im Textfuß an. In dieser ist dann das Protokollkürzel *http* enthalten. In einer Informatik-Newsgroup, welche beispielsweise Netzwerkprotokolle zum Thema hat, würde dies zu einer unverhältnismäßig hohen Häufigkeit des Termes *http* führen. Allerdings ist dieser Term auch thematisch signifikant und kann nicht ausgelassen werden. Aus diesen beiden Beispielen geht hervor, dass es angezeigt ist, Standardtexte wenn möglich zu entfernen, um die oben beschriebenen Fehlgewichtungen von Termen zu vermeiden. Dies geschieht mittels regulärer Ausdrücke. Um diese Ausdrücke erstellen zu können, ist natürlich eine zumindest stichprobenartige Inspektion der Newsgroupbeiträge notwendig. Der Prototyp bietet die Möglichkeit eine Liste von regulären Ausdrücken anzugeben, die dann benutzt werden, um Standardtexte zeilenweise zu entfernen.

4.2.3 Dokumenterstellung

Wenn die Grundbereinigung, welche in den letzten beiden Abschnitten beschrieben wurde, beendet ist, werden allen Beiträgen, die zu einem Artikeln gehören, zu einem Dokument zusammengefasst. Hierbei ist die Reihenfolge der Beiträge innerhalb des Dokumentes ohne Belang. Als letztes wird dem Dokument noch das Artikelthema (der Betreff

[1] Das hier vorgestellte Clusterverfahren nutzt nicht direkt die Termgewichtung. Jedoch kann sie bei einem Termauswahlschritt angewandt werden.

oder das Subject) hinzugefügt. Dies wird getan, weil im Betreff die Grundproblematik des Artikels mittels weniger Schlagwörter erfasst wird. Die Wörter des Betreffs beschreiben das Thema also oft sehr prägnant und werden aber nicht immer im eigentlichen Text wiederholt. Die in der Abbildung 3.1 skizzierte Transfomation der Artikel in Dokumente ist nun abgeschlossen.

4.2.4 Erstellung der Termvektoren

Nachdem so zusammenhängende Dokumente erstellt wurden, wird auf diese ein Tokenizer angewandt, der die Texte in einzelne Tokens[2] zerlegt. Im nächsten Schritt werden sogenannte Stoppworte entfernt. Dies sind in erster Linie nicht-bedeutungstragende Worte wie Artikel, Füllwörter, etc (siehe 4.2).

Die verbleibenden Worte werden auf ihre Stammformen reduziert. Hierfür wird ein Porter-Stemming-Algorithmus eingesetzt (siehe [Por80]). Die so transformierten Worte werden im Weiteren als Terme bezeichnet. In den meisten auf Termvektoren basierenden Clusterverfahren genügt es, eine gerichtete Beziehung von einem Dokument auf seine Terme zu verwenden. Da das hier angewendete Verfahren auf der Dokumenthäufigkeit eines Terms beruht und aus den häufigen Termen die Clusterkandidaten hervorgehen, wird an dieser Stelle eine bidirektionale Beziehung zwischen Dokumenten und Termen erstellt, um die spätere Arbeit zu erleichtern und effizienter gestalten zu können.

4.2.5 Auswahl der relevanten Terme

Der letzte Vorverarbeitungsschritt ist die Auswahl der für das Clusteranalyseverfahren relevanten Terme. Diese Auswahl kann im Prototyp einerseits vom Anwender selbst getroffen werden oder andererseits durch die Angabe einer oberen und einer unteren Schranke für die Dokumenthäufigkeit eines Terms auch automatisch erfolgen. Für die Selektion durch den Anwender spricht die hohe Quote an Termen, die durch Standardtexte in ihrer Häufigkeit mindestens beeinflusst werden[3]. Auch kann bei dem hier vorgestellten Clusteranalyseverfahren die gezielte Termselektion zu einer qualitativen Verbesserung führen. Um die Selektion durchzuführen, wird dem Anwender die Liste aller Terme angezeigt. Da es sehr viele Terme geben kann und die meisten nicht in Betracht kommen, kann die Termliste auf verschiedene Weisen sortiert werden, um dem Anwender eine Entscheidungshilfe zu bieten. Die einfachste Sortierung ist die nach der Häufigkeit der Terme. Allerdings ist diese nur begrenzt aussagefähig für die inhaltliche Relevanz der Terme für die Domäne der Newsgroup. Daher werden noch zwei weitere Sortierungsoptionen angeboten. Bei der Ersten wird mithilfe der tfxidf Indexierung (siehe Abschnitt 2.3) eine Termgewichtung berechnet, welche die inhaltliche Relevanz eines Terms wiedergeben soll. Für jeden Terme der Dokumentenkollektion wird seine tfxidf Gewichtung für jedes Dokument, in dem er vorkommt, berechnet. Die einzelnen Dokumentgewichtungen eines Terms werden aufaddiert. Formalisiert wird dies in der Formel 4.1. $TW(t_j)$ steht hierbei für diese summierte Gewichtung eines Terms t_j in der Dokumentenkollektion D.

[2] Ein Token ist hier ein Term oder Wort.

[3] Selbst bei guter automatischer Bereinigung der Newsgroup Artikel durch reguläre Ausdrücke verbleibt ein gewisser Anteil an Standardtexten in den Dokumenten.

$$TW(t_j) = \sum_{d_i \in D \wedge t_j \in d} tfxidf(t_j, d_i) \tag{4.1}$$

Um die Gewichtungen auf das Intervall [0,1] zu normieren, werden sie durch die höchste Termgewichtung dividiert.

$$NTW(t_j) = \frac{TW(t_j)}{max\{TW(t_i)|t_i \in T\}} \tag{4.2}$$

Die Terme können nun anhand dieser Termgewichtung sortiert werden. Für die zweite Sortiermethode nach inhaltlicher Relevanz wird dem eigentlichen Dokumentenkorpus der Newsgroup ein zweiter domänenfremder Korpus zugefügt. Dieser dient dazu, die idf-Komponente der Terme anzupassen. Der idf-Wert eines Terms wird mit steigender Dokumenthäufigkeit geringer. Die Idee ist nun durch die Hinzunahme eines domänenfremden Korpus, die Terme der betrachteten Newsgroup, die auch häufig in diesem Referenzkorpus auftreten, geringer zu gewichten als Terme, die nicht oder nur sehr selten dort vorkommen. Wenn ein Term, der beispielsweise in einer technischen Newsgroup sehr häufig ist, dies auch in einer Newsgroup für Fußball ist, so ist er vermutlich inhaltlich nicht so relevant für die technische Newsgroup, wie dies seine Häufigkeit nahe legt. Die idf-Komponenten der Terme werden nun entsprechend modifiziert, die Berechnung der Termgewichtung entspricht ansonsten den Formeln 4.1 und 4.2. Die absteigende Sortierung nach dieser korpus-basierten Termgewichtung gibt dem Anwender nun eine gute Entscheidungshilfe, welche Terme domänenrelevant sind und er kann die auswählen, welche für die Clusteranalyse verwendet werden sollen.

5 Initiales Clustering

In diesem und dem nächsten Kapitel soll das Konzept und die Umsetzung einer inkrementellen hierarchischen Clusteranalyse für die Artikel bzw. Dokumente einer Newsgroup beschrieben werden. Das hier vorgestellte Verfahren kann in zwei unterschiedliche Phasen eingeteilt werden. In der ersten Phase wird das initiale Clustering durchgeführt. Dieses Clustering erstellt eine hierarchische Struktur für einen gegebenen Dokumentencorpus. In der zweiten Phase wird diese Cluster-Hierarchie inkrementell erweitert. D. h. es werden weitere Beiträge bzw. Dokumente hinzugefügt und die Cluster-Hierarchie wird angepasst. In diesem Kapitel soll der initiierende Teil des Verfahrens erläutert werden.

Als Grundlage dient eine Struktur oder Hierarchie von *frequent term sets* (fts). In 2.3.4.2 wurden zwei Ansätze kurz erläutert, die ebenfalls mit *frequent term sets* bzw. *frequent item sets* arbeiten. Das hier präsentierte Verfahren weicht jedoch von diesen Ansätzen in vielerlei Hinsicht ab. Aus diesem Grund und auch um das hier dargestellte Verfahren besser beschreiben zu können, wird auf die Struktur und auf die Konstruktion der fts in den Abschnitten 5.1 und 5.3.1 genauer eingegangen. Ein großer Unterschied zu den in Abschnitt 2.3.4.2 beschriebenen Verfahren, ist das hier benutzte Kostenmaß bzw. Clusterkriterium. Dieses wird in Abschnitt 5.2 erläutert. Im Abschnitt 5.3 wird dann, neben der Konstruktion der fts-Hierarchie, das Verfahren des initialen Clustering beschrieben. Hierbei handelt es sich um eine Auswahl der *besten* fts, um eine Cluster-Hierarchie zu bilden.

5.1 Grundstrukturen des fts-Clustering

Nachdem im vergangenen Kapitel die Vorverarbeitung für die Clusteranalyse beschrieben wurde, soll in diesem Abschnitt erläutert werden, wie aus den selektierten Termen in 4.2.5 eine Cluster-Hierarchie erstellt wird.

In dieser Arbeit wird ein Ansatz mit einer Überdeckung[1] der Dokumente gewählt, da davon auszugehen ist, dass viele Dokumente thematisch in mehrere Klassen gleichzeitig eingeordnet werden können. Auch wird in dieser Arbeit eine hierarchische Struktur präferiert, da die Dokumente sich schon aufgrund ihres inneren Aufbaus in eine Struktur vom allgemeinen zum speziellen Fall einordnen lassen. Wie in 3.1 beschrieben, besteht ein Newsgroupartikel, ein Dokument, aus einem oder mehreren Beiträgen. Dabei beziehen sich die einzelnen Beiträge immer auf Vorangegangene[2]. Oftmals wird in späteren Beiträgen das Thema des Newsgroupartikels spezialisiert, da die Problemstellung genauer gefasst wird. So werden vor Beantwortung einer Anfrage häufig erst noch Eingrenzungsnachfragen gestellt, da die ursprüngliche Fragestellung zu allgemein gehalten ist, um sie direkt zu beantworten. Dies ist vor allem in technischen Newsgroups der Fall. In Newsgroups, die eher als Diskussionsforen konzipiert sind, ist diese Form der Spezialisierung nicht so ausgeprägt. Dafür ist es häufiger der Fall, dass ein Thema erweitert wird oder

[1] Ein Dokument kann also in mehreren Klassen bzw. Clustern auftreten, welche direkte Kind-Cluster eines Eltern-Clusters sind.

[2] Außer natürlich der Eröffnungsbeitrag.

im Verlauf einer Diskussion mehrere Themengebiete angeschnitten werden. Dies spricht dann auch für einen überdeckenden hierarchischen Ansatz, also eine Quasihierarchie (siehe 2.1.2.3).

Als Ausgangspunkt zur Erstellung einer Quasihierarchie dient das Verfahren, welches in 2.3.4.2 beschrieben wurde. Die Essenz dieses Verfahrens ist die Auswahl von Clustern aus einer Menge von Clusterkandidaten, die aus den sogenannten *frequent term sets*, also von häufig auftretenden Termmengen, gebildet werden. Die Auswahl von Clustern aus der Kandidatenmenge allerdings wird hier von den Blättern (*bottom-up*) der fts-Hierarchie ausgeführt[3]. Es werden jeweils die *besten* Clusterkandidaten pro Ebene und Eltern-Cluster ausgewählt, um die Menge der Dokumente des jeweiligen Eltern-Clusters zu überdecken.

5.1.1 fts-Hierarchie

An dieser Stelle soll die Erstellung und Strukturierung der Menge der Clusterkandidaten erläutert werden. Dies soll anhand eines kleinen Beispieles geschehen. Vereinfachend wird angenommen, dass die zugrundeliegende Dokumentenmenge aus zehn Dokumenten mit nur drei relevanten Termen besteht. Die Abbildung 5.1 zeigt die entsprechende Zuordnung der Dokumente zu den Termen, die sie beinhalten.

{oracle}	{D1,- -,D3,D4,D5,- -,D7,D8, - -,- -}
{databas}	{D1,D2,- -,D4,D5 ,- -,D7,- -,D9,D10}
{db2}	{- -,D2,- -,- -,D5, D6,D7,- -,- -,D10}

Abbildung 5.1: Zuordnung von Dokumenten zu Termen.

In Abbildung 5.2 wird die Beziehungsstruktur der Potenzmenge der Terme graphisch gezeigt. Diese ergibt sich, wenn die Ober- bzw. Untermengenbeziehung als verbindende Relation angenommen wird. Die Termmengen bilden hierbei einen Verband. Aus der Potenzmenge der Termmenge sollen nun die Mengen ausgeschlossen werden, die nicht häufig sind. Hierfür wird eine minimale Dokumenthäufigkeit, also eine untere Schranke, der *MinSupport* für die Termmengen festgelegt.

Definition 5.1.1 (**MinSupport**)**.** Der *MinSupport* definiert den minimal notwendigen Anteil an Dokumenten aus der Dokumentkollektion, den eine Termmenge erreichen oder überschreiten muss, um häufig zu sein, um also ein *frequent term set* zu bilden. Der *MinSupport* liegt in dem Intervall $]0, 1]$.

Wenn eine Termmenge diese Schranke überschreitet, wird sie als Clusterkandidat für ihre Term-Untermengen gesetzt. Ansonsten wird sie und auch ihre Term-Obermengen, im weiteren Verlauf des Verfahrens nicht mehr betrachtet.

Definition 5.1.2 (**Definierende Termmenge**)**.** Ein frequent term set *definiert* einen Clusterkandidaten. Der Clusterkandidat enthält die Dokumente, welche die Termmenge des ihn definierenden fts enthalten.

[3] Das Clusterverfahren aus 2.3.4.2 geht *top-down* vor.

Wendet man auf das Beispiel einen MinSupport von 40% an, so sind alle einfachen Termmengen, also alle Termmengen, die nur einen Term beinhalten, Clusterkandidaten der leeren Menge (der Wurzel {}). Des Weiteren gibt es nur noch zwei weitere fts. Und zwar die Termpaare $\{databas, db2\}$ und $\{databas, oracle\}$.

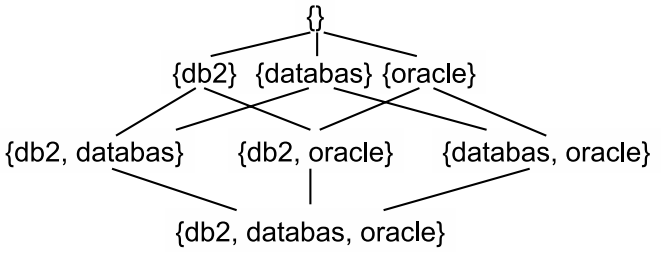


Abbildung 5.2: Verbandstruktur der Termmengen.

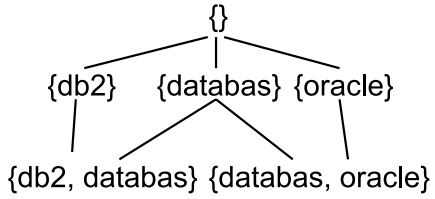


Abbildung 5.3: fts-Struktur der Termmengen.

Entfernt man nun alle Termmengen, welche nicht häufig sind, aus dem Verband, erhält man die in 5.3 gezeigte Struktur. Dieser Teilverband wird im Weiteren als fts-Struktur oder fts-Hierarchie bezeichnet. Er stellt die Grundlage für das nachfolgende Clusterverfahren dar. Jeder Knoten des Graphen ist ein Clusterkandidat für die mit ihm verbundenen Eltern-Cluster (Kante nach oben) und hat seine *eigenen* Clusterkandidaten in seinen Kind-Clustern (Kante nach unten).

5.1.2 Überlappungsminimale Überdeckung

In [BEX02] wird eine Auswahl der *besten* Cluster aus der Menge der Clusterkandidaten anhand der überlappungsminimalen Überdeckung getroffen. Dies soll gewährleisten, dass die Cluster sich thematisch maximal voneinander unterscheiden und die Newsgroupdomäne möglichst gut partitioniert wird. Allerdings ergeben sich zwei Probleme bei diesem Ansatz.

Erstens ist der Raum der möglichen Lösungen die Potenzmenge der jeweiligen Clusterkandidaten und diese Menge der Lösungsmöglichkeiten kann auch nicht so weit eingeschränkt werden, dass sich die Komplexität von $O(2^n)$ auf einen polynominellen Faktor verringern würde. Die Berechnung der überlappungsminimalen Überdeckung ist also sehr ineffizient. In [BEX02] wird daher ein greedy-Algorithmus zur Approximation der überlappungsminimalen Überdeckung vorgeschlagen, der eine lineare Laufzeit in der Menge der Clusterkandidaten hat. Diese einfache Approximation liegt allerdings bei Dokumenten einer Newsgroup, wie Experimente gezeigt haben, oft sehr weit vom Optimum entfernt.

Das zweite Problem, welches sich bei den Evaluationen mit den Dokumenten einer Newsgroup und dem Ansatz der überlappungsminimalen Überdeckung gezeigt hat, ist die *buschige* Struktur der Cluster-Hierarchie. Mit anderen Worten, es wird häufig eine große Anzahl an Clustern aus den Clusterkandidaten gewählt. 25-35 Cluster bei 100 Kandidaten sind keine Seltenheit. Eine solche Struktur ist von einem Menschen nur sehr schwer zu überschauen und gibt auch nur in den wenigsten Fällen die inhärente Struktur einer Dokumentenmenge wieder. Grund für diesen Effekt ist die in [BEX02] getroffene Prämisse, dass alle Dokumente abgedeckt werden müssen. Nun kann es in der Menge der Dokumente aber welche geben, die nur wenigen oder sogar nur einem fts zugeordnet sind und die dadurch das Hinzunehmen der entsprechenden Clusterkandidaten erzwingen, auch wenn diese Kandidaten eine sehr hohe Überlappung mit den bereits gewählten Clustern haben.

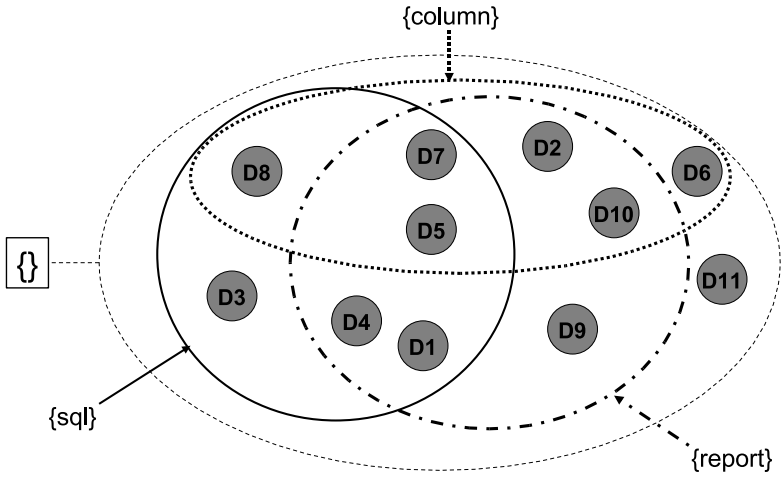


Abbildung 5.4: Diagramm der Dokument-Term Beziehung

In Abbildung 5.4 wird die Beziehung zwischen den in Abbildung 5.1 angegebenen Termen und Dokumenten in einem Kreisdiagramm gezeigt, wobei die umschließenden Kreise die Terme oder Clusterkandidaten sind und die kleinen die jeweiligen Dokumente, die den Term enthalten, von dem sie umschlossen werden. Allerdings ist in der Abbildung 5.4 nur die Ebene der Kind-Clusterkandidaten des Wurzel-Clusters (hier durch {} angedeutet), also alle 1-fts visualisiert, um das Diagramm übersichtlich zu halten. Im Diagramm ist noch ein weiteres, bisher nicht genanntes Dokument zu sehen, und zwar $D11$. Dieses Dokument enthält keinen der häufigen Terme und kann von daher auch nicht überdeckt werden und scheidet somit auch aus der weiteren Betrachtung aus. Dokumente, die von keinem Kandidaten abgedeckt werden können, sind für dieses Clusterverfahren *Ausreißer*.

Das Diagramm zeigt weiterhin, dass um alle Dokumente, die abdeckbar sind, abzudecken, alle drei Clusterkandidaten gewählt werden müssen. Dies lässt schon erahnen, was passiert, wenn eine große Dokumentmenge mit vielen Clusterkandidaten abgedeckt werden soll. Gerade bei sehr heterogenen Dokumenten[4] wie denen einer Newsgroup ist es notwendig, eine große Anzahl an Kandidaten auszuwählen, um die Dokumentenmenge vollständig abzudecken. Dies führt dann zu der oben erwähnten *buschigen* Struktur des

[4] Heterogen in Bezug auf die Wortwahl und Länge.

Clusterings.

5.2 Erweitertes Kostenmaß

In dieser Arbeit soll ein erweitertes Kostenmaß vorgestellt werden, mit dessen Hilfe man ein initiales Clustering einer Dokumentenmenge und auch die Inkrementierung dieses Clusterings vornehmen kann. Dieses Kostenmaß soll zum einen den Nachteil der *buschigen* Struktur des Clusterings beheben, die entsteht, wenn man nur das sehr einfache Kostenmaß der minimalen Überlappung (siehe 5.1.2) zwischen den einzelnen Clustern heranzieht. Zum anderen soll dieses erweiterte Kostenmaß auch eine Verbindung schaffen zwischen dem lokal orientierten Clustering eines beliebigen Clusters, welches nur die lokalen Begebenheiten (Überlappung zwischen den Kind-Clustern) betrachtet, und einem global orientierten Clustering, welches auch die Qualität des Clusterings der Kandidaten dieses Clusters mit einbezieht. Die Qualität eines Clusterings, gemeint ist hier die Überdeckung einer beliebigen Dokumentenmenge, wird hier anhand von Kosten gemessen. Geringe Kosten bedeuten eine hohe Qualität und umgekehrt. Es sei allerdings bemerkt, dass *Qualität* hier nicht allgemeingültig definiert ist, sondern eben in Abhängigkeit von dem weiter unten beschriebenen Kostenmaß. Die Schaffung eines globalen Kostenmaßes, welches die Qualität des Gesamtclusterings misst, hat zum einen den Vorteil, dass das Clusterverfahren so konstruiert werden kann, dass es das globale Optimum findet. Zum anderen ist ein Anwender häufig an speziellen Sachverhalten interessiert und wird sich, soweit es geht, in der Hierarchie zu den Blättern durcharbeiten wollen. Daher ist es notwendig, im Clustering auf oberer Ebene (also näher zur Wurzel hin) auch auf die Qualität der Nachfolger zu achten. Denn wenn die Qualität in den blattnahen Clustern gering ist, so wird der Anwender kaum Nutzen aus dem Clustering ziehen können.

Allgemein gesprochen soll das Kostenmaß zum einen die lokal entstehenden Kosten in einem beliebigen Cluster der Hierarchie berücksichtigen, zum anderen aber auch eine globale Sichtweise verwenden, welche die Kosten des gesamten Teilbaumes unterhalb des Clusters mit einbezieht. Um diesen Anforderungen gerecht zu werden, setzt sich das Kostenmaß grob aus zwei Komponenten zusammen: einem lokalen Kostenmaß und einer globalen Kostenmaßerweiterung. Das lokale Kostenmaß wiederum setzt sich aus zwei unterschiedlichen Faktoren zusammen, die in den beiden folgenden Abschnitten unabhängig voneinander beschrieben werden sollen, um dann in ein gemeinsames (lokales) Maß überführt zu werden. Danach wird dieses lokale Maß zu einem globalen Kostenmaß erweitert. Dieses globale Maß gibt dann die Qualität eines Teilbaumes in der Cluster-Hierarchie an. Daraus folgt, dass dann im Wurzel-Knoten des Clusterings (die leere Termmenge) die Qualität des gesamten Clusterings wiedergeben wird.

5.2.1 Überlappungskosten

Wie oben bereits beschrieben, ist die Grundidee des als Vorbild dienenden Clusteringverfahrens, eine überlappungsminimale Überdeckung der Dokumentenmenge zu erreichen. Intuitiv enthält diese Formulierung auch ein natürliches Kostenmaß. Die Überlappungen, die sich zwischen den gewählten Clustern ergeben, kann man als Kosten interpretieren, und zwar dergestalt, dass die Anzahl der Dokumente im Schnitt zweier Cluster die Kosten für die gleichzeitige Wahl dieser beiden Cluster im Clustering sind. Diese Rechnung kann

auf das gesamte Clustering ausgedehnt werden. Hierfür wird die Summe aller Dokumentenschnitte zwischen allen gewählten Clustern des gleichen Eltern-Clusters gebildet. So erhält man die Gesamtanzahl an Überlappungen bzw. die Gesamtkosten im Clustering. Je geringer diese Kosten sind, desto *besser* ist das Clustering. Es sei darauf hingewiesen, dass hier nur das flache - lokale - Clustering unter einem beliebigen Knoten in der fts-Hierarchie gemeint ist. Also wird nicht der Schnitt zwischen zwei Clustern in unterschiedlichen Ebenen der Hierarchie und/oder auch nur mit verschiedenen Eltern-Cluster berechnet, sondern isoliert die Kosten für das Clustering eines einzelnen Clusters.

Man kann die Berechnung der Überlappungskosten auch aus der Sicht der Dokumente beschreiben. Ein Dokument verursacht dann ab dem zweiten gewählten Cluster in dem er enthalten ist, pro Cluster einen Kostenpunkt. Um diese Kostenberechnung zu formalisieren, soll zuerst ein paar Definitionen eingeführt werden:

- C_i := Ein beliebiger Cluster i in der Clusterhierarchie.
- d := Ein Dokument aus der Menger aller Dokumente D_i in C_i.
- M_i := Menge aller (gewählten) Cluster(-kandidaten) in C_i (Menge seiner Kinder).
- N_i := Menge aller (nicht gewählten) Clusterkandidaten von C_i.
- Cd_i := Menge aller Dokumente d, für die gilt $\exists C \in M_i$ mit $d \in C$.
- $m_i(d)$:= Anzahl aller Cluster C, für die gilt $C \in M_i \wedge d \in C$.

Wichtig ist hier die Unterscheidung zwischen den Objekten in der Menge M_i und den Objekten in der Menge N_i. Die Menge M_i enthält alle ausgewählten Cluster. Alle Clusterkandidaten befinden sich in der Menge N_i. Diese beiden Mengen sind disjunkt. Jedes fts ist, wie oben bereits ausgeführt, ein Clusterkandidat für seine direkten Vorfahren in der fts-Hierarchie. Wird er von einem seiner Eltern-Cluster gewählt, wird er für diesen als ein *gewählte Clusterkandidat* oder einfach Cluster bezeichnet. Nachdem ein Cluster C_i also selbst geclustert wurde[5], sind die zur Überdeckung gewählten Cluster in M_i und die nicht gewählten (Clusterkandidaten) in N_i. Mit Hilfe dieser Definitionen lassen sich die Überlappungskosten eines Dokumentes in einer Überdeckung des Clusters C_i durch die folgende Formel beschreiben:

$$cost_{ov}(d) = m_i(d) - 1 \tag{5.1}$$

Aus dieser folgt direkt die Formel der Überlappungskosten eines Cluster.

$$OC(C_i) = \sum_{d \in Cd_i} cost_{ov}(d) \tag{5.2}$$

Umgangssprachlich ausgedrückt, sind die Überlappungskosten eines Clusters (OC - Overlap Cost) also gleich der Summe der Mehrfachverwendungen aller Dokumente in seinen Kind-Clustern. Die Prämisse, dass alle Dokumente abgedeckt werden müssen, wie

[5] Präziser wäre hier zu sagen, dass seine Dokumentmenge D_i geclustert wurde. Da der Cluster aber als ein Repräsentant seiner Dokumentmenge angesehen werden kann, wird im Weiteren von dem „clustern eines Clusters“ gesprochen.

sie von den Autoren in 2.3.4.2 vorausgesetzt wurde, soll in diesem Verfahren ausdrücklich nicht gelten. Ein Dokument muss also nicht in einem gewählten Cluster vorhanden sein, auch wenn es Clusterkandidaten für dieses Dokument gibt.

Konkret auf das Beispiel in Abbildung 5.4 bezogen, folgt aus der Formel 5.2, dass die Wahl eines einzelnen Clusters die Überlappungskosten von null nach sich zieht. Die Wahl der beiden Clusterkandidaten $\{sql\}$ und $\{column\}$ Clusters führt zu Überlappungskosten von drei und die Wahl aller Cluster hat Kosten von neun zur Folge. Wenn nur die Überlappungskosten in dieser Form benutzt würden, wäre immer die Wahl eines oder sogar keines Clusters optimal. Dies ist natürlich nicht der gewünschte Effekt. Aus diesem Grund wird ein zweiter lokaler Kostenfaktor eingeführt.

5.2.2 Kosten nicht genutzter Möglichkeiten

In dem zugrundegelegten Verfahren aus 2.3.4.2, wurde für das Clustering einer Dokumentmenge angenommen, dass alle Dokumente mindestens einem Cluster zugeordnet werden müssen[6]. Dieses Vorgehen führt, wie oben erläutert, bei Newsgroups allerdings zu einem recht *buschigen* Clustering - ein Cluster hat sehr viele Kinder oder gewählte Cluster. Auch sind eben die Überlappungskosten sehr hoch, unter der Prämisse, dass alle Dokumente abgdeckt werden sollen. Ursache für diesen Effekt ist unter anderem die Tatsache, dass ein kurzes Dokument häufig nur eine geringe Anzahl an fts hat, im schlimmsten Fall enthält das Dokument nur einen fts. Dann muss dieser Clusterkandidat als Cluster ausgewählt werden, um das Dokument auch abzudecken. Es ist allerdings fraglich, ob ein unter solchen Umständen gewählter Cluster wirklich etwas zur sinnvollen Überdeckung der Dokumentmenge beiträgt. Auch ist es bei thematischen Ausreißern häufig der Fall, dass sie nur einen oder sehr wenige Clusterkandidaten haben.

Eine mögliche Lösung dieses Problems ist, dass man Dokumente, die nicht über eine vorgegebene Mindestanzahl an Cluster-Kandidaten verfügt, aus dem Clusteringprozess herausnimmt, das Clustering mittels der Überlappungskosten durchführt und erst danach die entfernten Dokumente wieder einsortiert. Hierbei kann es natürlich sein, dass ein Dokument keinem gewähltem Cluster zugefügt werden kann. In diesem Fall wird er einem Dummy-Cluster für gemischte Themen, einem sogenannten *Misc*-Cluster, zugeordnet. Diesem Sondercluster werden auch die Dokumente zugeordnet, die in keinem fts enthalten sind. Schwierig bei diesem Vorgehen ist allerdings, dass eine Grenze für die Mindestanzahl an Cluster-Kandidaten festgelegt werden muss, unter Umständen also mehrere Grenzwerte ausprobiert werden müssen, bevor ein akzeptables Ergebnis erreicht wird. Es soll daher stattdessen ein weiterer Kostenfaktor eingeführt werden, welcher sich mit den Überlappungskosten kombinieren lässt.

Jedes Dokument in einem Eltern-Cluster hat eine bestimmte Anzahl von Cluster-Kandidaten. Die Idee ist nun, analog zu den Kosten der Überlappungen, die ein Dokument verursachen kann, Kosten für seine *Nichtzuordnung* oder, anders ausgedrückt, Kosten für seine nicht genutzten Möglichkeiten fest zu setzen. Und hier bietet sich als Kostenfaktor eben die Anzahl der vorhandenen Cluster-Kandidaten eines Dokumentes an.

- $n_i(d)$:= Anzahl aller Clusterkandidaten C, für die gilt $C \in N_i \wedge d \in C$

[6] Dokumente für die kein Cluster-Kandidat existiert, werden an dieser Stelle nicht weiter betrachtet.

Mit dieser Zusatzdefinition lassen sich Formeln für die Kosten ungenutzter Möglichkeiten herleiten analog zu denen für die Überlappungskosten.

$$cost_{up}(d) = n_i(d) \tag{5.3}$$

$$UC(C_i) = \sum_{d \in D_i/Cd_i} cost_{up}(d) \tag{5.4}$$

Aus der Formel 5.4 geht hervor, dass ein Dokument nur dann Kosten für ungenutzte Möglichkeiten (UC - Unused possibillities Cost) verursachen kann, wenn es keinem gewähltem Cluster zugeordnet ist ($d \in D_i/Cd_i$).

Dieser Kostenfaktor soll wieder an dem Beispiel in Abbildung 5.4 näher erläutert werden. Bei der Wahl eines einzelnen Clusters sind die Kosten ungenutzter Möglichkeiten gleich der Summe der Anzahl von Clusterkandidaten der nicht abgedeckten Dokumente, also bei der Wahl von $\{sql\}$ gleich 6, da $D2$ und $D10$ zwei Cluster-Kandidaten haben, die sie abdecken könnten und $D6$ und $D9$ jeweils einen. Die Wahl der beiden Clusterkandidaten $\{sql\}$ und $\{column\}$ Clusters führt zu Kosten ungenutzter Möglichkeiten von eins, da nur das Dokument $D9$ noch von einem Kandidaten überdeckt werden kann. Die Wahl aller Cluster hat natürlich die Kosten von null zur Folge, da alle Dokumente, die abgedeckte werden können, auch wirklich abgedeckt werden. Wenn nur die Kosten ungenutzter Möglichkeiten betrachtet werden, ist also immer die Wahl aller oder zumindest einer Teilmenge der Cluster-Kandidaten, welche alle Dokumente abdeckt, optimal. Die Kosten ungenutzter Möglichkeiten verhalten sich bezogen auf die Menge der gewählten Cluster reziprok, also umgekehrt zu den Kosten der Überlappungen.

5.2.3 Lokale Gesamtkosten

Aus den Kostenfaktoren in den beiden vorangegangenen Abschnitten lassen sich nun die Gesamtkosten und ein lokales Kostenmaß für ein Clustering herleiten.

$$LCost(C_i) = UC(C_i) + OC(C_i) = \sum_{d \in D_i/Cd_i} cost_{up}(d) + \sum_{d \in Cd_i} cost_{ov}(d) \tag{5.5}$$

Die Gesamtkosten des Clusterings $LCost(C_i)$ (*Local Cost*) eines Clusters C_i in der Cluster-Hierarchie entspricht also der Summe der Überlappungen der Dokumente die mindestens einem gewählten Cluster zugeordnet sind, addiert zu der Summe der Cluster-Kandidaten aller nicht überdeckten Dokumente. Aus diesen Kosten lässt sich nun auch ein lokales Clusterkriterium herleiten. Das Clusterkriterium für das Clustering eines Clusters ist die Minimierung der lokalen Gesamtkosten. Das Kostenmaximum dieser Formel wird erreicht, wenn kein Clusterkandidat zur Überdeckung ausgewählt wird. Wenn also ein Cluster, der über Clusterkandidaten verfügt, nicht geclustert wird, so entsprechen die Kosten genau der Summe über die Anzahl der Clusterkandidaten in den Dokumenten des betrachteten Clusters. Wenn der so berechneten Kostenwert um die Menge der abdeckbaren Dokumente verringert wird, so ist dies Ergebnis gleich den Kosten, die sich ergeben, wenn alle Clusterkandidaten ausgewählt sind. Sind also alle Cluster-Kandidaten

ausgewählt, so entsprechen die Überlappungskosten den Kosten der nicht genutzten Möglichkeiten aller Dokumente des betrachteten Clusters abzüglich der Anzahl aller (abdeckbaren) Dokumente des betrachteten Clusters. Die Kosten bei Nichtclustern sind also auf alle Fälle höher als die bei Durchführung eines Clusterings. Auch ist leicht einzusehen, dass schon die Wahl eines einzigen Clusterkandidaten die Gesamtkosten, wie in Formel 5.5 berechnet, senkt. Das Ziel ist es, eine Auswahl von Clusterkandidaten zu finden, welche die Gesamtkosten minimiert.

Zur Verdeutlichung dient nochmal das Beispiel aus Abbildung 5.4. Werden alle drei Kandidaten ausgewählt, so ergeben sich Gesamtkosten von neun durch die vorhandenen Überlappungen. Wird kein Kandidat gewählt, so betragen die Kosten 19 aufgrund der ungenutzten Möglichkeiten. Die kostenminimale Überdeckung der Dokumente wird erreicht bei Auswahl der Kandidaten $\{sql\}$ und $\{column\}$. Die Überlappungskosten liegen bei drei für die Dokumente $D5$, $D7$ und $D8$, welche in beiden Clustern vorkommen und die Kosten ungenutzter Möglichkeiten liegen bei eins für das Dokument $D9$. Jede andere Überdeckung mit zwei Kandidaten verursacht Kosten von fünf, da der Schnitt jeweils aus vier Dokumenten besteht und immer ein Dokument nicht abgedeckt werden kann.

Ein Aspekt der Formel 5.1 soll an dieser Stelle näher diskutiert werden. Ein Dokument, welches abgedeckt ist, verursacht erst in dem Moment Kosten, in dem ein weiterer Clusterkandidat ausgewählt wird, welcher dieses Dokument enthält. Wenn man das Kostenmaß *Überlappungen* isoliert betrachtet, ist diese Sichtweise intuitiv naheliegend und bedarf keiner weiteren Erklärung. Im Zusammenhang mit den Kosten ungenutzter Möglichkeiten sind die Überlappungskosten insgesamt geringer bewertet als der Kostenfaktor der ungenutzten Möglichkeiten. Besonders groß ist eben der Unterschied der Kosten an den möglichen Extrempunkten, also bei Auswahl aller Kandidaten oder Auswahl keines Kandidaten, wobei die Auswirkung auf die inhaltliche Interpretierbarkeit bei beiden Möglichkeiten dieselbe ist - es gibt keinen Erkenntnisgewinn. Die Begründung für diese Ungleichgewichtung ist folgende: Das in dieser Arbeit vorgeschlagene Verfahren orientiert sich an Kosten, welche durch Dokumente entstehen oder als durch sie verursacht angesehen werden. Wird ein Dokument erstmalig durch die Auswahl eines Clusterkandidaten abgedeckt, so ist dies ein gewünschter Vorgang und sollte nicht *bestraft* werden und sei es nur durch einen Kostenpunkt. Natürlich ist eine andere Gewichtung denkbar, z. B. durch das Fortlassen der Subtraktion von eins bei den Überlappungskosten und / oder durch einen zwischen null und eins skalierbaren Gewichtungsfaktor. Allerdings sollen diese Möglichkeiten in dieser Arbeit nicht weiter verfolgt werden, da Experimente mit der Formel 5.5 gute Ergebnisse gezeigt haben.

Die lokalen Gesamtkosten sind an sich nicht so aussagekräftig, da die gleichen Gesamtkosten in Clustern mit sehr unterschiedlicher Dokument- und Clusterkandidatenanzahl entstehen können. Da Dokumente in diesem Ansatz als die *Verursacher* der Kosten angesehen werden, sollen die Gesamtkosten eines Clusters daher auf die Anzahl seiner Dokumente normiert werden.

$$LCost_d(C_i) = LCost(C_i) / |D_i| \tag{5.6}$$

$LCost_d(C_i)$ gibt also die durchschnittlichen Kosten wieder, die ein Dokument in dem Cluster C_i verursacht. Das Kriterium für das Clustering ist also, die durchschnittlichen Kosten eines Dokumentes zu minimieren. Im Weiteren wird die Angabe der Kosten pro Dokument verwendet, da diese einen besseren intuitiven Hinweis auf die Güte der jewei-

ligen Überdeckung bietet als der absolute Wert. Diese Wahl hat keinen Einfluss auf die Aussage späterer algorithmischer Berechnungen.

5.2.4 Globale Gesamtkosten

Das in den vorhergehenden Abschnitten entwickelte lokale Kostenmaß kann zum flachen Clustering eingesetzt werden oder aber auch top-down für hierarchisches Clustering. Wie bereits oben ausgeführt soll der hier vorgestellte Ansatz zur Erzeugung einer Quasihierarchie aber auf einem Kostenmaß beruhen, welches die Kosten, respektive die Güte nicht nur lokal betrachtet, sondern die gesamte (Teil-)Hierarchie zur Berechnung verwendet. Das lokale Kriterium (5.2.3) wird nun weiter ausgebaut zu einem globalen Clusterkriterium, mit Hilfe dessen eine kostenminimale Hierarchie erstellt werden kann.

$$GCost(C_i) = \frac{LCost_d(C_i) * W(C_i) + \sum_{C_j \in M_i} GCost(C_j) * W(C_j)}{W(C_i) + \sum_{C_j \in M_i \wedge GCost(C_j) > 0} W(C_j)} \tag{5.7}$$

Die Berechnung dieses globalen Clusterkriteriums erfolgt über die rekursive Formel 5.7. $W(C)$ ist ein Gewichtungsfaktor für die lokalen Kosten $LCost_d(C_i)$ des Eltern-Clusters und für die globalen Kosten $GCost(C_j)$ der Kind-Cluster. Dieser Gewichtungsfaktor ist in Abhängigkeit von dem jeweiligen Cluster zu berechnen. Er soll angeben wie die Kind-Cluster im Verhältnis zu ihrem Eltern-Cluster bewertet werden sollen. Eine mögliche Gewichtung ist beispielsweise die Anzahl der Dokumente in den jeweiligen Clustern. Dies würde auf alle Fälle den Eltern-Cluster ein größeres Gewicht verleihen, da er mindestens gleich viele und im überwiegenden Teil der Fälle mehr Dokumente hat als seine Kind-Cluster. Wollte man die Kosten der Kind-Cluster stärker einfließen lassen, kann anstelle der reinen Anzahl der Dokumente auch der Logarithmus der Dokumentanzahl verwendet werden.

Die Funktion $GCost(C_i)$ gibt für jeden Cluster C_i die gewichteten Kosten des gesamten Teilbaumes unterhalb des Clusters C_i an. Je geringer nun dieser Kosten sind, desto besser ist das Clustering der betrachteten Teilhierarchie.

Da es sich um eine rekursive Formel handelt, muss noch der Startwert definiert werden. In diesem Fall ist das der Wert der globalen Kosten in den Blättern der Hierarchie. Da in den Blättern keine Cluster-Kandidaten vorhanden sind und folglich auch keine Kosten für ungenutzte Möglichkeiten oder Überlappungen entstehen können, sind die Kosten hier gleich 0. Zu beachten ist auch, dass der Gewichtungsfaktor eines Clusters in der Summe des Nenners nur dann addiert wird, wenn die globalen Kosten des Clusters größer null sind.

Zur näheren Erläuterung ist in Abbildung 5.5 ein Auszug einer Clusterhierarchie dargestellt. Als Gewichtungsfaktor im diesem Beispiel dient die Anzahl der Dokumente. Im Graphen sind die Cluster als Knoten symbolisiert. Über den Knoten steht der Clustername und in den Knoten stehen jeweils die lokalen und die globalen Kosten (LCost(C) / GCost(C)). Neben den Knoten C und G sind die Dokumentanzahlen der korrespondierenden Cluster vermerkt.

Die Cluster A und B sind Blätter und haben Kosten von 0. Daher tragen sie auch nichts zu den globalen Kosten des Clusters C bei. Anders ist die Situation beim Cluster

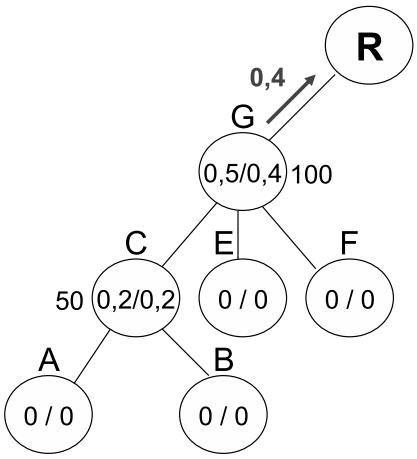


Abbildung 5.5: Berechnung der globalen Kosten.

G. Hier muss zumindest der Cluster C bei der Berechnung der globalen Kosten berücksichtigt werden. Die Berechnung sieht wie folgt aus:

$$GCost(G) = \frac{LCost_d(G)*W(G)+GCost(C)*W(C)}{W(G)+W(C)} = \frac{0{,}5*100+0{,}2*50}{100+50} = 0,4$$

Dieses Resultat wird dann wiederum weiter nach oben an den Cluster R gereicht.

Wenn man für einen Cluster auf diese Weise nicht nur die lokalen Kosten betrachtet, also die Überlappungskosten und die Kosten nicht genutzter Möglichkeiten, sondern auch die Kosten, die in den gewählten Clusterkandidaten entstanden sind und diese an den Eltern-Cluster weiterreicht, so erhält man im Wurzelknoten die gewichteten Gesamtkosten des Clusterings pro Dokument. Die Formel 5.7 kann direkt zur Erstellung einer Cluster-Hierarchie genutzt werden. Das Vorgehen hierfür soll im kommenden Abschnitt 5.3 erläutert werden.

5.3 Initiales Clustering

Das intitiale Clustering erstellt eine Quasihierarchie aus einer gegebenen Dokumentenmenge. Dies geschieht in zwei separaten Schritten, die in den folgenden beiden Abschnitten erläutert werden sollen. Als erstes wird die hierarchische Menge aller *frequent term sets* oder auch Clusterkandidaten erstellt. Dies ist, wie in Abschnitt 5.1.1 erläutert, ein Teilverband des vollständigen Verbandes der Potenzmenge der fts. Jeder Clusterkandidat und auch der Wurzelcluster, der per Definition bereits ein gewählter Cluster ist, hat also eine Menge an Clusterkandidaten, welche auch wiederum eine Menge an Clusterkandidaten haben (Mit Ausnahme der Blatt-Cluster, deren Termobermengen nicht genügend Dokumente enthalten, um häufig zu sein). Im zweiten Schritt wird für jeden Clusterkandidaten *bottom-up* die kostenminimale Auswahl aus seinen Clusterkandidaten ausgewählt. Hierzu wird das Clusterkriterium des vorangegangenen Abschnittes benutzt.

5.3.1 Hierarchische Erstellung der Clusterkandidaten

In Abschnitt 5.1.1 wurde bereits die Struktur der fts-Hierarchie beschrieben. Hier soll die Erzeugung dieser Struktur erläutert werden.

Die Erzeugung wird ebenenweise durchgeführt. Als erstes werden die 1-fts erstellt, dann die Ebene der 2-fts, usw. Wenn ein neues n-fts erzeugt wird, werden alle (n-1)-fts, welche Term-Untermengen des n-fts sind, mit diesem verbunden. Das neue fts wird der Menge der Clusterkandidaten der entsprechenden (n-1)-fts zugefügt. Die erste Ebene der 1-fts ist einfach aus der Vorverarbeitung 4.2.5 herzuleiten: Alle Terme, die den MinSupport überschreiten und die vom Anwender selektiert wurden, sind *frequent terms*. Sie sind gleichzeitig die Kandidaten des Wurzelclusters. Zur Identifikation der Cluster dienen die Terme als Namen oder Label. Bei n-fts, mit n>1 ist der Clustername die sortierte und durch Komma getrennte Konkatenation der Termnamen. Der Pseudocode in Abbildung 5.6 soll die Vorgehensweise verdeutlichen.

```
1   FrequentTermList          := All frequent terms
2   FTSQueue                  := Initial filled with 1-fts
3   FTSHash                   := Empty

4   Loop (FTSQueue Not Empty)
5       ProofFTS := Next from FTSQueue
6       For each FTerm ∈ FrequentTermList
7           If Not FTerm ⊆ ProofFTS Then
8               If Not FTerm ∪ ProofFTS ∈ FTSHash  Then
9                   If  Documents(FTerm) ∩ Documents(ProofFTS) > MinSupport Then
10                      NewFTS := FTerm ∪ ProofFTS
11                      Candidates(ProofFTS) ∪ NewFTS
12                      NewFTS -> FTSQueue
13                      NewFTS -> FTSHash
14                  End If
15              Else
16                  Candidates(ProofFTS) ∪ FTSHash[FTerm ∪ ProofFTS]
17          End If
18      End For
19  End Loop
```

Abbildung 5.6: Erzeugung der fts-Hierarchie

Es gibt eine Liste mit den 1-fts, also den häufigen Termen (*FrequentTermList*). Des Weiteren gibt es eine Warteschlange (*FTSQueue*), welche initial ebenfalls alle 1-fts enthält. Und es gibt einen Hash (*FTSHash*), der die erstellten k-fts (k $>$ 1) anhand ihres Namens speichert. In jedem Durchlauf der Schleife in Zeile 4 wird das erste fts aus der Warteschlange entfernt. Es werden nun sukzessive alle einfachen Erweiterungsmöglichkeiten für diese Termmenge überprüft. Es wird also nachgesehen, um welche 1-fts die Termmenge erweitert werden kann und ob diese neu entstandene Termmenge ebenfalls häufig ist. Im Detail werden alle 1-fts aus der Liste *FrequentTermList* durchlaufen und in Zeile 7 überprüft, ob sie Teilmengen des aktuell betrachteten fts sind. Es wird also geprüft, ob das 1-fts $\subseteq$ des k-fts auf der Termebene ist. Ein fts ist dann eine Teilmenge eines anderen, wenn seine Termmenge eine Untermenge von diesem ist. Beispielsweise ist

$\{sql\}$ natürlich eine Untermenge der Menge $\{databas, sql\}$. Ist dies der Fall, kann zum nächsten 1-fts aus der Liste übergegangen werden. Wenn nicht, wird in Zeile 8 geprüft, ob die sich neu ergebende Termmenge bereits erstellt wurde, also schon im *FTSHash* ist. Dies ist notwendig, da die Termmenge $\{column, databas, sql\}$ zum Beispiel, durch die Vereinigung der beiden fts $\{column, databas\}$ und $\{sql\}$ initial erstellt werden kann und zu einem späteren Zeitpunkt wird dann die Vereinigung von $\{column, sql\}$ und $\{databas\}$ geprüft. Da im zweiten Fall das fts-Objekt bereits existiert, darf es nicht neu angelegt werden, sondern das existierende muss nur noch als Kandidat für $\{column, sql\}$ vermerkt werden (Zeile 16). Wenn das fts noch nicht existiert, wird der Schnitt der Dokumentmengen der beiden fts gebildet. Ist die Dokumentenmenge dieses Schnittes größer als die vorgegebene Grenze des MinSupportes (Zeile 9), wird ein neues fts erstellt (Zeile 10). Dieses neue fts hat natürlich die Vereinigungsmenge der Termmengen beider fts und den Schnitt der beiden Dokumentmengen als Attribute. Das neue fts wird der Kandidatenmenge des zu prüfenden fts hinzugefügt (Zeile 11). Als letztes wird das neu entstandene fts nun noch ans Ende der Warteschlange angefügt und dem Hash hinzugefügt. Sind in der Warteschlange keine fts mehr enthalten, ist die fts-Hierarchie erstellt.

Der hier vorgestellte Algorithmus ist an die Methodik des Apriori-Algorithmus (siehe [AS94]) angelehnt. Es wird pro fts-Ebene ausprobiert, um welche 1-fts die jeweiligen k-fts erweitert werden können, d. .h welche neuen (k+1)-Termmengen den MinSupport überschreiten. Erfüllt eine getestete Termmenge nicht den MinSupport, werden auch die korrespondierenden Termobermengen nicht mehr geprüft. Dieses „generate-and-test“ Verfahren kann für die Erstellung der fts-Hierarchie gut eingesetzt werden, da viele Termmengen frühzeitig ausscheiden und die Teilstrukturen des Termmengenverbandes unterhalb dieser Termmengen nicht weiter überprüft werden müssen und so der Suchraum für fts stark eingeschränkt werden kann. Dieser Suchraum hat im worst-case eine Größe von 2^n in den häufigen Termen. Ein Punkt, warum nicht ein effizienterer Algorithmus wie beispielsweise *FPGrowth* (siehe [HPY00]) oder *Pattern Decomposition* (siehe [ZCC]) eingesetzt wird, ist die geringe Anzahl an initialen Objekten (hier also 1-fts)[7]. Wie in [HPY00] und [ZCC] ausgeführt wird, sind die dort vorgestellten Verfahren zum Finden häufiger Muster oder Objektmengen bei einer hohen Anzahl von initialen Objekten (ab einer Anzahl von 20.000 Objekten) wesentlich effizienter als ein Apriori-Algorithmus. Bei einer geringeren Menge ist der Vorteil allerdings marginal. Auch hat sich bei Evaluationen mit dem oben beschriebenen Algorithmus gezeigt, dass seine Laufzeit keinen nennenswerten Anteil an der Gesamtlaufzeit des Clusteringverfahrens hat.

5.3.2 Rekursive Auswahl der Cluster

Die Konstruktion der Quasihierarchie erfolgt nun aus der fts-Hierarchie unter Anwendung des in 5.2.3 beschriebenen Kostenmaßes. Das heißt, aus der Menge der jeweiligen Clusterkandidaten eines Clusters werden diejenigen ausgewählt, welche zusammen die minimalen globalen Kosten für den Eltern-Clusters ergeben. Da zur Berechnung der minimalen Kosten für einen Cluster auch die Kosten der Kind-Cluster benötigt werden, muss die Erstellung der Hierarchie in den Blättern begonnen werden. Mittels einer Tiefensuche kann diese Voraussetzung erfüllt werden. Die Abbildung 5.7 zeigt den entsprechenden

[7] Die Anzahl der häufigen Terme bei einem MinSupport von 5% liegt bei Newsgroupartikeln zwischen 70 und 200.

Rahmenalgorithmus.

```
doBottomUpClustering(CurrentCluster)
1   ClusterCandidates := Candidates(CurrentCluster)
2   For each ClusterCandidate ∈ ClusterCandidates
3       If Not IsClustered(ClusterCandidate)
4           doBottomUpClustering(ClusterCandidate)
5       End If
6   End For
7   doClustering(CurrentCluster)
```

Abbildung 5.7: Bottom-Up Clustering

Die Prozedur **doBottomUpClustering** wird für den Wurzelcluster aufgerufen. Der Wurzelcluster ist die per Definition bestehende leere Termmenge, welche die 1-fts als Clusterkandidaten hat. Dann wird die Prozedur **doBottomUpClustering** rekursiv für die Kandidaten aufgerufen. Hierauf folgt das eigentliche Clustering mittels der Prozedur **doClustering**. Diese Prozedur erstellt ein flache Überdeckung der Dokumente des Eltern-Clusters. Hierfür wird die Teilmenge aller Clusterkandidaten gesucht, welche die minimalen globale Kosten verursacht. Um dieses Optimum allerdings zu finden, muss in der Potenzmenge der Clusterkandidaten gesucht werden. Diese hat die Kardinalität 2^n und führt bei vollständiger Suche zu exponentiellem Zeitaufwand.

Im Folgenden soll ein *branch-and-bound* Algorithmus vorgestellt werden, welcher unter günstigen Umständen die Laufzeit der Suche nach dem Optimum stark reduziert. Im worst-case bleibt die Laufzeit jedoch weiterhin exponentiell und bei hohen N (also einer hohen Anzahl von Clusterkandidaten) ist er nicht brauchbar. Daher werden in Anschluss zwei Modifikationsmöglichkeiten dieses Algorithmus' vorgestellt, welche die Laufzeit weiter verkürzen sollen. Die erste Modifikation ist die Einschränkung der Rundenanzahl des *branch-and-bound* Algorithmus' in Verbindung mit einer Vorsortierung. Bei diesem Verfahren ist es allerdings wahrscheinlich, dass die Suche zu früh abgebrochen wird und man nur eine Annäherung zum globalen Optimum findet. Die zweite Modifikationsmöglichkeit enthält eine *greedy* Komponente, welche den Suchraum einschränkt, so dass die Laufzeit effizienter ist als beim *branch-and-bound* Algorithmus. Doch wird auch bei diesem Verfahren selten das globale Optimum, sondern häufig lediglich eine Approximation gefunden.

5.3.2.1 Branch-and-bound

Die Abbildung 5.8 zeigt den *branch-and-bound* Algorithmus, welcher das globale Optimum findet, also die Teilmenge der Clusterkandidaten selektiert, welche die globalen Kosten des Eltern-Clusters minimiert.

Die in Abbildung 5.8 gezeigte Prozedur wird von der Prozedur **doClustering** aus der Abbildung 5.7 aufgerufen. Diese übernimmt vor dem Aufruf der eigentlichen Clusterprozedur **doBaBClustering** noch andere Aufgaben. In **doClustering** wird, falls notwendig, eine Vorsortierung der Clusterkandidaten vorgenommen (siehe Abschnitt 5.3.2.2). Außerdem wird in dieser Prozedur geregelt, welcher Algorithmus für das flache Clustering benutzt wird: der in diesem Abschnitt beschriebene *branch-and-bound* Algorithmus,

```
CandidateList        := All clustercandidates for the actual parent cluster
BestChosenClusters   := Empty

doBaBClustering(iNext,ChosenClusters)
1   For iNext to Size(CandidateList)
2      If GCost(ChosenClusters ∪ CandidateList(iNext)) < GCost(ChosenClusters) AND
3       (LCost(ChosenClusters ∪ CandidateList(iNext)) < LCost(ChosenClusters) OR
4        GCost(ChosenClusters ∪ CandidateList(iNext)) < GCost(BestChosenClusters))
    Then
5          doBaBClustering(iNext + 1, ChosenClusters ∪ CandidateList(iNext))
6      End If
7   End For
8   If GCost(ChosenClusters) < GCost(BestChosenClusters)  Then
9      BestChosenClusters := ChosenClusters
10  End If
```

Abbildung 5.8: branch-and-bound Algorithmus

der rundenzahlbasierte Algorithmus aus Abschnitt 5.3.2.2 oder der *greedy* Algorithmus aus Abschnitt 5.3.2.3.

In der rekursiven Prozedur in Abbildung 5.8 wird überprüft, ob die Hinzunahme eines weiteren Clusterkandidaten zu einer Senkung der globalen Kosten führt (Zeile 2) und ob die lokalen Kosten sinken (Zeile 3) oder die globalen Kosten des neu hinzugekommenen Clusterkandidaten geringer sind, als die bisher niedrigsten, die für den Eltern-Cluster erreicht werden konnten. Wenn dies der Fall ist, wird die neu entstehende Clustermenge eine Rekursionsstufe weiter gereicht (Zeile 5). Aufgerufen wird die Prozedur initial mit null für *iNext* und der leeren Menge für *ChosenClusters* aufgerufen. Bei den Clusterkandidaten (*CandidateList*) ist eine (beliebige) Reihenfolge vorgegeben und mit dem Zähler *iNext* kann ein spezifischer Kandidat angesprochen werden. Die lokalen (**LCost**) und globalen Kosten (**GCost**) sind für die leere Menge als $+\infty$ definiert. Für eine Clustermenge ungleich der leeren Menge sind die lokalen und die globalen Kosten definiert wie in Abschnitt 5.2.3 bzw. 5.2.4 beschrieben. Als letztes wird in der Prozedur überprüft, ob die in dem Rekursionsschritt gefundenen globalen Kosten geringer sind als die der bisher besten Clustermenge (Zeile 8). Wenn ja, wird die hier gefundene Clustermenge als kostenminimale (beste) Menge gesetzt.

Das Ziel der Prozedur 5.8 ist es, die Teilmenge der Clusterkandidaten zu finden, die das Minimum der globale Kosten ergibt, also das Finden des globalen Optimums[8]. Um den Raum der möglichen Lösungen von $O(2^n)$ aber nicht komplett ab zu suchen, wird der Suchraum systematisch abgesucht und Teile, welche nicht das Optimum enthalten können, werden ausgeschlossen. Der hier eingesetzte Algorithmus implementiert ein *branch-and-bound* Verfahren.

Die Abbildung 5.9 zeigt einen solchen Entscheidungsbaum für fünf Clusterkandidaten exemplarisch. Die Knoten des Entscheidungsbaumes sind jeweils Clusterkandidaten und sind in der Abbildung mit den Buchstaben A, B, C, D und E gekennzeichnet. Die Zahlen in den Knoten dienen einerseits der Identifikation, geben aber auch die Durchlaufreihenfolge (Tiefensuche) an. Eine Lösung aus dieser Sicht ist ein Pfad von der Wurzel zu einem

[8] *Global* ist hier natürlich nicht auf die globalen Kosten aus Abschnitt 5.2.4 bezogen.

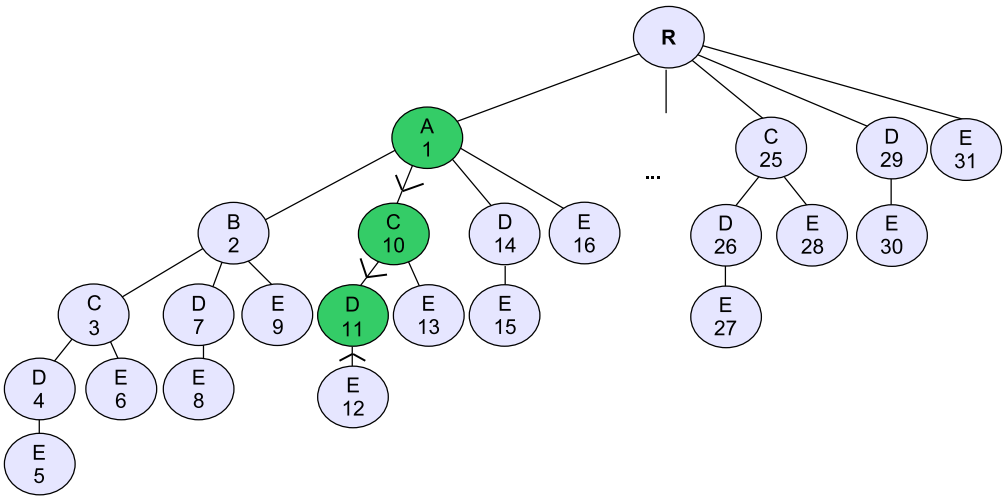


Abbildung 5.9: *branch-and-bound* Entscheidungsbaum

beliebigen Knoten. Die Menge der Knoten auf dem gewählten Pfad entsprechen der gewählten Teilmenge der Clusterkandidaten. Das globale Optimum wird in Abbildung 5.9 durch die grünen Knoten angedeutet. Der entsprechende Pfad läuft über die Knoten 1, 10 und 11. Das „$<$“ zwischen 1 und 10 sowie zwischen 10 und 11 bedeutet, dass die Kosten (globale wie lokale) durch die Hinzunahme von 10 und 11 jeweils gesunken sind. Das „$>$“ zwischen Knoten 11 und 12 zeigt an, dass die Kosten dort gestiegen sind. Die Teilmenge für die minimalen global Kosten besteht also aus den Clusterkandidaten A, C und D. Der Kandidat E kann nichts mehr zur Verbesserung beitragen.

Das Ziel der Bedingungen in den Zeilen 2 - 4 (siehe Abbildung 5.8) ist die Beschränkung des Entscheidungsbaumes, d. h. die Suche dort abzubrechen, wo das globale Optimum nicht mehr zu erwarten ist. Angenommen, in dem Beispiel stiegen die Kosten zwischen den Knoten 2 und 3, so würde der Algorithmus den Teilbaum mit den Knoten 4, 5 und 6 nicht mehr überprüfen, sondern direkt mit dem Knoten 7 weiter machen. Allgemein gesprochen, wird bei jedem erneuten Versuch, einen weiteren Kandidaten zur Menge der bisher gewählten Kandidaten (*ChosenClusters*) hinzuzufügen, überprüft, ob dieser das Ergebnis in Bezug auf alle Kandidatenobermengen der bisher betrachteten Kandidatenmenge verbessern kann oder ob er für diese Kandidatenmenge ausgeschlossen werden kann, womit also der vollständige Teilbaum unter der Kandidatenmenge *ChosenClusters* $\cup$ *CandidateList(iNext)* nicht überprüft werden muss.

Entscheidend ist natürlich, dass nicht der Ast abgeschnitten wird, in dem sich das globale Optimum befindet.

Der Beweis, dass die obige Prozedur dies gewährleistet wird über Widerspruch geführt. Vereinfachend für die Schreibweise wird angenommen, dass der Knoten C_k dem Clusterkandidaten k in der betrachteten Reihe von Clusterkandidaten und $\widehat{C_k}$ der Menge der Cluster $\{C_1, C_2, ..., C_k\}$ entspricht[9].

[9] Die Indices $1, ..., n$ bezeichnen einen beliebigen Pfad im Entscheidungsbaum, wobei die entsprechenden Cluster entsprechend ihrer Reihenfolge auf dem Pfad nummeriert sind.

Beweis:

In der Prozedur aus Abbildung 5.8 wird davon ausgegangen, dass das Hinzufügen eines Clusterkandidaten immer zu einer Verbesserung der globalen Kosten führen muss. Ist dies nicht der Fall, so kann die bisher gewählte Menge an Clusterkandidaten vereinigt mit dem betreffenden Kandidaten nicht Untermenge der Clustermenge des globalen Optimums sein. Im Folgenden soll durch die Annahme, dass das Optimum gerade durch einen solchen Fall gebildet wird, ein Widerspruch erzeugt und somit die Richtigkeit der Prozedur bewiesen werden.

Annahme:

Der Pfad $1, 2, 3, ..., n$ im Entscheidungsbaum beschreibe das globale Optimum für den Cluster C.

$GCost(\widehat{C_n} \prec C) \leq GCost(M \prec C) \forall M \in P(N)$

$\widehat{C_n} \prec C$ steht für den Cluster C mit der gewählten Clustermenge $\widehat{C_n}$. $P(N)$ bezeichnet die Potenzmenge der Clusterkandidaten von C und $GCost(\widehat{C_n} \prec C)$, somit die minimalen globalen Kosten (globales Optimum) für das Clustering von C.

Die Annahme ist, dass auf dem Pfad die globalen Kosten an einen Knoten k $<$ n, gleich bleiben oder steigen.

$GCost(\widehat{C_{k-1}} \prec C) \leq GCost(\widehat{C_k} \prec C) \wedge$
$GCost(\widehat{C_{k-1}} \prec C) \geq GCost(\widehat{C_n} \prec C)$
mit $k < n$

Für das Steigen oder Gleichbleiben der globalen Kosten kann es generell drei Gründe geben:

1. Die lokalen Kosten $LCost(C)$ steigen und die globalen Kosten des Clusterkandidaten C_k sind kleiner gleich den Kosten $GCost(\widehat{C_{k-1}} \prec C)$.

2. Die globalen Kosten des neu hinzu genommenen Clusterkandidaten C_k, liegen über dem bisherigen gewogenen Schnitt[10] und die lokale Verbesserung kann das nicht ausgleichen.

3. Die lokalen Kosten und die globalen Kosten liegen über dem bisherigen gewogenen Schnitt $GCost(\widehat{C_{k-1}} \prec C)$.

zu Punkt 1)

[10] Also den globalen Kosten des Clusters C mit der bisherigen Menge der ausgewählten Clusterkandidaten.

Sind die lokalen Kosten in Knoten k gestiegen, so sind die Kosten, die der Cluster C_k durch (Dokumenten)-Schnitt mit den bereits gewählten Clustern $C_1, ..., C_{k-1}$ verursacht, größer als die Kosten, die durch das Abdecken von Dokumenten eingespart werden können, die bisher nicht abgedeckt waren. Vereinfachend kann von einer Kostendifferenzrechnung pro Cluster gesprochen werden. Den Kosten durch Schnitt stehen *Einnahmen* durch Abdecken von Dokumenten[11] gegenüber, die sonst von keinem anderen gewählten Clusterkandidaten abgedeckt werden. Diese Kostendifferenzrechnung pro Cluster kann sich natürlich ändern, wenn weitere Cluster hinzugefügt werden. Allerdings werden nur mehr Kosten aus Sicht dieses gewählten Clusterkandidaten C_k angesammelt, da Dokumente, die bisher nur durch diesen Clusterkandidaten abgedeckt wurden, nun auch von den neuen Clusterkandidaten abgedeckt werden können und dadurch die Schnittkosten für den betrachteten Clusterkandidaten C_k noch weiter steigen. Aus diesen Betrachtungen folgt, dass das Entfernen des Clusterkandidaten C_k aus der Menge der gewählten Cluster $\widehat{C_n}$ zu einer Senkung der lokalen Kosten führen muss und dadurch gilt:

$$GCost(\widehat{C_n}/C_k \prec C) < GCost(\widehat{C_n} \prec C)$$

Das führt zum Widerspruch mit der Annahme, da $GCost(\widehat{C_n} \prec C)$ nicht das globale Optimum sein kann. Gleiches gilt auch, wenn die lokalen Kosten bei Hinzunahme des Clusters C_k stagnierten.

zu Punkt 2)

Die globalen Kosten eines Clusters C ergeben sich als gewogenes Mittel aus den globalen Kosten der Clusterkandidaten und den ebenfalls gewogenen lokalen Kosten des Clusters C (siehe Formel 5.7). Die globalen Kosten $GCost(C_k)$ des Clusters C_k liegen über den Kosten $GCost(\widehat{C_{k-1}} \prec C)$, andernfalls könnte nicht $GCost(\widehat{C_{k-1}} \prec C) < GCost(\widehat{C_k} \prec C)$ unter der Annahme sinkender oder gleich bleibender lokaler Kosten gelten. Damit gilt also folgendes:

$$GCost(C_k) > GCost(\widehat{C_{k-1}} \prec C) \wedge GCost(C_k) > GCost(\widehat{C_n} \prec C)$$

Da sich die Kostendifferenz für einen Cluster durch die Hinzunahme weiterer Cluster nicht verbessern kann (siehe Punkt 1),

$$GCost(\widehat{C_n}/C_k \prec C) < GCost(\widehat{C_n} \prec C)$$

$\Rightarrow$ Widerspruch zur Annahme.

zu Punkt 3)

Hier folgt der Widerspruch aus den Punkten 1) und 2).

Aus den Punkten 1 - 3 folgt, dass es kein globales Optimum auf einem Pfad geben kann, der einen Anstieg der globalen Kosten in einem Knoten hat! Also wird mit dem Algorithmus das globale Optimum gefunden. **qed**

[11] Anders ausgedrückt, werden Kosten ungenutzter Möglichkeiten gesenkt.

Dieser Beweis zeigt, dass die Bedingung in Zeile 2 nur Kandidatenmengen ausschließt, welche nicht das globale Optimum enthalten können. Allerdings wurde in diesem Beweis nicht auf die Bedingungen in den Zeilen 3 und 4 eingegangen. Jedoch greifen diese beiden Bedingung einen Sonderaspekt des Beweispunktes 1 auf. Und zwar wird überprüft, ob die globalen Kosten des betrachteten Kandidaten C_k größer sind als die niedrigsten bisher gefunden globalen Kosten und ob die lokalen Kosten gestiegen sind. Wenn dies nämlich der Fall ist, so können die globalen Kosten dieser Kandidatenmenge *ChosenClusters* $\cup$ *CandidateList(iNext)* nicht unter die Kosten der bisher besten Menge sinken, da der Faktor der zu der Verbesserung der globalen Kosten geführt hat (die globalen Kosten des Kandidaten C_k) größer ist als das bisherige Optimum.

Der Algorithmus arbeitet also korrekt und findet das globale Optimum, aber er hat im worst-case immer noch eine Laufzeit von $O(2^n)$ in den Clusterkandidaten. In den nächsten beiden Abschnitten sollen nun Verfahren erläutert werden, die bessere Laufzeiteigenschaften haben, jedoch oft nur zu Näherungslösungen führen.

5.3.2.2 Modifikation: Rundenanzahl und Vorsortierung

Eine Komponente dieser Modifikation ist die Vorsortierung der Clusterkandidaten mittels des Überlappungskoeffizienten. Es gelten die folgenden Definitionen:

- $N :=$ Menge aller Clusterkandidaten von C
- $n(d) := |\{CC \in N | d \in CC\}|$
- $dc(C) :=$ Anzahl der Dokumente in C

So ergibt sich der Überlappungskoeffizient wie folgt:

$$Overlap(C_i) = \sum_{d \in C_i} n(d)/dc(C_i) \tag{5.8}$$

Die Clusterkandidaten mit den niedrigsten Überlappungen im Verhältnis zu ihrer Dokumentanzahl sollen zuerst betrachtet werden. Wird die Menge der Clusterkandidaten nach also dem Überlappungskoeffizienten $Overlap(C_i)$ vorsortiert, so werden recht schnell gute Ergebnisse mittels des *branch-and-bound* Algorithmus gefunden.

Es wird nun ein Zähler in die Prozedur 5.8 eingebaut, welcher bei jedem rekursiven Aufruf der Prozedur inkrementiert wird. Überschreitet dieser Zähler eine vom Anwender vorgegebene Rundenanzahl, so wird der Algorithmus abgebrochen und das bisher beste Ergebnis wird genommen. Auf diesem Wege kann der Anwender die Laufzeit des Clusterverfahrens wirkungsvoll einschränken.

5.3.2.3 Modifikation: n-Greedy

Bei dieser Modifikation des *branch-and-bound* Algorithmus werden in jeder Rekursionsstufe nur die n besten Clusterkandidaten nacheinander ausgewählt, um die bisher gewählten Clusterkandidaten zu ergänzen. Die Abbildung 5.10 zeigt den groben Ablauf des Algorithmus.

Gestartet wird die Prozedur **doGreedyClustering** mit dem Wert null für $iNext$ (der Index der global bekannten Liste von Clusterkandidaten) und einer leeren Liste

```
BestChosenClusters   := Empty

doGreedyClustering(iNext, ChosenClusterList)
1    If GCost(ChosenClusterList) < GCost(BestChosenClusters) Then
2       BestChosenClusters := ChosenClusterList
3    End If
4    ProofList := GetNBestCandidates(iNext, ChosenClusterList)
5    For each Candidate ∈ ProofList
6         Index  :=  GetIndexFrom(Candidate)
7         doGreedyClustering(Index + 1, ChosenClusterList ∪ Candidate)
8    End For
```

Abbildung 5.10: n-greedy Modifikation

für (*ChosenClusterList*). Auf jeder Rekursionsstufe werden die n besten Kandidaten selektiert (Zeile 4). n ist dabei eine vom Anwender festgelegte natürliche Zahl und die Prozedur **GetNBestCandidates** holt die n Kandidaten aus der global bekannten Liste der Clusterkandidaten, für die gilt: Jeder der Kandidaten hat einen Index größer $iNext$ (in der Kandidatenliste) und seine Hinzunahme zu den bisher gewählten Clustern *ChosenClusterList* führt zu einer Senkung der globalen Kosten. Sollten nicht mehr genug Kandidaten vorhanden sein, welche einen Index größer $iNext$ haben, so gibt die Prozedur **GetNBestCandidates** nur eine entsprechend kleinere Liste mit Kandidaten zurück (u. U. auch eine leere Liste). Für die Liste mit selektierten Clusterkandidaten, wird die Prozedur **doGreedyClustering** dann rekursiv wieder aufgerufen. Am Beginn der Prozedur wird jeweils überprüft, ob das Ergebnis der vorhergehenden Rekursion besser war, als das bisher beste. Wie oben bereits definiert, gibt die Prozedur **GCost** für die leere Menge den Wert ∞ zurück. Wenn der Vorgang abgeschlossen ist, enthält die Liste *BestChosenClusters* die Menge der Clusterkandidaten, welche die geringsten globalen Kosten während der Suche erzielt haben.

Diese oben beschriebene Vorgehensweise führt dazu, dass von einem Entscheidungsbaum (oder Teilbaum) immer maximal die vom Anwender angegebene Anzahl n von Teilbäumen durchsucht wird. Das Kriterium zur Auswahl der entsprechenden Teilbäume sind die globalen Kosten, da dies auch das Clusterkriterium ist. Der Anwender entscheidet durch die Angabe der maximal zu betrachtenden Kandidaten pro Rekursion n einerseits über die Anzahl der Durchläufe für das Clustering einer Dokumentmenge, andererseits aber auch über die Güte, da die Wahrscheinlichkeit das globale Optimum zu finden, mit steigender Anzahl der Versuche steigt.

5.3.3 Besonderheiten des Verfahrens

Ungeachtet der speziellen Ausprägung des Algorithmus, wird immer eine strukturell gleiche Cluster-Hierarchie aus der fts-Hierarchie erstellt. Ein Cluster enthält oder besitzt Kinder (bis auf Blätter), welche auch wiederum Cluster sind. Diese Kind-Cluster können sich überschneiden. Zwei Cluster derselben Ebene, also Cluster, die beide aus jeweils einem k-fts hervorgegangen sind, können auch einen Kind-Cluster gemeinsam haben. Beispielsweise können die Cluster $\{column, sql\}$ und $\{column, database\}$ bei-

de u. a. den Nachfolgecluster $\{column, database, sql\}$ haben. Dies ist allerdings nicht zwingend der Fall. In der fts-Hierarchie ist die gemeinsame *Elternschaft* gegeben, sobald $\{column, database, sql\}$ eine häufige Termmenge ist. In der Clusterhierarchie kann dieser Fall auftreten, aber es ist auch möglich, dass der Clusterkandidat $\{column, database, sql\}$ nur für einen oder auch für keinen der beiden Cluster ausgewählt wird.

5.3.3.1 Misc-Cluster

In den bisherigen Betrachtungen wurde nicht näher auf Dokumente eingegangen, welche keinem Cluster zugeordnet werden können. Diese Dokumente erscheinen natürlich in keinem Kind-Cluster. Sie sollen allerdings auch für den Anwender sichtbar sein. Um dies zu gewährleisten, wird jedem Cluster, der selbst auch geclustert werden konnte, ein Dummy-Cluster zugefügt: der so genannte Misc-Cluster[12]. Diesem Cluster werden als erstes alle Dokumente zugeordnet, welche durch keinen Clusterkandidaten abgedeckt werden können. Und dem Misc-Cluster werden nach erfolgtem Clustering zusätzlich auch die Dokumente zugeordnet, die zwar mindestens einem Kandidaten zugehören, von dessen Kandidaten aber keiner während des Clusterings ausgewählt wurde. Für das Clustering an sich und auch für das spätere Inkrementieren hat der Misc-Ordner keine Bewandtnis. Er dient lediglich dazu, dem Anwender eine Zugriffsmöglichkeit auf die nicht zugeordneten Dokumenten zu gewähren.

5.3.3.2 Parallelhierarchien

Der Algorithmus aus Abschnitt 5.3.2 erstellt das Clustering *bottom-up*, da für das clustern einer Dokumentenmenge die globalen Kosten der Kind-Cluster benötigt werden. Aus der Menge der Clusterkandidaten werden die Cluster ausgewählt, welche die Dokumentmenge eines Eltern-Cluster am besten (also mit den geringsten globalen Kosten) abdeckt. So entstehen aus der Sicht des Eltern-Clusters zwei disjunkte Mengen: die Menge der (verbliebenen) Clusterkandidaten und die Menge der (gewählten) Cluster. Jeder Cluster beider Mengen besitzt selbst wieder zwei solcher Mengen, insofern er über Kandidaten verfügt. Durch das Clustering wird so aus der fts-Hierarchie eine Teilstruktur ausgewählt, die Cluster-Hierarchie. Die Cluster-Hierarchie ist die Struktur, welche sich ergibt, wenn man ab dem Wurzel-Cluster jeweils den Mengen der gewählten Cluster folgt. Neben dieser Cluster-Hierarchie bleibt die fts-Hierarchie bestehen, die aus sämtlichen Clustern, den Kandidaten und den gewählten Clustern besteht.

Die Abbildung 5.11 soll diese Doppelstruktur verdeutlichen. Die grün ausgefüllten Kreise symbolisieren Cluster der Cluster-Hierarchie und die fett gedruckten Kanten die *Eltern-Kind-Relation* von einem Cluster(-kandidaten) zu einem von ihm gewählten Cluster. Die Beziehung zwischen Clustern und ihren nicht gewählten Clusterkandidaten (*Eltern-Kandidat-Relationen*) werden durch die dünnen Linien wiedergegeben. Weiße Knoten (nicht ausgefüllte Kreise) sind fts, die von keinem ihrer Eltern-Cluster gewählt wurden. Graue Knoten sind gewählte Cluster eines nicht zur Clusterhierarchie gehörenden Eltern-Cluster. Knoten, die zu Hälfte grün und zur anderen Hälfte grau hinterlegt sind, gehören zur Cluster-Hierarchie und wurden zusätzlich von einem nicht zur Cluster-Hierarchie gehörenden Cluster gewählt. Dies ist bei den Knoten **H** und **I** der Fall. Der

[12] *Misc* steht für das englische *miscellaneous* (Diverses).

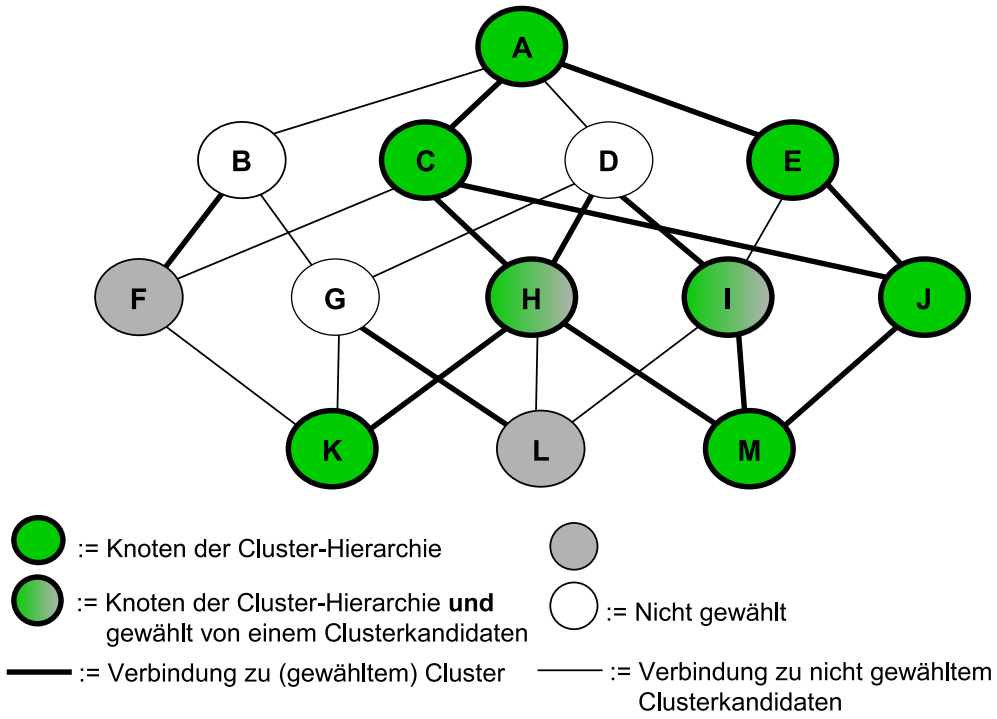


Abbildung 5.11: fts- und Cluster-Hierarchie

Knoten **H** ist in der Menge der gewählten Cluster für den Clusterhierarchie-Knoten **C** und er ist gewählt von dem *nur* zur fts-Hierarchie gehörenden Cluster **D**. Zur fts-Hierarchie gehören alle Knoten, zur Cluster-Hierarchie aber lediglich die Knoten **A**, **C**, **E**, **H**, **I**, **J**, **K** und **M**. Heraus zu heben sind zwei Sachverhalte:

1. Der Cluster **B** ist zwar nicht in der Cluster-Hierarchie, aber er ist selbst geclustert. Er hat einen gewählten Cluster (**F**) und einen Kandidaten (**G**).

2. Der Cluster **M** hat drei Eltern-Cluster - **H** ,**I** und **J**.

Diese Sichtweise zweier paralleler Strukturen ist für die im nachfolgenden Kapitel beschriebene Inkrementierung der Cluster-Hierarchie wichtig, da beispielsweise gewählte Cluster durch Clusterkandidaten ersetzt werden können, um so die globalen Kosten zu senken.

5.3.3.3 Globales Optimum

An dieser Stelle soll nochmals darauf hingewiesen werden, dass nach durchgeführtem Clustering die globalen Kosten in der Wurzel die Güte des gesamten Clusterings wiedergeben. Wenn also das Clustering mittels des *branch-and-bound* Algorithmus aus Abschnitt 5.3.2.1 durchgeführt wird, so steht der kleinst mögliche Wert der globalen Kosten in der Wurzel. Es handelt sich bei der so erstellten Cluster-Hierarchie um die mit der höchst möglichen Güte (nach dem in dieser Arbeit vorgestellten Maß und unter den vorgegebenen Parametern: MinSupport, vorselektierte Terme und Gewichtungsfaktor der

Kosten). Diese Schlussfolgerung folgt direkt aus den Beobachtungen, dass für jeden Cluster die Überdeckung (die Clusterkandidatenmenge) mit den minimalen globalen Kosten gefunden wurden und dass für diese globalen Kosten sowohl die lokalen Kosten für die Überdeckung der Dokumente des Clusters (Überlappungen und ungenutzte Möglichkeiten), als auch die (globalen) Kosten der Kinder (gewichtet) herangezogen wurden (siehe Formel 5.7).

6 Inkrementelle Erweiterung

Im vorherigen Kapitel wurde beschrieben, wie unter Zuhilfenahme des dort definierten Kostenmaßes (5.2) ein initiales Clustering durchgeführt wird. In diesem Kapitel wird nun der zentrale Vorgang dieser Arbeit beschrieben: Die Inkrementierung einer bestehenden Clusterhierarchie. Der folgende Abschnitt beschreibt zunächst, in welchen unterschiedlichen Zusammenhängen das Zufügen eines neuen Beitrages zu einem Thread und somit zu einem Dokument stehen kann. Des Weiteren soll aufgezeigt werden, welche Folgen diese Vorgänge für die fts haben. In einem weiteren Abschnitt wird erläutert, welche Strukturänderungen an der fts-Hierarchie und der Cluster-Hierarchie durchgeführt werden müssen bzw. können, um dem Zufügen neuer Dokumente gerecht zu werden. Hierbei wird besonderer Wert darauf gelegt, die möglichen Strukturänderungen operationalisierbar zu machen. Mit anderen Worten, einem Anwender soll die Möglichkeit gegeben werden zu entscheiden, welche Arten von Strukturänderungen an der Cluster-Hierarchie durchgeführt werden und welche nicht. Hierfür wird ein skalierbares Maß entwickelt. Im letzten Abschnitt soll dann der Gesamtablauf der Inkrementierung dargestellt und die einzelnen Algorithmen erläutert werden.

6.1 Inkrementierung und ihre Auswirkungen auf die fts-Hierarchie

Der essentielle Bestandteil des Clusteringansatzes, welcher in dieser Arbeit vorgestellt wird, ist das (inkrementelle) Nachfügen oder Erweitern von Dokumenten in eine bereits bestehenden Cluster-Hierarchie. Das heißt, Dokumente, die nach dem initialen Clustering der Newsgroup hinzugefügt werden, sollen in die bestehende Hierarchie aufgenommen werden. Und bereits bestehende Dokumente, die durch einen Beitrag erweitert werden, sollen in die entsprechenden Cluster bzw. fts aufgenommen werden, denen sie durch neu hinzugekommene Terme zusätzlich angehören. Da dies die grundlegenden Aktionen der Inkrementierung sind, sollen sie hier zunächst näher betrachten werden. Dies soll in den folgenden beiden Unterabschnitten anhand eines Artikels exemplarisch geschehen.

Allerdings muss nicht nur das Zufügen neuer Dokumente oder die Veränderung in bestehenden Dokumenten isoliert berücksichtigt werden, sondern auch die Auswirkungen auf die Gesamtstruktur des Clusterings. So kann es durch die Erhöhung der Dokumentenanzahl zu Veränderungen in der Menge der *frequent term sets* kommen. Es können neue fts hinzukommen, aber auch alte unter die Schranke für den *MinSupport* sinken. Somit würde die Menge der Clusterkandidaten bei einem erneuten Clustering anders ausfallen und es würde unter Umständen auch eine andere Cluster-Hierarchie erzeugt werden. Diese Veränderungen sollen dann in den Unterabschnitten 6.1.3 und 6.1.4 näher betrachtet werden.

6.1.1 Neues Dokument

Ein neuer Thread wird eröffnet. Nachdem die Vorverarbeitungsschritte abgeschlossen sind, besteht für das neue Dokument seine Termvektordarstellung und es kann den ein-

zelnen Clustern zugeordnet werden[1]. Beginnend an der Wurzel wird überprüft, ob das Dokument Terme enthält, die bestimmend für Kind-Cluster (*frequent term set*) sind[2]. Wenn dies der Fall ist, wird das Dokument den entsprechenden Clustern zugeordnet, wenn nicht kommt das Dokument in den entsprechenden Misc-Cluster. Wurde das Dokument einem Cluster (außer dem Misc-Cluster) zugeordnet, wird in diesem wieder nachgesehen, ob das Dokument bestimmende Cluster-Terme besitzt und es wird wieder entsprechend zugeordnet. Dies wird solange rekursiv wiederholt, bis das Dokument, in alle Cluster aufgenommen wurde, deren fts es enthält. Durch dieses Vorgehen bleibt die dem initialen Clusteringverfahren innewohnende Semantik bewahrt, dass jedes Dokument eines Clusters die Terme enthält, die den Cluster definieren.

Aus dem Zufügen eines gänzlich neuen Dokumentes kann die Veränderung der MinSupport-Schranke folgen. Da diese Schranke eine prozentuale ist, kann der Wert der minimal notwendigen Dokumente mit einem neu aufgenommenen Dokument steigen und eine Termmenge, die zuvor ein fts war, aber das neue Dokument nicht *erhalten* hat, kann unter diese Schranke sinken und somit ihren Status als *häufige Termmenge* verlieren. Andererseits kann die absolute Schranke auch gleich bleiben und ein Term oder eine Menge von Termen kann den Schwellwert überschreiten und so ein fts werden.

6.1.2 Erweitertes Dokument

Ein Dokument, das besteht, wird erweitert, es kommt also ein weiterer Beitrag (Kommentar) zu einem Thread hinzu. In diesem Fall muss ähnlich verfahren werden, wie im Falle eines neuen Dokumentes. Die fts-Hierarchie muss durchsucht werden, um die Cluster zu lokalisieren, denen das Dokument zugeordnet werden kann. Allerdings ist es häufig der Fall, dass einem Cluster das Dokument bereits zugeordnet ist. In diesem Fall ist nichts zu tun. Ansonsten wird das Dokument dem Cluster zugeordnet. Es muss aber darauf geachtet werden, ob das Dokument das erste Mal einem Kind-Cluster zugeordnet wird und vorher aber bereits dem Elter-Cluster zugeordnet war. Wenn dies der Fall ist, wird das Dokument bisher im Misc-Cluster unter dem Eltern-Cluster zu finden sein und muss dementsprechend dort jetzt entfernt werden.

Die Erweiterung eines Dokumentes zieht in keinem Fall die Veränderung der absoluten MinSupport-Schranke nach sich. Ein Term oder eine Menge von Termen, dem ein erweitertes Dokument zugeordnet wurde, kann diese Schranke überschreiten und somit ein fts werden.

6.1.3 Neue fts

Ein neues fts kann, wie oben dargestellt, in zwei Fällen entstehen. Zum einen kann ein neues Dokument in die Newsgroup aufgenommen werden, aber die absolute Schranke des MinSupports bleibt unverändert. In diesem Fall kann, wie bereits in 6.1.1 beschrieben, ein neues fts aus Termmenge entstehen, die in dem Dokument enthalten sind. Zum anderen kann ein Dokument erweitert werden. In diesem Fall bleibt die absolute Minsupport

[1] Für die späteren strukturellen Überprüfungen und Änderungen wird das Dokument auch allen Clusterkandidaten - also jedem fts der fts-Hierarchie - zugeordnet.

[2] Der Term, der in der Termmenge des Clusters, aber nicht in der Termmenge des Eltern-Clusters vorhanden ist, ist bestimmend für den Cluster. Unter dem Wurzelcluster sind natürlich alle Terme der 1-fts bestimmende Terme.

Grenze in jedem Fall gleich und einem Term oder einer Menge von Termen wird das Dokument neu zugeordnet und überschreitet dadurch den Grenzwert. Dies sind rein strukturelle Betrachtungen, denen aber auch eine inhaltliche Interpretation zugrunde gelegt werden kann.

Newsgroups sind einem zeitlichen Wandel unterworfen, was die in ihnen behandelten Thematiken anbelangt. Wenn man eine Newsgroup betrachtet, die beispielsweise eine Software wie Microsoft Access zum Thema hat, so wird mit dem Erscheinen einer neuen Version dieser Software auch das Auftreten neuer Fragen verbunden sein. Inhaltlich kann das Erscheinen eines neuen fts also bedeuten, dass eine Thematik in die Newsgroup neu aufgenommen wurde oder stärkere Bedeutung erlangt. Es sollte in diesem Fall also geprüft werden, ob ein neuer Cluster in die Hierarchie aufgenommen wird.

6.1.4 Wegfall von fts

Das Wegfallen eines fts kann nur auftreten, wenn eine neues Dokument hinzu gekommen ist und die absolute MinSupport Grenze gestiegen ist. In diesem Fall kann ein fts, welches die Bedingung bisher genau erfüllt hat, unter die Grenze sinken. Hier ist die inhaltliche Interpretation des Vorgangs analog zu der eines neuen fts.

So wie neue Thematiken Einzug in eine Newsgroup halten, können auch alte entfallen. Beispielsweise kann das Erscheinen der neuen Version einer Software dazu führen, dass die Anfragen zur alten Version abnehmen und damit im Verhältnis weniger Dokumente sich auf diese Version beziehen. Somit werden die mit der alten Version verbundenen Schlagworte ebenfalls weniger häufig auftreten und vorhandene häufige Termmengen verlieren eventuell ihren Status als *häufig*. Wenn es sich um fts handelt, die keinen Cluster in der bestehenden Cluster-Hierarchie hervorgebracht haben, so braucht hier nichts Weiteres getan zu werden. Wenn aber doch, so muss der betroffene Cluster aus der Hierarchie entfernt werden.

6.1.5 Beispiel

An dieser Stelle soll noch einmal das Beispiel aus Abbildung 5.1 in Abschnitt 5.1.1 aufgegriffen werden, um die oben beschriebenen Vorgänge zu erläutern. In diesem Beispiel wurden die drei Terme *databas*,*db2* und *oracle* und ihre Zuordnung zu insgesamt zehn Dokumenten betrachtet. Dieses Beispiel soll hier um die zwei Terme *xml* und *java* erweitert werden. Die erweiterte Zuordnungstabelle ist in Abbildung 6.1 zu sehen.

{oracle}	{D1,- -,D3,D4,D5,- -,D7,D8,- -,- -}
{databas}	{D1,D2,- -,D4,D5,- -,D7,- -,D9,D10}
{db2}	{- -,D2,- -,- -,D5,D6,D7,- -,- -,D10}
{***xml***}	{- -,- -,- -,D4,- -,- -,D7,- -,D9,- -}
{***java***}	{- -,D2,- -,- -,- -,- -,D7,- -,- -,D10}

Abbildung 6.1: Erweiterte Zuordnung von Dokumenten zu Termen

Die MinSupport-Schranke von 40% soll auch hier gelten. Die beiden neuen Terme erreichen somit nicht die minimal notwendige Anzahl von vier Dokumenten, so dass die

fts-Hierarchie identisch ist mit der in Abbildung 5.3 gezeigten.

6.1.5.1 Erweiterung von Dokumenten

Wenn nun die Dokumente $D3$ und $D5$ durch weitere Beiträge erweitert werden und diese Beiträge den Term *xml* enthalten, verändert sich die Darstellung der Dokument-Term-Zuordnung, wie in Abbildung 6.2 gezeigt.

{oracle}	{D1,- -,D3,D4,D5,- -,D7,D8,- -,- -}
{databas}	{D1,D2,- -,D4,D5,- -,D7,- -,D9,D10}
{db2}	{- -,D2,- -,- -,D5,D6,D7,- -,- -,D10}
{***xml***}	{- -,- -,*D3*,D4,*D5*,- -,D7,- -,D9,- -}
{***java***}	{- -,D2,- -,- -,- -,- -,D7,- -,- -,D10}

Abbildung 6.2: Neuer Beitrag für $D3$ und $D5$

Da der Term *xml* nun in insgesamt fünf Termen enthalten ist und sich die absolute MinSupport-Schranke nicht verändert hat, ist der Term nun auch häufig. Des Weiteren ist der Schnitt mit dem Term *databas* auf vier Dokumente angewachsen, so dass auch die Termemenge $\{databas, xml\}$ zu einem fts geworden ist.

Die entsprechende fts-Hierarchie ist in Abbildung 6.3 dargestellt.

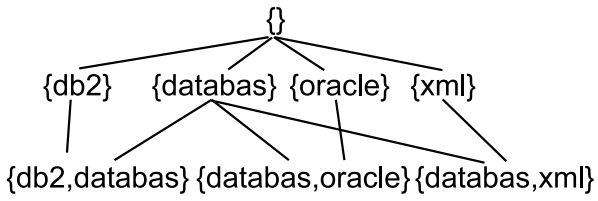


Abbildung 6.3: Neue fts-Hierarchie

6.1.5.2 Neue Dokumente

In Abbildung 6.4 wird die Erweiterung der Dokumentenmenge um drei Dokumente gezeigt. Die absolute MinSupport-Schranke steigt damit auf fünf Dokumente. Da der Term *java* nun in zwei weiteren Dokumenten vorkommt, kann auch er diese Schranke überschreiten und wird damit häufig. Die anderen Terme bleiben mit Blick auf die fts-Hierarchie unverändert.

Bei den k-fts mit k $>$ 1 gibt es weitere Änderungen. Die Termmenge $\{databas, db2\}$ kommt in keinem der neuen Dokumente vor und da es daher weiterhin nur vier Dokumente gibt, welche diese Termmenge beinhalten, ist sie nicht mehr häufig und fällt somit aus der fts-Hierarchie. Dafür ist, wie in Abbildung 6.5 gezeigt wird, das fts $\{databas, oracle, xml\}$ neu hinzu gekommen.

{oracle}	{D1,- -,D3,D4,D5,- -,D7,D8,- -,- -,D11,- -, - -}
{databas}	{D1,D2,- -,D4,D5,- -,D7,- -,D9,D10,D11,- -,D13}
{db2}	{- -,D2,- -,- -,D5,D6,D7,- -,- -,D10,- -,D12, - -}
{xml}	{- -,- -,D3,D4,D5,- -,D7,- -,D9,- -,D11, - -,D13}
{java}	{- -,D2,- -,- -,- -,- -,D7,- -,- -,D10, - -,D12,D13}

Abbildung 6.4: Neue Dokumente

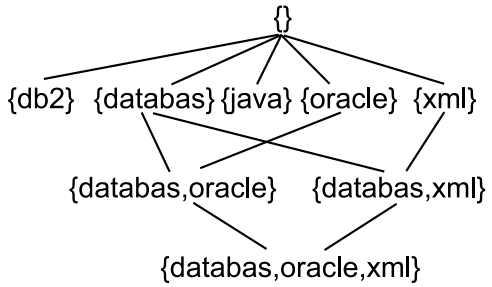


Abbildung 6.5: fts-Hierarchie nach Zufügen neuer Dokumente

6.2 Strukturänderungen

Die ersten Abschnitte dieses Kapitels haben sich mit den grundlegenden Vorgängen der Inkrementierung beschäftigt. Und es wurden die Effekte auf die fts-Hierarchie aufgezeigt. Nun soll auch auf die daraus folgenden Änderungen an der hierarchischen Struktur des Clusterings eingegangen werden. Hierbei soll das Augenmerk auf bestimmte, aus Sicht des Anwenders wichtige Gesichtspunkte gelegt werden.

Wie bereits erwähnt, ist es für den Anwender, der das Clustering für seine Zwecke verwendet, wichtig, dass die hierarchische Struktur sich möglichst wenig verändert[3]. Allerdings sollte die Qualität des Clusterings unter dieser Prämisse nicht zu stark leiden. Es ist davon aus zu gehen, dass das Zufügen von weiteren und das Erweitern von bestehenden Dokumenten zur Folge hat, dass ein erneutes initiales Clustering mit der so entstandenen erweiterten Dokumentenmenge eine andere Clusterstruktur hervorbringen würde, als dies der Fall beim ersten Clustering war. Da das hier beschriebene Clusteringsverfahren auf der Minimierung des in Abschnitt 5.2 beschriebenen Kostenmaßes beruht, ist davon auszugehen, dass ein komplettes *Reclustering* geringere Kosten, also eine bessere Qualität aufweist, als die triviale inkrementelle Erweiterung (die triviale Erweiterung ist das einfache Einordnen der Dokumente in die bestehenden Cluster). Es ist also wichtig zu prüfen, welche strukturellen Änderungen an der Hierarchie die Güte der Inkrementierung so weit verbessern, dass der erwartete Qualitätsverlust akzeptabel ist. Auch ist es notwendig, die Performance der einzelnen Strukturänderungen zu beachten, da diese bei jedem Hinzufügen von Beiträgen zur Newsgroup durchgeführt werden müssen. Im Gegensatz zum initialen Clustering, welches einmalig ausgeführt wird, ist die Inkrementierung recht häufig durchzuführen. Also sollte sie in einer für den Anwender

[3] Hier ist die Cluster-Hierarchie gemeint. Dem Anwender wird die fts-Hierarchie nicht gezeigt.

akzeptablen Zeit durchgeführt werden können.

Im Folgenden sollen die möglichen strukturellen Veränderungen im einzelnen beschrieben so wie ihre Auswirkungen auf die Cluster-Hierarchie und ihre performancetechnischen Details näher beleuchtet werden. Die Sichtweise des Anwenders soll operationalisierbar gemacht werden. Um dies zu ermöglichen, wird jeder möglichen Strukturänderung ein *Kostenmaß*[4] angeheftet. Dieses Maß soll später dem Benutzer die Möglichkeit geben zu entscheiden, welche strukturellen Änderungen durchgeführt werden dürfen und welche nicht. Weiterhin dient das Maß auch dazu, bei der späteren Evaluation entscheiden zu können, welche strukturellen Änderungen an einem Clustering durchgeführt werden müssen, um qualitativ nicht zu stark gegen ein erneutes Reclustering ab zu fallen. Der Grad der Schwere von strukturellen Änderungen soll mit einer Punkteskala gemessen werden. Jede mögliche Änderung erhält hierfür eine Wertigkeit, die in Punkten, den sogenannten SCP (Structural Change Points) angegeben wird.

Definition 6.2.1 (**Structural Change Points** und **Strukturänderungsstufe**)**.** Die Angabe eines *SCP*-Wertes aus den natürlichen Zahlen, reguliert die Art und die *Schwere* der Strukturänderungen, welche an der Cluster-Hierarchie durchgeführt werden dürfen. Der von einem Anwender angegebene SCP-Wert bezeichnet die *Strukturänderungsstufe*.

In den folgenden Abschnitten sollen vier Strukturänderungen vorgestellt werden. Jede dieser Strukturänderungen wird in zwei Unterabschnitten näher erläutert. Im ersten wird jeweils die Strukturänderung einleitend beschrieben. Im zweiten Unterabschnitt wird dann die Vergabe der SCP erläutert.

Das gesamte Verfahren der Inkrementierung wird dann in Abschnitt 6.3 ausgeführt.

6.2.1 Reclustering eines Misc-Clusters

Ein (Eltern-)Cluster enthält auf seiner Nachfolger-Ebene einen sogenannten *Misc*-Cluster, welcher die Dokumente aufnimmt, die keinem der ausgewählten Cluster zugeordnet sind. Ein Dokument kann entweder nicht zugeordnet werden, weil es in keinem Clusterkandidaten enthalten ist oder weil keiner der Clusterkandidaten, in denen es enthalten ist, ausgewählt wurde. Im zweiten Fall hat das initiale Clusteringverfahren keinen der entsprechenden Kandidaten ausgewählt, da keiner zu einer Minimierung der globalen Kosten des Eltern-Clusters beigetragen hätte. Wenn nun weitere Dokumente dem Misc-Cluster zugeordnet werden und diese Dokumente in Clusterkandidaten, aber eben nicht in gewählten Clustern enthalten sind, so hat dies automatisch ein Ansteigen der Kosten für nicht genutzte Möglichkeiten (siehe 5.2.2) zur Folge. Durch diese Veränderung in der Zusammensetzung ist es möglich, dass das Hinzunehmen eines Clusterkandidaten die lokalen Kosten senkt. Auch kann es sein, dass die globalen Kosten eines Kandidaten selbst gesunken sind. Wenn ein weiterer Clusterkandidat nach dem initialen Clustering ausgewählt wird, um bisher nicht abgedeckte Dokumente zu überdecken, so wird dies als *Reclustering* des entsprechenden Misc-Clusters bezeichnet.

Um entscheiden zu können, ob weitere Clusterkandidaten zur Überdeckung eines (Eltern-)Clusters herangezogen werden sollen, müssen bestimmte Grundvoraussetzungen erfüllt werden. Zum einen müssen für den Eltern-Cluster und seine Kind-Cluster die globalen Kosten gepflegt werden, da diese das entscheidende Kriterium für das Clustering

[4] Dieses Kostenmaß hat nichts mit dem Kostenmaß des Clusterings gemein, sondern soll die *schwere* der Strukturänderung aus Sicht des Anwenders objektivieren.

sind. Zum anderen muss für alle Clusterkandidaten nachgesehen werden, ob sie unter die MinSupport-Schranke gesunken sind. Ist dies der Fall, müssen die Kosten des Eltern-Clusters, globale wie lokale, entsprechend aktualisiert werden. Fällt ein Clusterkandidat unter den MinSupport, so ist er *invalide* und darf nicht mehr zur Menge der Clusterkandidaten des Eltern-Clusters gezählt werden.

Definition 6.2.2 (Invalide Cluster(-kandidaten)). Ein Cluster oder ein Clusterkandidat wird als *invalide* bezeichnet, wenn er bzw. seine Menge definierender Terme nicht mehr den MinSupport erfüllt, also nicht mehr in einer ausreichenden Anzahl von Dokumenten vorkommt, um häufig zu sein.

Ist ein Cluster(-kandidat) Teil der Clusterhierarchie (also ist er von der Wurzel aus über einen Pfad, welcher nur gewählte Cluster enthält, zu erreichen), so ist er ein gewählter Cluster für mindestens einen seiner Eltern-Cluster. Für diesen und eventuell weitere Cluster darf er nicht gestrichen werden, solange dies nicht ausdrücklich gefordert ist (siehe 6.2.2). Dadurch sinken die lokalen Kosten des Eltern-Clusters, da die Summe ungenutzter Möglichkeiten sich verringert. Dies ist sehr wahrscheinlich auch der einzige Kosteneffekt eines entfallenden Clusterkandidatens auf seinen Eltern-Cluster, weil er nicht über eigene globale Kosten verfügt.

Structural Change Points

Die Hinzunahme eines Clusters in eine bestehende Cluster-Hierarchie bedeutet für den Anwender keinen schwerwiegenden Eingriff in die Struktur. Der Anwender bekommt lediglich die Möglichkeit, Dokumente in einem neuen Cluster zu entdecken. Allerdings stellt es nichtsdestotrotz einen Eingriff dar und wird mit einem Strukturänderungspunkt (*Structural Change Point* - SCP) bewertet.

6.2.2 Entfernen eines Clusters wegen Unterschreiten des MinSupportes

Wie in Abschnitt 6.1.4 erläutert, können fts während des inkrementellen Prozesses den MinSupport unterschreiten. Solange das fts keinen gewählten Cluster in der Clusterhierarchie hervorgebracht hat, sondern nur als Kandidat gehandelt wurde, hat diese Unterschreitung keine direkten Folgen für die bestehende Cluster-Hierarchie. Sobald es sich aber um einen Cluster in der Cluster-Hierarchie handelt, muss dieser aus der Struktur entfernt werden und die Dokumente, welche nur durch diesen Cluster abgedeckt sind müssen dem Misc-Cluster zugeordnet werden. Dieser Aktion sollte ein Reclustern des entsprechend Misc-Clusters folgen, wie es im vorherigen Abschnitt 6.2.1 beschrieben wurde.

Structural Change Points

Im Gegensatz zum Reclustering eines Misc-Clusters wird hier ein bestehender Cluster aus der Struktur entfernt. Ein Cluster bietet dem Anwender die Möglichkeit auf Dokumente zuzugreifen, die über bestimmte Attribute identifiziert werden (hier die entsprechenden Termmengen). Diese Möglichkeit wird dem Anwender durch das Löschen eines Clusters genommen. Andererseits wird vermutlich ein Cluster, der so nah an der Support-Grenze war, nicht so wichtig sein wie ein anderer. Und er wird auch keine Kinder

haben, da diese auch unterhalb der MinSupport-Schranke lägen. Als Folge erhält diese Strukturänderung zwei SCP.

In dieser Betrachtung wird kein Fall weiter aufgeschlüsselt, in dem ein invalider Cluster selbst über Kind-Cluster verfügt. Seine Entfernung aus der Cluster-Hierarchie also bedeutet, dass kein Blatt, sondern ein Teilbaum entfernt wird. Hierfür könnte eine eigene Bewertung der Strukturänderung vorgenommen und SCP-Werte (je nach größe des Teilbaumes) vergeben werden. Dies soll allerdings nicht geschehen. Der Grund hierfür ist der, dass dieser Fall (invalid werden eines Nicht-Cluster-Blattes) nur höchst selten auftritt[5]. Daher würde eine Fallunterscheidung nicht oder nur in Ausnahmefällen zu unterschiedlichen Ergebnissen in der Inkrementierung führen.

6.2.3 Entfernen eines validen Blattes

Die beiden ersten Strukturänderungen bezogen sich auf Fälle, die den trivialen Änderungen an den fts geschuldet sind. Der in 6.2.2 beschriebene Vorgang des Entfernens eines invalid gewordenen Clusters ist quasi ein deterministischer Vorgang, da die grundlegende Bedingung für das Bestehen des Clusters, nämlich die Häufigkeit seiner Termmenge nicht mehr gegeben ist. Und das Reclustering der Misc-Cluster ist im Grunde nur eine Fortsetzung des initialen Clusterings. Im Folgenden soll eine Strukturänderung beschrieben werden, die darauf beruht, dass sich die Verteilung innerhalb der Überdeckung eines Clusters wandelt. Und zwar sollen valide[6] Blätter innerhalb einer Cluster-Hierarchie betrachtet werden.

Diese Blätter tragen auf eine einfach nachvollziehbare Weise zu den globalen Kosten ihres Eltern-Clusters bei. Da sie selbst keine globalen Kosten haben, gehen sie in die globalen Kosten (siehe Formel 5.7) ihres Eltern-Clusters nur über den Faktor der lokalen Kosten ein. Ihr Beitrag ist am besten über die in dem Beweis in Abschnitt 5.3.2.1 definierte Kostendifferenz zu erläutern. Die Kostendifferenz eines gewählten Clusters C_i, ist ja gerade die Differenz aus den Kosten für die Überschneidungen mit anderen gewählten Clustern und der Verringerung der Kosten für nicht genutzte Möglichkeiten. Es ist also leicht nachzuvollziehen, dass der Cluster ein kostensenkender Faktor ist, solange er weniger Überlappungskosten verursacht als er Kosten für nicht genutzte Möglichkeiten *einspart*. Doch sobald dieses Verhältnis umschlägt, fallen die lokalen und die globalen Kosten des Eltern-Clusters geringer aus, wenn der betrachtete Cluster entfernt wird.

Wenn ein valides Blatt aus der Menge der gewählten Cluster eines Eltern-Clusters entfernt wird, so müssen die nun nicht mehr abgedeckten Dokumente in den Misc-Cluster verlagert werden.

Die globalen Kosten eines Clusters können aber nicht nur verbessert werden, wenn ein Blatt entfernt wird, welches mehr Überlappungskosten verursacht, als es Kosten nicht genutzter Möglichkeiten einspart. Auch das Austauschen eines Cluster(-Blattes) mit einem oder mehreren Kandidaten kann die globalen Kosten senken. Um dies zu prüfen, wird das entsprechende Blatt entfernt, die nicht mehr abgedeckten Dokumente dem Misc-Cluster zugeordnet und anschließend ein Reclustering des Misc-Clusters durchgeführt. Die sich neu ergebenden Kosten werden mit denen vor dem Reclustering verglichen. Sind sie besser, wird die Ersetzung entsprechend durchgeführt.

[5] Bei der Evaluation ist es nicht vorgekommen.

[6] Mit valide ist hier gemeint, dass ihr erzeugendes fts nicht den MinSupport unterschritten hat.

Structural Change Points

Wie im beim Wegfall eines Clusters durch Unterschreiten des MinSupportes wird auch hier ein Blatt entfernt. Jedoch ist das Entscheidungskriterium an dieser Stelle der Kostenfaktor. Der Anlass für das Entfernen ist also nicht zwingend, so wie im Fall eines Clusters, dessen Termmenge den MinSupport unterschritten hat. Daher wird diese Aktion mit drei SCP bewertet.

6.2.4 Entfernen eines inneren Knoten

Statt nur Blätter zu entfernen oder auszutauschen, kann es auch sinnvoll sein, ganze Teilhierarchien (also innere Knoten) zu entfernen. Allerdings ist hier das Entscheidungskriterium nicht so einfach nachvollziehbar, wie bei den vorangegangenen Strukturänderungen, da ein innerer Knoten nicht nur über die lokalen Kosten zu den globalen Kosten seines Eltern-Clusters beiträgt, sondern auch durch seine eigenen globalen Kosten. Bei einem Blatt ist prinzipiell die Überprüfung der Kostendifferenz ausreichend um festzustellen, ob der entsprechende Cluster aus der Überdeckung seines Eltern-Clusters genommen werden kann, um die globalen Kosten zu senken. Bei einem gewählten Cluster, der selbst auch über Kinder verfügt, ist es notwendig, die Formel der globalen Kosten 5.7 mit und ohne diesen Cluster zu berechnen. Hat der gewählte Cluster bei der Inkrementierung zusätzliche Dokumente erhalten, so bedeutet dies nicht nur, dass sich die Verteilungsverhältnisse des Eltern-Clusters verändert haben. Auch die Verteilung innerhalb des betroffenen Kind-Clusters haben sich verändert[7], woraus auch eine Veränderung der globalen Kosten dieses Cluster folgt.

Die Entscheidung für das Entfernen eines Clusterkandidaten (mit eigenen Kind-Clustern) aus der Menge der gewählten Cluster des Eltern-Clusters, ist also nur durch ein *try-and-error* Verfahren zu fällen, da das Entscheidungskriterium die Veränderung in den globalen Kosten ist.

Structural Change Points

Da durch das Entfernen eines inneren Knoten ein ganzer Teilbaum verloren geht, ist die strukturelle Veränderung natürlich als schwerwiegender zu bewerten als bei den vorangegangenen Punkten. Auch kann diese nicht pauschalisiert betrachtet werden, da es sich um Teilhierarchien von sehr unterschiedlicher Art und Größe handeln kann. So erscheint hier eine Punktstaffelung angemessen, die diesen Gesichtspunkten Rechnung trägt. Diese Staffelung ist rekursiv angelegt, um die Strukturkomplexität einer Teilhierarchie angeben zu können. Als Grundpunktzahl soll, wie bei dem Entfernen eines validen Cluster-Blattes, drei SCP gelten. Zusätzlich werden für jede Hierarchiestufe unter dem betrachteten Cluster, ein SCP bei genau einem Nachfahren-Cluster und zwei SCP bei mehr als einem Nachfahren-Cluster[8] zu der Grundpunktzahl addiert. Die Abbildung 6.6 zeigt ein Beispiel für die Vergabe der SCP.

Wie zu sehen ist, haben Blätter einen SCP von drei. Dies deckt sich mit der Definition aus Abschnitt 6.2.3. Das Entfernen eines validen Blattes ist somit ein Spezialfall des Entfernens eines inneren Knoten, wie es hier definiert wird. Der Cluster C hat nur

[7] Es sei denn, für keines der neuen Dokumente gibt es einen Kandidaten in dem Kind-Cluster.
[8] Misc-Cluster zählen hier nicht.

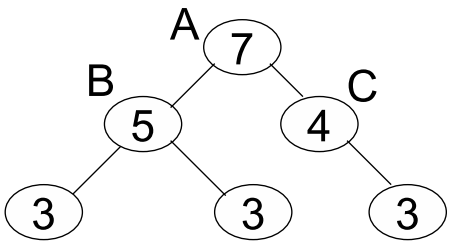


Abbildung 6.6: Vergabe der *Structural Change Points* (SCP)

einen Kind-Cluster. Er hat also nur eine Ebene mit Nachfahren und dort auch nur einen Nachfahren. Daher bekommt der Cluster C einen zusätzlichen SCP und hat damit insgesamt vier SCP. Der Cluster B verfügt über zwei Kind-Cluster, welche beide Blätter sind. Damit erhält Cluster B die maximale zusätzliche Punktzahl von zwei SCP und hat insgesamt fünf SCP. Cluster A hat die beiden Cluster B und C als Kind-Cluster und erhält zwei zusätzliche SCP für diese Ebene der direkten Nachfahren-Cluster (Kinder) und nochmals zwei für die nächste Ebene (Enkel-Cluster), da es auf dieser Ebene auch einmal mindestens zwei Nachfahren-Cluster (bei Cluster B) gibt und hat somit insgesamt sieben SCP.

Die SCP-Berechnung lässt sich durch folgende Fallunterscheidung zusammenfassen:

1. Blatt-Cluster: Per Definition drei SCP.
2. Cluster mit einem Kind: SCP des Kind-Clusters + 1
3. Cluster mit mehr als einem Kind: Maximaler SCP der Kind-Cluster + 2

Bei der Berechnung der SCP handelt es sich um eine rekursive Vorgehensweise, da für die Berechnung des SCP Wertes eines Clusters die SCP Werte seiner Kind-Cluster bekannt sein müssen. Die Formel 6.1 gibt diese rekursive Berechnung formal wieder.

$$SCP(C_i) = max(SCP(C_j) \| C_j \prec C_i) + \begin{cases} 1, & falls\, |M_i| = 1 \\ 2, & sonst \end{cases} \tag{6.1}$$

Nach der Erläuterung der Vergabe der SCP an die Cluster-Knoten soll hier nur kurz anhand eines Beispieles auf die Auswirkungen der vom Anwender gesetzten Strukturänderungsstufe eingegangen werden. Eine detaillierte Erörterung des Verfahrens wird in Abschnitt 6.3.5 durchgeführt.

Angenommen, der Anwender setzt vor Durchführung einer Inkrementierung die Strukturänderungsstufe auf vier. So wäre es erlaubt, den Knoten C in Abbildung 6.6, aus der Menge der für Knoten A gewählten Cluster zu entfernen. Das Entfernen von Knoten B wäre wiederum nicht erlaubt. Es ist allgemein nicht von Bedeutung für das Verfahren, wie viele Cluster entfernt werden, solange ihre jeweiligen SCP die gesetzte Strukturänderungsstufe nicht überschreiten.

6.2.5 Zusammenfassung

In den vergangenen Abschnitten wurden die möglichen Strukturänderungen an einer Cluster-Hierarchie beschrieben. Dabei wurden auch die *Structural Change Points* erläu-

tert, welche in Verbindung mit der *Strukturänderungsstufe* entscheiden, ob ein bestimmter Typ von Strukturänderung durchgeführt werden darf oder nicht. Der Anwender legt hierfür die Strukturänderungsstufe fest, welche während der Inkrementierung gelten soll. Die Strukturänderungsstufe ist eine natürliche Zahl ≥ 0. Beim Inkrementieren der Newsgroup dürfen anschließend nur Strukturänderungen durchgeführt werden, welche die angegebene Strukturänderungsstufe nicht überschreiten.

Die Tabelle 6.1 fasst die möglichen Strukturänderungen, die jeweils vorausgesetzte Strukturänderungsstufe und die damit verbundenen Aktionen zusammen.

SCP	**Strukturänderung**	**Aktionen**
0	keine	Einfügen neuer und geänderter Dokumente
1	Misc-Reclustering	1.Neu entstandene fts werden eingefügt 2.Bottom-up Reclustering der Misc-Cluster
2	Entfernen invalider Cluster	1.Entfernen aller invalider Cluster 2.Bottom-up Reclustering der Misc-Cluster
3	Entfernen valider Cluster-Blätter	Austausch valider Cluster-Blätter, wenn die globalen Kosten dadurch sinken
>3	Entfernen valider Cluster	Austausch valider Cluster mit SCP $\leq$ der angegebenen Strukturänderungsstufe, wenn die globalen Kosten dadurch sinken

Tabelle 6.1: Zusammenfassung der Strukturänderungen

Die Definition der SCP-Werte für valide Teilbäume (6.2.4) lässt nur eine relativ grobe Unterscheidung und Regulierung für die Strukturänderungen zu. Denkbar wäre auch die Vergabe von SCP anhand der Gesamtclusteranzahl, die sich in einem Teilbaum befindet. Dies würde eine feinere Unterscheidung der Fälle zulassen. Wie Tests ergeben haben, sind die Unterschiede in der Qualität zwischen den gewählten Stufen bei einer solch feinen Abstufung jedoch nicht merklich besser zu beobachten als bei der hier präferierten. Andererseits ist eine noch gröbere Abstufung (SCP-Wert anhand der Tiefe eines Teilbaumes) auch nicht ratsam, da dann viele Änderungen nicht mehr beobachtet werden können.

6.3 Gesamtablauf der Inkrementierung

In den vorangegangenen Abschnitten wurden die möglichen Strukturänderungen im Einzelnen abstrakt beschrieben. In diesem Abschnitt wird nun der Gesamtablauf der Inkrementierung erläutert. Hierbei werden die notwendigen algorithmischen Vorgänge, mit Blick auf die vom Anwender gewählte Strukturänderungsstufe näher beleuchtet. Denn durch die Eingabe eines maximalen SCP Wertes wird festgelegt, welche Änderungen an der Struktur der Cluster-Hierarchie durchgeführt werden dürfen. Daher sollen die einzelnen Strukturänderungen und auch alle anderen Vorgänge (Hinzufügen der neuen Dokumente, Erweitern der fts-Hierarchie, etc.) in der Reihenfolge beschrieben werden, in der sie nach aufsteigenden SCP-Wert ausgeführt werden.

6.3.1 Einfügen von Dokumenten

Der erste Schritt der Inkrementierung ist trivialerweise das Hinzufügen der neuen oder geänderten Dokumente zu den entsprechenden fts. Die Dokumente werden also der fts-Struktur und damit implizit auch der Clusterhierarchie zugewiesen. Es sei noch einmal bemerkt, das ein fts auch immer ein Clusterkandidat ist und in manchen Fällen auch ein gewählter Cluster. Aus diesem Grund werden die Begriffe Cluster(-kandidat)und *frequent term set* hier synonym verwendet.

Die Prozedur **addDocuments** in der Abbildung 6.7 zeigt das Vorgehen beim Einfügen von neuen oder geänderten Dokumenten. Die Unterscheidung zwischen neuen und geänderten Dokumenten wird im Weiteren an den Stellen hervorgehoben wo sie wirklich relevant ist. Ansonsten wird der Ausdruck neue Dokumente gebraucht. Im ersten Teil der Prozedur werden diese Dokumente den 1-fts zugewiesen. Hierbei macht sich die Prozedur zunutze, dass die 1-fts genau aus einem Term bestehen und daher einfach zu entscheiden ist, ob ein Dokument einem 1-fts zugeordnet werden kann[9]. Ist dies der Fall, so wird das entsprechende Dokument in die Menge der Dokumente des fts aufgenommen (Zeile 5). Wichtig ist hierbei, dass durch die Mengeneigenschaft ein Dokument nicht mehrfach in den fts und Clustern vorkommen kann.

```
FTSList          := List with initial all 1-fts
DocumentList     := List with new or changed Documents

addDocuments(FTS )
1    ClusterQueue := Empty
2    For each FTS ∈ FTSList
3        For each Document ∈ DocumentList
4            If Term(FTS) ∈ Document Then
5                Documents(FTS) ∪ Document
6            End If
7        End For
8        ClusterQueue ∪ Descendants(FTS)
9    End For
10   While (ClusterQueue Not Empty)
11       Documents := DocumentList
12       Cluster := Next from ClusterQueue
13       For each Ancestor ∈ Ancestor(Cluster)
14           Documents ∩ Documents(Ancestor)
15       End For
16       Documents(Cluster) ∪ Documents
17       ClusterQueue ∪ Descendants(Cluster)
18   End Loop
```

Abbildung 6.7: Einfügen neuer und geänderter Dokumente

In der äußeren Schleife (Zeile 2) werden alle 1-fts durchlaufen und für jedes fts wird dann in einer inneren (Zeile 3) Schleife über alle neuen Dokumente iteriert. In der äußeren Schleife werden auch alle Kinder der fts einer Warteschlange (*ClusterQueue*) zugewiesen. Diese Warteschlange wird dann im Anschluss genutzt, um auch über alle weiteren fts ite-

[9] Ob das Dokument also den Term enthält

rieren zu können. Die Warteschlange funktioniert nach dem *fifo*(first-in-first-out) Prinzip. Es wird immer vom Anfang entfernt (Zeile 12) und am Ende eingefügt (Zeile 17). Wobei auch hier die Mengeneigenschaft vorausgesetzt wird. Es kann also kein Element zweimal enthalten sein. Dies ist notwendig, da ja zwei k-fts ein gemeinsames (k+1)-fts als direkten Nachfahren haben können. Im Weiteren wird für jeden in der Warteschlange enthaltenen Cluster über die Menge seiner direkten Vorfahren iteriert (Zeile 13) und die Schnittmenge der Dokumentmengen aller Vorfahren des Clusters gebildet (Zeile 14)[10]. In der so ermittelte Dokumentmenge enthält jedes Dokument die Termmenge des betrachteten Clusters. Diese Dokumente können also dem Cluster zugefügt werden (Zeile 16). Zum Abschluss der Betrachtung des Clusters wird die Menge der Cluster-Kinder an das Ende der Warteschlange eingefügt. Durch das *fifo* Prinzip wird sichergestellt, dass die Eltern-Cluster vor ihren Kind-Clustern die neuen Dokumente erhalten haben.

Am Ende diese Abschnittes soll der Zeitaufwand für diesen Schritt abgeschätzt werden. Sei die Anzahl der neuen Dokumente mit d bezeichnet. Die Anzahl der 1-fts mit k und die Anzahl aller fts mit n. Für den ersten Teil (Zeilen 2-9) kann die obere Schranke für die Laufzeit mit $O(k*d)$ abgeschätzt werden. Im zweiten Teil wird über alle fts (abzüglich der 1-fts) iteriert und für jedes fts wird die Menge seiner direkten Vorfahren betrachtet. Die Kardinalität der Vorfahren-Menge ist immer gleich der Anzahl seiner Terme. Da die j-te Ebene der fts-Hierarchie alle j-fts enthält, ist die Maximale Anzahl von Vorfahren also durch die Tiefe der fts-Hierarchie limitiert. Sei die Tiefe der Hierarchie mit l angegeben, somit gilt für die obere Schranke der Laufzeit $O(n * l)$.

6.3.2 Erweitern der fts-Hierarchie

Das Hinzufügen der Dokumente für die Inkrementierung ist unabhängig davon, welche Strukturänderungen durchgeführt werden dürfen. Es muss in jedem Fall ausgeführt werden. Die weiteren Schritte sind aber von dem gewählten SCP abhängig. So ist es nicht notwendig zu überprüfen, ob es neue fts gibt und diese gegebenenfalls in die fts-Hierarchie einzufügen, wenn der SCP-Parameter auf 0 gesetzt ist. In diesem Fall wird das Reclustering der Misc-Cluster nicht durchgeführt, somit müssen auch nicht die Clusterkandidaten der Cluster bekannt sein. Ist aber ein SCP von mindestens eins gewählt, so muss dieser Schritt durchgeführt werden.

Im Prinzip läuft die Erstellung der neuen fts analog zur initialen Erstellung der fts-Hierarchie ab (siehe Abschnitt 5.3.1). Es kann auch die in Abbildung 5.6 gezeigte Prozedur herangezogen werden. Nur werden die Startparameter anders belegt. Die Liste *FrequentTerms* wird mit allen 1-fts belegt, die entweder neu sind oder die mindestens ein Dokument hinzubekommen haben[11]. Die Warteschlange (*FTSQueue*) wird am Anfang der Prozedur wie beim initialen Erstellen der fts-Hierachie mit allen 1-fts befüllt. Der Hashtable (*FTSHash*) wiederum enthält zu Beginn alle bereits bestehenden fts. Ansonsten ist hier die Vorgehensweise die gleiche wie bei der Initiierung. Auf diese Weise

[10] Wenn der Schnitt zweier Dokumentmengen gebildet wird. Wobei in den Dokumenten der ersten Menge, immer der Term A und in den Dokumenten der zweiten Menge immer der Term B enthalten ist, so enthält der Schnitt die Menge an Dokumenten, welche beide Terme beinhalten.

[11] An dieser Stelle sei der Vollständigkeit halber nochmals darauf hingewiesen, dass ein Term nur dann ein fts werden kann, wenn er zum einen den MinSupport überschreitet, also in einer Mindestanzahl von Dokumenten vorkommt *und* wenn er zum anderen vom Anwender als Kandidat zum Clustering ausgewählt wurde.

werden alle neuen fts erzeugt und mit ihren Eltern-fts verbunden.

Die Laufzeit dieses Schrittes ist, wie auch bei der initialen Erstellung der fts-Hierarchie, im schlechtesten Fall exponentiell in der Anzahl der 1-fts. Doch auch hier ist das schnelle Ausscheiden von Termmengen aufgrund des nicht Überschreitens des MinSupportes der Grund dafür, dass dieser Schritt in der Praxis wesentlich effizienter ist, als die worst-case Betrachtung es nahe legt.

6.3.3 Entfernen invalider Cluster

Wenn Dokumente neu hinzukommen, können Cluster(-kandidaten) den MinSupport unterschreiten. Ihre Termmengen unterschreiten also die mindestens notwendige Anzahl an Dokumenten, in denen sie enthalten sein müssen um häufig zu sein. In diesem Fall wird der Cluster(-kandidat) invalide. Ob das entsprechende fts entfernt wird, ist davon abhängig, welcher SCP-Wert gesetzt ist. Ist ein SCP von mehr als eins gesetzt, so wird das fts in jedem Fall entfernt(siehe 6.2.2). Sollte es für alle seine Eltern-fts nur ein Clusterkandidat gewesen sein, so kann es ohne weiteres entfernt werden und nur die Kosten (lokale und globale) der Eltern-fts müssen neu berechnet werden. Ist das fts für ein Eltern-fts ein gewählter Cluster, so müssen zusätzlich zur Neuberechnung der Kosten die nicht mehr überdeckten Dokumente dem Misc-Cluster des Eltern-fts zugeordnet werden. Sobald ein SCP von weniger als zwei gesetzt ist, dürfen nur fts, welche nicht in der Cluster-Hierarchie vorhanden sind, entfernt werden. Jeder Cluster in der Cluster-Hierarchie muss allerdings in dieser und somit auch in der fts-Hierarchie belassen werden. Dies ist der Fall, weil erst ab einer Strukturänderungsstufe von zwei, invalide Cluster aus der Cluster-Hierarchie entfernt werden dürfen. Wird also ein Cluster bzw. ein fts invalide, so darf das fts nicht aus der fts-Hierarchie entfernt werden, da es sich, wie oben ausgeführt, bei der Cluster-Hierachie um eine Unterstruktur der fts-Hierarchie handelt und so der Cluster identisch ist mit dem fts[12]. Um diese Fallunterscheidung bei einem gesetzten SCP von weniger als zwei zu verdeutlichen, soll hier nochmal das Beispiel aus Abbildung 5.11 herangezogen werden. Angenommen sei der Fall, dass die beiden Blatt-fts **K** und **L** invalide werden. Da **L** für die Cluster **H** und **I** der Clusterhierarchie lediglich ein Cluster-Kandidat und für den Cluster **G** zwar gewählt, aber **G** selbst nicht in der Cluster-Hierarchie ist, kann **L** ohne weiteres aus der fts-Hierarchie entfernt werden. Das fts **K** muss allerdings in der fts-Hierarchie verbleiben, da es für den Cluster **H** ein gewählter Cluster ist und somit Teil der Cluster-Hierarchie ist.

Um diesen Schritt durchzuführen, wird die fts-Hierarchie mittels Tiefensuche durchlaufen und in *post-order* für die Kinder jedes fts überprüft, ob sie den MinSupport noch erfüllen. Ist dies für ein Kind-fts nicht der Fall, wird es entsprechend der oben genannten Bedingungen entfernt oder in den Hierarchien belassen. Die Laufzeit dieses Schrittes ist linear in der Anzahl der fts.

6.3.4 Reclustering der Misc-Cluster

Das Erweitern der fts-Hierarchie und das Entfernen invalider Cluster(-kandidaten) schafft die Voraussetzungen für das Reclustern der Misc-Cluster. Dieses wird bei einem SCP ab eins durchgeführt. Der Gesamtablauf ist der gleiche wie beim initialen Clustering

[12] Es handelt sich bei Clustern und fts um die gleichen Objekte in zwei Strukturen, von denen die eine (Cluster-Hierarchie) eine Unterstruktur der anderen (fts-Hierarchie) ist.

(siehe Abschnitt 5.3.2). Die fts-Hierarchie wird „bottom-up“ durchlaufen (siehe Abbildung 5.7) und für jedes fts wird die „branch-and-bound“-Prozedur **doBaBClustering** aus Abbildung 5.8 aufgerufen. Doch werden auch hier andere Startparameter verwendet. Die Menge der Cluster-Kandidaten (*Candidates*) besteht hier wirklich nur aus den bisher nicht gewählten Cluster, eben den verbliebenen Kandidaten. Die Menge der besten bisher gewählten Cluster ist identisch mit den bereits gewählten Clustern.

Mittels der Prozedur **doBaBClustering** wird also nachgesehen, ob einer oder mehrere der verbliebenen Cluster-Kandidaten die globalen Kosten des Eltern-Clusters senken können. Da beim initialen Clustering das optimale oder, wenn ein Approximationsverfahren angewandt wurde, zumindest eine sehr gute Annäherung an das kostenoptimale Clustering gefunden wurde, ist davon auszugehen, dass nur sehr wenige oder gar keine Kandidaten das ursprüngliche Ergebnis verbessern können. Also ist der Zeitaufwand für das Reclustering eines Misc-Clusters als eher gering einzuschätzen, da nur selten eine tiefe Rekursion durchgeführt werden muss. Sei n die Gesamtzahl aller fts und k die Anzahl der 1-fts, so kann die Laufzeit für das Reclustering aller Misc-Cluster der fts-Hierarchie mit $O(n * k)$ abgeschätzt werden, da alle fts durchlaufen werden müssen und jedes j-fts (jeder Cluster) maximal $k - j$ Kandidaten haben kann.

6.3.5 Reclustering eines Teilbaumes

In diesem Abschnitt wird das Entfernen valider Cluster aus der Clusterhierarchie beschrieben. Damit ist zum einen das Entfernen valider innerer Cluster gemeint, als auch das Entfernen valider Blätter. Diese beiden Vorgänge sollen zusammen erläutert werden, da der letztere ein Spezialfall des ersteren ist. Im Weiteren soll daher nur noch vom Entfernen eines (gewählten) Clusters die Rede sein. Auch ist mit dem Wort *Entfernen* der Vorgang nicht vollständig beschrieben, da ein Cluster nicht nur entfernt wird, wenn ohne ihn die globalen Kosten seines Eltern-Clusters geringer sind als mit ihm, sondern auch wenn eine Teilmenge aus der Menge der Cluster-Kandidaten existiert, durch die der Cluster ersetzt werden kann und dies zu einem Sinken der globalen Kosten führt. Des Weiteren ist es auch möglich, dass mehr als ein Cluster entfernt bzw. ersetzt wird. Daher ist der Ausdruck *partielles Reclustering* besser geeignet, um den Vorgang zu beschreiben.

Das partielle Reclustering wird ab einem SCP von drei ausgeführt. Prinzipiell wird dabei über die Menge der gewählten Cluster eines Eltern-Cluster iteriert und für jeden dieser Kind-Cluster überprüft, ob:

1. Der SCP kleiner oder gleich der gesetzten Strukturänderungsstufe ist.

2. Die globalen Kosten des Eltern-Clusters durch das Ersetzen des Kind-Cluster durch eine Teilmenge der Kandidaten gesenkt werden können.

Sind beide Bedingungen erfüllt, so wird die Ersetzung ausgeführt, wobei es auch sein kann, dass der Kind-Cluster durch die leere Menge ersetzt wird, d. h. das schon das reine Entfernen des Kind-Cluster zu einer Verbesserung führt. Die Abbildung 6.8 zeigt die Prozedur **doReclustering**, welche das partielle Reclustering ausführt.

Die Prozedur wird mit der Menge der gewählten Cluster für den Parameter *ChosenClusters* gestartet. Für jeden gewählten Cluster wird als erstes überprüft, ob sein SCP-Wert über dem gesetzten SCP liegt (Zeile 2). Ist dies der Fall, so wird mit dem nächsten

```
Candidates          := All clustercandidates for the actual parent cluster
BestChosenClusters  := Allready chosen clusters
SetSCP              := SCP set for partiall reclustering

doReclustering(ChosenClusters)
1    For each Cluster ∈ ChosenClusters
2       If Cluster.SCP ≤ SetSCP Then
3          If GCost(ChosenClusters - Cluster) < GCost(ChosenClusters) Then
4             doReclustering(ChosenClusters / Cluster)
5          Else
6             NewClusters := doBaBClustering(0, (ChosenClusters - Cluster))
7             If GCost(NewClusters) < GCost(ChosenClusters) Then
8                doReclustering(NewClusters)
9             End If
10         End If
11      End If
12   End For
13   If GCost(ChosenClusters) < GCost(BestChosenClusters) Then
14      BestChosenClusters := ChosenClusters
15   End If
```

Abbildung 6.8: Partielles Reclustering

Cluster fortgefahren. Wenn nicht, wird für den Cluster als nächstes überprüft, ob durch sein Entfernen aus der Überdeckung des Eltern-Clusters die Kosten direkt gesenkt werden können (Zeile 3). Ist dies der Fall, wird mit der neuen Menge gewählter Cluster die Prozedur rekursiv wieder aufgerufen. Wenn nicht, wird erst ein Reclustering des Misc-Clusters durchgeführt (Zeile 6). Für dieses Reclustering wird wieder die in Abschnitt 5.3.2 (siehe Abbildung 5.8) erläuterte Prozedur **doBaBClustering** benutzt. Ob hierfür die Menge der Cluster-Kandidaten den zur Überprüfung entfernten Cluster enthält oder nicht, ist ohne Belang. Sollte er wieder zur Menge der gewählten Cluster hinzugefügt werden, so sind die globalen Kosten mindestens ebenso hoch, wie vor dem Reclustering, also würde dann nicht die Bedingung in Zeile sieben erfüllt und somit auch die Rekursion nicht fortgeführt. Die Prozedur **doBaBClustering** gibt hier die Menge der Cluster wieder, die die Teilmenge der bisher gewählten Cluster ohne den zu prüfenden enthält, erweitert um eventuell weitere Cluster-Kandidaten. Sind die Kosten dieser so ermittelten neuen Menge (*NewClusters*) geringer als die der bisher gewählten Cluster (mit dem zu prüfenden Cluster) wird ebenfalls die Prozedur rekursiv wieder aufgerufen. Wenn durch Austauschen von Clustern die globalen Kosten nicht weiter gesenkt werden können, wenn also auf einer Rekurssionstufe über alle gewählten Cluster iteriert ist, wird überprüft ob die aktuelle Auswahl besser ist als die bisher Beste (Zeile 13).

Die in Prozedur **doReclustering** gezeigte Vorgehensweise beim partiellen Reclustering hat spezielle Auswirkungen. Das Ergebnis muss nicht optimal sein. Wenn statt der Betrachtung einzelner gewählter Cluster die Menge der nicht zum Austausch zugelassenen Cluster[13] als gewählt gesetzt wird und mit dieser Voraussetzung ein Reclustering mittels der Prozedur **doReclustering** erfolgt, so führt dies zu den minimalen globalen Kosten, die unter der Vorgabe der SCP erreicht werden können. Allerdings führt diese

[13] Alle Cluster deren SCP größer, als der gesetzte ist.

Art des Reclusterings bei hohen SCP auch zu hohen Laufzeiten. Denn ist das SCP hoch genug gewählt, wird ein komplettes Reclustering betrieben mit der dementsprechenden oberen Schranke für die Laufzeiten von $O(2^n)$(siehe Abschnitt 5.3.2). Auch folgt die inkrementierte Clustering-Hierarchie nicht mehr aus der initialen, sondern ist de facto ein komplett neues Clustering, was nicht Ziel einer Inkrementierung sein soll.

Die oben beschriebene Prozedur führt unter Umständen also zu einem suboptimalen Ergebnis des partiellen Reclustering. Dennoch berücksichtigt sie dem Umstand, dass die Reihenfolge der Überprüfung und Entfernung der gewählten Cluster Einfluss auf das Ergebnis haben kann. So kann das Ersetzen eines Clusters dazu führen, dass keine weiteren Ersetzungen mehr vorgenommen werden, da sie nicht mehr zu niedrigeren Kosten führen. Würde aber anstelle dieses Clusters ein oder mehrere andere Cluster ausgetauscht, welche aber in der Reihenfolge nach diesem kommen (also später überprüft werden), könnten man eventuell geringere Kosten erreichen als bei der ersten Ersetzung. Also lohnt es sich, alle austauschbaren Cluster zu überprüfen und mit den neu entstandenen Mengen gewählter Cluster rekursiv in gleicher Weise zu verfahren.

7 Evaluation

Nachdem in den vergangenen beiden Kapiteln das Verfahren für das inkrementelle Clustering von Dokumenten einer Newsgroup vorgestellt wurde, wird es in diesem Kapitel bewertet. Für diesen Zweck wurden vier verschiedene Newsgroup ausgewählt und Beiträge aus diesen Newsgroups für das Clustering geladen. Die Beiträge sowie ihre (Thread-)Struktur wurden gespeichert und danach in der in Kapitel 4 beschriebenen Form weiterverarbeitet. Die Bewertung des in dieser Diplomarbeit vorgestellten Clusterverfahrens wird mittels der so entstandenen Testkollektionen durchgeführt. Dabei werden zum einen die für das *initiale Clustering* (siehe Kapitel 5) vorgestellten Algorithmen zur Erstellung der Cluster-Hierarchie in ihrer Performance und der Güte der durch sie erstellten Cluster-Hierarchien verglichen und zum anderen wird das Verfahren mit dem Verfahren verglichen welches in Abschnitt 2.3.4.2 beschrieben wurde.

Da der Schwerpunkt dieser Diplomarbeit auf der inkrementellen Erweiterung und ihrer Gütesensitivität gegenüber der Art der Strukturänderungen liegt, werden nur Bewertungen anhand des in Abschnitt 5.2 beschriebenen Kostenmaßes durchgeführt.

Im Nachfolgenden werden die vier Testkollektionen vorgestellt, auf denen die Evaluation durchgeführt wurde. Im Anschluss daran werden die Ergebnisse vorgestellt, die mit dem Verfahren für das initiale Clustering erreicht wurden. Im Abschnitt 7.2 werden dann die Ergebnisse der Inkrementierung beschrieben.

7.1 Newsgroups

Die Evaluation wurde auf vier Newsgroups durchgeführt. Die erste ist eine Newsgroup, welche die Microsoft Applikation *Access* zum Thema hat. In ihre werden Fragen zur Erstellung von Tabellen, Formularen, Berichten, VBA-Programmierung und Ähnlichem beantwortet. Die Zweite Newsgroup beschäftigt sich mit politischen Themen. In ihr werden vor allem politische Themen diskutiert, die in den Vereinigten Staaten von Amerika von Relevanz sind, wie zum Beispiel die Präsidentenwahlen oder der Irak-Krieg. Die dritte Newsgroup befasst sich wieder mit einem technischen Thema: die Hardwarekonfiguration von Computern u.Ä. Als letztes wurde das Clusteringverfahren auf einer Newsgroup getestet, die sich mit einer breiten Palette von Themen beschäftigt. Als herausragende Gebiete sind vor allem Philosophie und Religion zu nennen. Aber auch ökonomische Sachverhalte werden in ihr erörtert.

Name	Server	Newsgroup
Access	msnews.mircosoft.com	microsoft.public.access
Politik	news.syr.edu	alt.politics
Hardware	news.syr.edu	alt.philosophie
Philosophie	textnews.news.cambrium.nl	alt.comp.hardware

Tabelle 7.1: Newsgroups

Die Tabelle 7.1 listet die vier Newsgroups mit ihren im Weiteren verwendeten Namen, ihren jeweiligen Server- und den konkreten Gruppennamen auf. Mittels des Server- und Gruppennamens einer Newsgroup lassen sich die Beiträge einer Newsgroup mithilfe eines Newsreaders (beispielsweise Outlook Express oder Netscape Messenger) abrufen und betrachten. Allen Newsgruppen ist gemeinsam, dass sie

1. über das NNTP-Protokol erreichbar sind,
2. keiner Registrierung bedürfen und
3. dass sie in englischer Sprache geführt werden.

Aus allen Newsgroups wurden in einem Zeitraum von knapp drei Monaten sukzessive Beiträge geladen. Da die Beiträge in unterschiedlichen Newsgroups unterschiedlich lange gespeichert werden, sind die jeweils ersten Beiträge, die in die Testkollektionen geladen wurden, zu sehr verschiedenen Zeiten verfasst worden. Die folgende Auflistung zeigt für die jeweilige Newsgroup das Datum des ersten und des letzten geladenen Beitrages und die Anzahl der Tage zwischen beiden Terminen.

	Erster Beitrag	Letzter Beitrag	Tage
Access	27.01.05	21.09.05	237
Politik	06.06.05	24.08.05	79
Hardware	17.04.05	08.08.05	113
Philosophie	02.05.05	16.09.05	137

Tabelle 7.2: Zeitraum der initialen Erfassung

Nach Abschluss der initialen Erfassung der Beiträge erfolgte die in Abschnitt 4 beschriebene Vorverarbeitung:

1. Entfernen von Beitragswiederholungen.
2. Entfernen von Standardtexten. Hierbei berücksichtigt:
   - Webadressen,
   - E-Mailadressen,
   - automatisch generierte Einschübe wie „Tom Mayer wrote in message“ oder „26 + 6 = 1 It's Irish Math“.
3. Zusammenfassen der Beiträge zu Dokumenten.
4. Erstellen der Termvektoren.
5. Auswahl der relevanten Terme.

In der Tabelle 7.3 wird nun die statistische Zusammensetzung der einzelnen Testkollektionen in Bezug auf die Beiträge und Dokumente gezeigt.

	Access	Politik	Hardware	Philosophie
Beiträge	33573	**40313**	6874	19305
Beiträge pro Tag	141,66	**510,29**	60,83	140,91
Dokumente	**10522**	9534	1496	2584
Beiträge pro Dokument	3,19	4,23	4,59	**7,47**
Einfache Dokumente	1267(12%)	**3019(32%)**	220(15%)	320(12%)
Dokumente $>$ 15 Terme	**10056**	8676	1411	2506

Tabelle 7.3: Dokumentbezogene Daten der Testkollektionen

In jeder Zeile der Tabelle ist eine statistische Kennzahl für jede Newsgroup angegeben. Die vier Newsgroups weisen in den einzelnen Kennzahlen zum Teil sehr unterschiedliche Ausprägungen auf. Manche dieser Kennzahlen sind in ihrer Bedeutung nicht sofort ersichtlich. Diese sollen im Folgenden erläutert werden. Auch wird ihre Bedeutung in einigen Fällen interpretiert.

Die *Beiträge pro Tag* sind durch die Division der Gesamtanzahl an Beiträgen mit der Dauer der initialen Erfassung (siehe Tabelle 7.2) entstanden. An dieser Kennzahl ist die *Frequentiertheit* der einzelnen Newsgroups sehr gut abzulesen. Wie deutlich zu sehen ist, werden in der Politik-Newsgroup die meisten neuen Beiträge je Tag (über 510) verfasst. Bei den *Beiträgen pro Dokument* ist die Philosophie-Newsgroup führend. Die Zahl von über sieben Beiträgen pro Dokument können damit erklärt werden, dass sie ein Forum für Diskussionen ist, im Gegensatz zu den technisch orientierten Newsgroups für Access und Hardware, welche eher auf Problemlösung ausgelegt sind. Allerdings ist es in diesem Zusammenhang sehr verwunderlich, dass die Politik-Newsgroup nicht so viele Beiträge pro Dokument aufweist, da diese auch als Diskussionsforum ausgelegt ist. Eine Erklärung dafür, dass das Politik-Forum[1] einen so geringen Beitragsschnitt pro Dokument hat, ist die wiederum sehr hohe Anzahl *einfacher Dokumente*. Als *einfach* wird ein Dokument bezeichnet, welches nur einen Beitrag enthält: den Eröffnungsbeitrag. Sehr viele Eröffnungsbeiträge bleiben im Politik-Forum also unbeantwortet. Diese überdurchschnittlich hohe Anzahl einfacher Dokumente drückt natürlich den Durchschnitt der Beiträge je Dokument. Im Weiteren werden die Foren, wie oben angedeutet, nach den technikorientierten (Access und Hardware) und den diskussionsorientierten (Politik und Philosophie) Foren unterschieden. Die erstgenannten sind aufgrund ihrer Domäne eher an praktischen Problemlösungen interessiert, währenddessen die letzteren dem Meinungsaustausch dienen.

Die Tabelle 7.3 zeigt Kennzahlen, die sich auf das Verhältnis der Terme zu den Dokumenten bezieht. Diese Zusammenhänge zwischen Termen und Dokumenten sind besonders für den häufigkeitsbasierten Ansatz wichtig, so wie er in dieser Diplomarbeit gewählt wurde.

[1] Wie bereits in der Einleitung erwähnt, sind Werbforen und Newsgroups inhaltlich und strukturell gleich. Daher wird der Begriff Forum auch in dem oben stehenden Zusammenhang verwendet.

	Access	Politik	Hardware	Philosophie
Dokument-länge (av)	122,66	348,39	200,17	**551,58**
Dokument-Term-Länge (av)	59,68(49%)	190,92(55%)	111,16(56%)	**234,61(43%)**
Terme	34644	**55567**	15951	40425
Terme > 1 %	903(2,61%)	3178(5,72%)	1765(**11,07%**)	**3928**(9,72%)
Terme > 5 %	259(0,75%)	952(1,71%)	542(**3,40%**)	**1094**(2,71%)
Terme > 10 %	113(0,33%)	**367**(0,66%)	209(**1,31%**)	452(1,12%)

Tabelle 7.4: Termbezogene Daten der Testkollektionen

Wichtig für die Interpretation dieser Kennzahlen ist die Tatsache, dass für ihre Berechnung nur Dokumente verwendet wurden, die über mehr als 15 Terme verfügen. Hierbei sind nur Terme gemeint, die nicht durch die Stoppwortentfernung ausgeklammert wurden.

Die *Dokumentlänge (av)* gibt die durchschnittliche Gesamtanzahl an Termen in einem Dokument an. Die *Dokument-Term-Länge (av)* wiederum zählt die durchschnittliche Anzahl an verschiedenen Termen in einem Dokument. Hinter der absoluten Dokument-Term-Länge wird das Verhältnis zur durchschnittlichen Dokumentlänge angegeben. Diese Verhältniszahl zeigt also den Grad der Mehrfachverwendung von Termen in den Dokumenten. Es zeigt sich, dass die Newsgroups sich in den absoluten Längen stark unterscheiden. Die Diskussionsforen, *Politik* und *Philosophie*, verfügen über signifikant mehr Terme in den Dokumenten als die technisch orientierten Foren *Access* und *Hardware*. Bei den Mehrfachnennungen wiederum ist keine klare Unterscheidung zwischen den Technik- und den Diskussionsforen zu erkennen. In der Zeile *Terme* ist die jeweilige Gesamtanzahl an vorkommenden Termen angegeben. Diese werden aber künstlich aufgebläht von den Schreibfehlern der Beitragsautoren. Daher werden in den letzten drei Zeilen der Tabelle, die absolute Häufigkeiten der Terme, die in mehr als 1%, 5% oder 10% aller Dokumente mindestens einmal enthalten sind und ihre die prozentualen Häufigkeiten (in Klammern) zur Gesamtzahl der Terme aufgezählt. Hier fallen vor allem die *Hardware* und die *Philosophie* Newsgroup auf. Ihre hohen Anteile an häufigen Termen[2] im Verhältnis zu der Gesamtanzahl der Terme in diesen beiden Foren ist teilweise auf die Angabe von Webadressen und Ähnlichem zurückzuführen, die während der Vorverarbeitung nicht gefiltert wurden.

7.2 Initiales Clustering

In diesem Abschnitt werden die Ergebnisse des initialen Clusterings beschrieben. Dafür wurden auf den Testkollektionen, die im vorhergehenden Abschnitt beschrieben sind, verschiedene Auswertungen vorgenommen. Als erstes werden im Abschnitt 7.2.1 die Ergebnisse des Clusterings der Testkollektionen erläutert. Dieses Clustering wurde mit dem in dieser Diplomarbeit entwickelten Verfahren ausgeführt. Der verwendete *branch-and-*

[2]Würde jeweils ein entsprechender MinSupport (1%, 5% oder 10%) vergeben, wären dies die häufigen Terme für das Clusterverfahren (ohne durchgeführte Termselektion).

bound Algorithmus findet das globale Optimum (siehe Abschnitt 5.3.2.1). Diese Ergebnisse sind zugleich die Vergleichsbasis für die dann folgenden Auswertungen.

7.2.1 Referenz-Clustering

Das im Folgenden beschriebene Clustering wurde für die vier Testkollektionen jeweils auf der vollständigen Dokumentenmenge ausgeführt. Verwendet wurde der in Abschnitt 5.3.2.1 beschriebene *branch-and-bound* Algorithmus für das Clustering der Kandidaten. Der MinSupport wurde für die einzelnen Newsgroups wie in Tabelle 7.5 gesetzt.

	MinSupport	minimal notwendige Dokumente
Access	6%	604
Politik	8%	495
Hardware	5%	71
Philosophie	8%	201

Tabelle 7.5: MinSupport für die einzelnen Newsgroups

Des Weiteren wurde für das Clustering die Anzahl der Dokumente in den Clusterkandidaten als Gewichtung für die Berechnung der globalen Kosten (siehe Abschnitt 5.2.4) verwendet. Diese Gewichtung soll im Folgenden als *Standardgewichtung* bezeichnet werden. Das so erzielte Clustering dient als Referenz für die verschiedenen Evaluationsprozeduren in den folgenden Abschnitten und wird im Weiteren als *Referenz-Clustering* bezeichnet.

Die Tabellen 7.6, 7.7, 7.8 und 7.9 zeigen die Strukturstatistiken der Cluster-Hierarchien für die einzelnen Newsgroups. Es werden verschiedene Kennzahlen angegeben, welche die Hierachien beschreiben. Dabei wird nach Ebene (Level) der Hierarchie unterschieden. Die Ebene 0 zeigt die Kennzahlen für die Wurzel des Clusterings, die Ebene 1 die Kennzahlen für die Clusterkandidaten, die aus den 1-fts hervorgegangen sind, usw.

Die Bedeutung der einzelnen Kennzahlen zeigt die folgende Tabelle.

1.	*FTS*	Gesamtanzahl der Clusterkandidaten bzw. fts
2.	*Cluster*	Gesamtanzahl der gewählten Cluster
3.	*mit Kand.*	Anzahl der Cluster mit Kandidaten
4.	*GC (av)*	Durchschnittliche globale Kosten
5.	*LC (av)*	Durchschnittliche lokale Kosten
6.	*OV (av)*	Durchschnittliche Kosten für Überlappungen
7.	*UP (av)*	Durchschnittliche Kosten für ungenutzte Möglichkeiten
8.	*Kinder (av)*	Durchschnittliche Anzahl der Kinder
9.	*Kand. (av)*	Durchschnittliche Anzahl der Kandidaten, die nicht gewählt wurden
10.	*Dok. (av)*	Durchschnittliche Anzahl der Dokumente
11.	*Dok. ohne Kand. (av)*	Durchschnittliche Anzahl der Dokumente ohne Clusterkandidaten (absolut / Verhältnis zur Dokumentanzahl)
12.	*Dok. in Misc (av)*	Durchschnittliche Anzahl der Dokumente in Misc-Clustern (absolut / Verhältnis zu Dokumentanzahl)

Bei den Kennzahlen, welche mit *(av)* gekennzeichnet sind, handelt es sich um Durchschnittswerte über die Cluster einer Ebene bzw. über alle Cluster der Hierarchie (Spalte *Gesamt*). Für die Zeilen 4, 5, 6, 7, 8, 9, 11 und 12 sind die Durchschnittswerte über die Cluster gebildet, die Kandidaten haben (siehe Zeile 3). Blatt-Cluster werden also nicht berücksichtigt. Diese haben keine globalen Kosten, keine Kind-Cluster, usw., daher würden durch sie die Werte verfälscht werden.

Weiterhin ist anzumerken, dass es sich bei den Kennzahlen für die Wurzel-Ebene (Level 0) um absolute Werte handelt, da es nur einen Cluster (die Wurzel) auf dieser Ebene gibt. Die globalen Kosten der Wurzel geben auch die Güte der jeweiligen Cluster-Hierarchie an und werden für die Evaluationen in den folgenden Abschnitten als Referenzwerte verwendet.

	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5	Gesamt
1.FTS	-	40	268	384	174	15	881
2.Cluster	1	13	24	21	12	1	72
3.mit Kand.	1	12	15	11	1	0	40
4.GC (av)	0,8211	0,3883	0,2629	0,1320	0	0	0,2789
5.LC (av)	1,3700	0,5574	0,3655	0,1320	0	0	0,3845
6.OV (av)	1299	80,3	23,6	0	0	0	67,1
7.UP (av)	634	61,9	34,1	14,5	0	0	52,5
8.Kinder (av)	13	2,0	1,3	1,0	0	0	1,7
9.Kand. (av)	27	5,9	3,3	1,3	0	0	4,1
10.Dok. (av)	1411	181,3	113,1	94,2	84,2	81,0	132,7
11.Dok. ohne	112	12,3	6,0	8,2	-	-	11,3
Kand (av)	7,94%	6,76%	5,30%	8,68%	-	-	8,48%
12.Dok. in	353	40,7	24,9	21,0	-	-	37,1
Misc (av)	25,02%	22,43%	22,04%	22,28%	-	-	27,94%

Tabelle 7.6: Referenz-Clustering: Access Newsgroup

7.2.2 Approximationsalgorithmen und Performance

Für die Referenz-Clusterings im vorangegangenen Abschnitt wurde der *branch-and-bound* Algorithmus verwendet. Dieser findet für jeden Cluster die (kosten-)optimale Überdeckung. Das Clustering für jeden einzelnen Cluster der Hierarchie hat also minimale *globale Kosten*. Daraus und aus der rekursiven Berechnung der globalen Kosten (siehe 5.2.4) folgt, dass die Kosten für das gesamte Clustering mit dem *branch-and-bound* Algorithmus minimal sind. Der Wurzel-Cluster hat minimale globale Kosten und somit hat das Clustering die optimale Güte (in Bezug auf das verwendete Kostenmaß).

Die Ergebnisse der Referenz-Clusterings werden in diesem Abschnitt mit den Ergebnissen der Clusterings verschiedener Approximationen verglichen. Dafür wurden Clusterings mit dem *n-Greedy* Algorithmus (siehe Abschnitt *n-Greedy*) und der *rundenbasierten* Variante des branch-and-bound Algorithmus (siehe Abschnitt 5.3.2.2) durchgeführt. Es wurden vier unterschiedliche Einstellungen für den Parameter n von *n-Greedy* verwendet (4, 8, 12 und 16). Das Clustering mit der rundenbasierten Modifikation des *branch-and-bound* Algorithmus wurde mit zwei verschiedenen Rundenzahlen (10 Millionen und 100 Millionen) durchgeführt.

Beide Approximationsverfahren sind Varianten des *branch-and-bound* Algorithmus. Sie beschränken jedoch die Anzahl der zu überprüfenden Kandidatenteilmengen weiter, die insgesamt für die Überdeckung eines Clusters (oder besser: seiner Dokumentenmenge) in Frage kommen[3]. Die *n-Greedy* Variante wählt nur die n besten Kandidaten um die Kandidatenteilmengen zu erweitern. Also wird in jedem Knoten des Entscheidungsbaumes (siehe Abbildung 5.9 in Abschnitt 5.3.2.1) zu jeweils n weiteren Kind-Knoten verzweigt (auch wenn evtl. mehr Möglichkeiten vorhanden wären). Die *rundenbasierten* Variante des branch-and-bound Algorithmus geht genauso wie der branch-and-bound Algorithmus

[3] Auch der branch-and-bound Algorithmus beschränkt den Raum der möglichen Lösungen. Jedoch ist im *worst-case* die Komplexität $0(2^n)$.

	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8	Gesamt
1.FTS	-	46	683	2800	4539	2822	496	19	1	7877
2.Cluster	1	8	33	47	28	7	3	1	0	86
3.mit Kand.	1	8	33	24	6	1	1	0	0	32
4.GC (av)	1,1143	0,6821	0,3244	0,2101	0,1002	0	0	0	0	0,3127
5.LC (av)	1,7841	1,4946	0,5019	0,2471	0,1079	0	0	0	0	0,5024
6.OV (av)	9836	2590,0	298,4	31,3	0	0	0	0	0	556,1
7.UP (av)	5643	1098,4	346,7	207,4	98,8	0	0	0	0	428,3
8.Kinder (av)	8	4,1	1,4	1,0	1,0	0	0	0	0	1,6
9.Kand. (av)	38	19,1	5,7	2,8	1,5	0	0	0	0	6,2
10.Dok. (av)	8676	2121,8	1075,2	851,7	778,2	779,3	744,7	726,0	0	1026,3
11.Dok. ohne	208	27,9	29,6	25,6	20,7	7,0	15,0	-	-	29,3
Kand (av)	2,40%	1,31%	2,75%	3,01%	2,66%	0,90%	2,01%	-	-	2,86%
12.Dok. in	1538	276,6	160,6	134,3	95,2	70,0	70,0	-	-	176,0
Misc (av)	17,73%	13,04%	14,93%	15,77%	12,23%	8,98%	9,40%	-	-	17,15%

Tabelle 7.7: Referenz-Clustering: Politik Newsgroup

	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5	Gesamt
1.FTS	-	40	268	384	174	15	881
2.Cluster	1	13	24	21	12	1	72
3.mit Kand.	1	12	15	11	1	0	40
4.GC (av)	0,8211	0,3883	0,2629	0,1320	0	0	0,2789
5.LC (av)	1,3700	0,5574	0,3655	0,1320	0	0	0,3845
6.OV (av)	1299	80,3	23,6	0	0	0	67,1
7.UP (av)	634	61,9	34,1	14,5	0	0	52,5
8.Kinder (av)	13	2,0	1,3	1,0	0	0	1,7
9.Kand. (av)	27	5,9	3,3	1,3	0	0	4,1
10.Dok. (av)	1411	181,3	113,1	94,2	84,2	81,0	132,7
11.Dok. ohne	112	12,3	6,0	8,2	-	-	11,3
Kand (av)	7,94%	6,76%	5,30%	8,68%	-	-	8,48%
12.Dok. in	353	40,7	24,9	21,0	-	-	37,1
Misc (av)	25,02%	22,43%	22,04%	22,28%	-	-	27,94%

Tabelle 7.8: Referenz-Clustering: Hardware Newsgroup

	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Gesamt
1.FTS	-	46	712	2400	2109	604	35	5860
2.Cluster	1	11	31	35	9	1	0	87
3.mit Kand.	1	11	28	9	1	0	0	49
4.GC (av)	1,1140	0,6534	0,2315	0,2087	0,1162	0	0	0,3355
5.LC (av)	1,9844	1,3147	0,3280	0,2331	0,1162	0	0	0,5569
6.OV (av)	3167	395,7	45,6	0	0	0	0	175,9
7.UP (av)	1806	218,5	60,3	61,0	28,0	0	0	129,5
8.Kinder (av)	11	2,8	1,2	1,0	1,0	0	0	1,7
9.Kand. (av)	35	16,9	3,8	2,8	1,0	0	0	7,1
10.Dok. (av)	2506	445,5	271,8	227,9	216,0	204,0	0	294,9
11.Dok. ohne	172	13,7	13,0	7,0	9,0	-	-	15,2
Kand (av)	6,86%	3,08%	4,78%	3,07%	4,17%	-	-	5,15%
12.Dok. in	772	75,4	43,1	36,8	37,0	-	-	63,5
Misc (av)	30,81%	16,91%	15,85%	16,14%	17,13%	-	-	21,53%

Tabelle 7.9: Referenz-Clustering: Philosophie Newsgroup

selbst vor, nur beendet sie nach der angegebenen Anzahl von Runden den Algorithmus und nimmt das bis zu diesem Zeitpunkt beste Ergebnis. Eine *Runde* ist dabei als ein Aufruf der rekursiven Methode definiert. Es wird also pro überprüfter Kandidatenteilmenge der Rundenzähler um eins erhöht.

Bei allen Approximationen wurden die gleichen Grundeinstellungen (Gewichtungsfaktor und MinSupport) wie beim Referenz-Clustering verwendet.

	Globale Kosten		Runden	Zeit (min)
Referenz-Clustering	0,99186426	100,000%	3.872.384.404	2011
4-Greedy	1,02947656	96,346%	15.684	15
8-Greedy	1,02362303	96,897%	465.464	15
12-Greedy	1,0188112	97,355%	2.934.441	18
16-Greedy	1,00950091	98,253%	20.479.028	23
Runden: 10 mio.	1,01484379	97,736%	67.497.678	35
Runden: 100 mio.	1,00102419	99,085%	476.577.265	172

Tabelle 7.10: Access Newsgroup

	Globale Kosten		Runden	Zeit (min)
Referenz-Clustering	1,114279323	100,000%	8.023.268.674	3570
4-Greedy	1,132917364	98,355%	1.656.205	291
8-Greedy	1,129822588	98,624%	8.849.384	324
12-Greedy	1,127125236	98,860%	60.762.700	354
16-Greedy	1,12585151	98,972%	80.250.096	360
Runden: 10 mio.	1,116204115	99,828%	296.609.673	370
Runden: 100 mio.	1,114279323	100,000%	1.187.576.524	643

Tabelle 7.11: Politik Newsgroup

	Globale Kosten		Runden	Zeit (min)
Referenz-Clustering	0,82106747	100,000%	6.288.649.085	408
4-Greedy	0,89335048	91,909%	16.367	3
8-Greedy	0,86683083	94,721%	1.780.373	4
12-Greedy	0,8630482	95,136%	14.100.659	6
16-Greedy	0,84690829	96,949%	55.819.659	8
Runden: 10 mio.	0,87205064	94,154%	46.016.932	5
Runden: 100 mio.	0,85105493	96,476%	262.065.454	18

Tabelle 7.12: Hardware Newsgroup

Die Tabellen 7.10, 7.11, 7.12 und 7.13 zeigen die Ergebnisse des Vergleichs zwischen den einzelnen Algorithmen. Für jedes Verfahren werden in den ersten beiden Spalten die globalen Kosten bzw. das Verhältnis der jeweiligen globalen Kosten zu den globalen

	Globale Kosten		Runden	Zeit (min)
Referenz-Clustering	1,1139836	100,000%	10.265.716.819	1238
4-Greedy	1,19573169	93,163%	44.985	57
8-Greedy	1,18162366	94,276%	1.372.603	61
12-Greedy	1,1617098	95,892%	2.577.661	67
16-Greedy	1,14593905	97,211%	42.074.120	75
Runden: 10 mio.	1,15379748	96,549%	232.676.314	72
Runden: 100 mio.	1,13547464	98,107%	975.352.567	116

Tabelle 7.13: Philosophie Newsgroup

Kosten des Referenz-Clusterings gezeigt (hier sei noch einmal angemerkt, dass die Güte einer Cluster-Hierarchie durch die globalen Kosten im Wurzel-Cluster angegeben ist). In der Spalte *Runden* wird die Anzahl der Runden gezeigt, die das gesamte Clustering bei dem jeweiligen Algorithmus benötigt hat. Für jeden Clusterkandidaten (außer den Kandidaten, welche Blätter sind) wird der *branch-and-bound* bzw. der *n-Greedy* oder der *rundenbasierte* Algorithmus aufgerufen. Und für jede Überprüfung einer Teilmenge von Clusterkandidaten wird jeweils eine Runde gezählt. Bei der rundenbasierten Variante wird für jeden Kandidaten, der geclustert wird, die maximale Rundenanzahl (die er verbrauchen darf) auf den angegebenen Wert (10 Millionen oder 100 Millionen) für die Runden gesetzt. Bei Kandidaten, welche selbst nur über wenige Clusterkandidaten verfügen, wird diese Rundenanzahl nicht erreicht. Für diese Kandidaten wird die optimale Überdeckung gefunden, da sich der rundenbasierte Algorithmus vom branch-and-bound nur durch den Abbruch der Suche *nach Erreichen* der gesetzten Rundenanzahl unterscheidet. Die in den einzelnen Kandidaten durchgeführte Rundenanzahl wird über alle Kandidaten aufaddiert und ist in der Spalte *Runden* zu sehen. Die Spalte *Zeit (min.)* zeigt die real verbrauchte Zeit in Minuten. Diese Angabe besitzt natürlich nur eine relative Aussagekraft, da hier der Rechnertyp berücksichtigt werden muss. Für Tests wurden ausnahmslos Computer mit folgender Konfiguration verwendet:

- Prozessor : AMD Athlon XP 2600+ (1,91 GHz)
- Hauptspeicher : 1 GB RAM (266 ms)

Die Ergebnisse zeigen deutlich, dass die Güte der Clusterings, die mit den Approximationsalgorithmen durchgeführt wurden je nach Newsgroup sehr unterschiedlich sind. In der *Politik* Newsgroup beispielsweise wird mit dem rundenbasierten Algorithmus bei 10 Millionen Runden bereits ein sehr gutes, bei 100 Millionen Runden sogar das optimale Ergebnis erzielt. Bei der *Hardware* Newsgroup wiederum sind die Ergebnisse der rundenbasierten Approximationsalgorithmen eher schlecht. Ein Grund kann aus den in 7.1 gezeigten Statistiken der Term-Dokument-Verteilungen kaum hergeleitet werden. Die Ergebnisse der Clusterings, die mit den n-Greedy Algorithmen erstellt wurden, sind ähnlich den Ergebnissen der rundenbasierten Approximationsalgorithmen.

7.2.3 Vergleich mit anderem Verfahren

Um das in dieser Diplomarbeit entwickelte Verfahren mit dem Verfahren aus [FWE03] zu vergleichen, wurde dieses Verfahren auf das *Access* Forum angewandt. Das Verfahren entspricht dem in Abschnitt 2.3.4.2 erläuterten. Die Abbildung 7.14 zeigt die Ergebnisse. Als Kostenmaß wurde das gleiche Maß, wie in Abschnitt 5.2 beschrieben, verwendet, also das gleiche wie auch für das Verfahren dieser Diplomarbeit.

Der Effekt der *buschigen* Struktur wird durch die angegebenen Kennzahlen belegt. Auf jeder Ebene des Clustering wurden fast alle Clusterkandidaten ausgewählt. Dies ist der Tatsache zuzuschreiben, dass alle Dokumente bei diesem Verfahren abgedeckt werden müssen. Um dies zu gewährleisten, müssen fast immer alle Kandidaten ausgewählt werden. Das hat natürlich auch zur Folge, dass die globalen Kosten hier wesentlich schlechter sind. Grund hierfür sind die hohen Kosten für die Überlappungen.

Dieser Nachteil des Verfahren aus [FWE03] war der Grund für die Veränderungen des Kostenmaßes(siehe Abschnitt 5.2) für das Clustering. Wie in den Ergebnissen der Referenz-Clusterings zu sehen ist, ist die Struktur der Cluster-Hierarchien weitaus weniger breit. Das Ziel einer kompakteren und damit übersichtlicher Hierarchie wurde also erreicht.

	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
FTS	-	40	381	616	215	2
Cluster	1	39	356	609	215	2
GC (av)	3,5401	2,8584	2,3119	2,0042	0,4291	0
LC (av)	8,4333	5,8243	3,3970	2,0476	0,4291	0
mit Kand.	1	38,0	205	162	1	0
Kinder (av)	39	16,7	7,9	4,8	2,0	0
Kand. (av)	1	3,0	0,8	0,0	0	0
Dok. (av)	10056	2431,1	1079,0	807,5	698,1	625,5

Tabelle 7.14: Vergleichsverfahren: Access Newsgroup

7.3 Inkrementelles Clustering

In diesem Abschnitt werden die Ergebnisse der Inkrementierung vorgestellt. Hierbei wird vor allem die Betrachtung des Zielkonfliktes zwischen *Güte* und *Strukturerhaltung* der Cluster-Hierarchie in den Vordergrund gestellt. Wie verhält sich also die Güte der Cluster-Hierarchie bei Wahl von unterschiedlichen Strukturänderungsstufen (siehe Abschnitt 6.2)? Dabei wird auch berücksichtigt, wie sich das Hinzufügen unterschiedlich großer Dokumentmengen bei der Inkrementierung auswirkt. Die Ergebnisse werden im folgenden Abschnitt 7.3.1 beschrieben. Im Abschnitt 7.3.2 werden dann die Clusteringergebnisse für alternative Gewichtungen der globalen Kosten exemplarisch an der *Access* Newsgroup gezeigt.

7.3.1 Intervall Inkrementierung

Die Evaluation der Inkrementierung wird auf allen vier Referenz-Clusterings durchgeführt. Hierfür wurde ein weiteres Clustering durchgeführt, welches weniger Dokumente aus den Testkollektionen verwendet. Dieses Clustering wird im Weiteren als Start-Clustering bezeichnet. Ausgehend von diesem Start-Clustering werden verschiedene Inkrementierungen durchgeführt, die alle bei der Dokumentenmenge des jeweiligen Referenz-Clusterings enden. Insgesamt wird die Inkrementierung mit zwei Parametern durchgeführt. Ein Parameter ist die Strukturänderungsstufe und der andere das Intervall der neu hinzugenommenen Dokumente. Die Tabelle 7.15 zeigt die Konfiguration für die einzelnen Newsgroups.

Die Bedeutung der einzelnen Kennzahlen der Tabelle 7.15 werden in der folgende Auflistung erklärt.

- *Start*: Die Dokumentanzahl des Start-Clusterings.
- *Ende* : Die Dokumentanzahl des Referenz-Clusterings.
- *max. SCP-Stufe* : Gibt an, bis zu welcher Strukturänderungsstufe die Inkrementierungen durchgeführt werden mussten. Diese Kennzahl wurde jeweils durch den maximalen SCP-Wert in der Start-Cluster-Hierarchie festgelegt (zur näheren Erläuterung, siehe Beispiel 7.3.1).
- *Intervall 0-3* : Bezeichnet die Anzahl der geänderten und neuen Dokumente, die während einer *Iteration* zum Start-Clustering hinzugenommen wurden. Das heißt, es wurden für jede durchgeführte Iteration so viele Beiträge dem Start-Clustering hinzugefügt, bis die Anzahl an geänderten und neuen Dokumente erreicht wurde, die in der Zeile *Intervall* angegeben ist. Während einer Iteration mit dem Intervall 250 wurden also nicht 250 neue Dokumente hinzugefügt. Es wurden sukzessive Beiträge nachgeladen. Für jeden Beitrag wurde nachgesehen, ob er ein Kommentar für einen anderen Beitrag war oder ob er ein neues Dokument eröffnete. Im ersten Fall wurde der neue Beitrag dem entsprechenden Dokument zugefügt und dieses Dokument wurde als *geändert* gekennzeichnet. Im zweiten Fall wurde ein neues Dokument eröffnet und erhielt den Status *neu*. Ein Beitrag konnte also auch einem bereits geänderten oder einem neu erstellten Beitrag zugefügt worden sein. In diesem Fall stieg die Anzahl der geänderten bzw. neuen Dokumente nicht. Die Summe der Anzahl der neuen und der geänderten Dokumente wurde nach jedem Beitrag daraufhin überprüft, ob die für das Intervall notwendige Anzahl erreicht ist. Ein Sonderfall ist das *Intervall 0*. In diesem wurden alle Beiträge der Testkollektionen in einer Iteration nachgeladen.
- *Iterationen* : Gibt an, wie oft die Inkrementierung mit der angegebenen Intervallgröße durchgeführt werden musste, um die Dokumentanzahl des Referenz-Clusterings zu erreichen.

Beispiel 7.3.1. An dieser Stelle wird das Beispiel aus Abbildung 6.6 in Abschnitt 6.2.4 nochmals aufgegriffen, um den Begriff der *maximalen SCP-Stufe* zu erläutern.

Sei der dort gezeigte Cluster A der Kind-Cluster der Wurzel mit dem höchsten SCP-Wert in der gesamten Cluster-Hierarchie (sieben SCP). Eine Inkrementierung die mit

der Strukturänderungsstufe sieben ausgeführt wird, überprüft jeden Kind-Cluster des Wurzel-Clusters daraufhin, ob es eine Verbesserung der globalen Kosten darstellt, wenn er entfernt wird. Es werden also alle Strukturänderungsmöglichkeiten für den Wurzel-Cluster berücksichtigt. Da nun der Wurzel-Cluster aufgrund der Definition der SCP (ein Cluster erhält immer den höchsten SCP-Wert seiner Kind-Cluster plus einen oder zwei Punkte) den Cluster mit dem höchsten SCP-Wert enthält, folgt daraus, dass auch für alle anderen Cluster der Hierarchie die maximale Anzahl an möglichen Strukturänderungen berücksichtigt werden, woraus wiederum folgt, dass die maximale Verbesserung der Güte erreicht wird, die im Rahmen der so zugelassenen Strukturänderungen möglich ist.

	Access	Politik	Philosophie	Hardware
Start-Clustering	6554	5474	1786	935
Referenz-Clustering	10056	8676	2506	1411
Differenz	3502	3202	720	476
SCP-Stufe	8	11	9	8
Intervall 0	alle	alle	alle	alle
Iterationen	1	1	1	1
Intervall 1	1000	1000	300	200
Iterationen	4	4	3	3
Intervall 2	500	500	200	100
Iterationen	9	9	5	6
Intervall 3	250	250	100	50
Iterationen	18	21	13	15

Tabelle 7.15: Konfiguration der Inkrementierung

Die Diagramme in der Abbildung 7.1 zeigen die zusammengefassten Ergebnisse dieser Evaluationen. Auf der X-Achse werden die angewandten Strukturänderungsstufen abgetragen. Die Y-Achse zeigt das Verhältnis der durch die Inkrementierung erreichten Güte (globale Kosten) zu der Güte des Referenz-Clusterings[4]. Es wird also die prozentuale Annäherung der einzelnen Inkrementierungen an das Clustering mit der optimalen Güte angezeigt. Die Zahlenwerte für die Güte sind im Anhang in Abschnitt A.2 aufgeführt.

Wie in den Diagrammen zu sehen ist, hat das Setzen einer höheren Strukturänderungsstufe im Allgemeinen auch eine steigende Güte der Cluster-Hierarchie zur Folge. Für diese Beobachtung gibt in den beiden Foren *Access* und *Politik* jedoch auch Ausnahmen. Auf die Ursache für diese Phänomen soll hier näher eingegangen werden, da es eine problematische Eigenart der Inkrementierung aufzeigt.

Um den Vorgang zu verdeutlichen, werden die Daten der Iterationen der Politik und der Access Newsgroup im Intervall 2 in einer anderen Form präsentiert. Die Güte jeder Iteration für die Strukturänderungsstufen 1, 2, 3, 5, 7 und 11 (Politik) und 1, 2, 3, 4, 6 und 8 (Access) in den Abbildungen 7.2 und 7.3 mit der Güte der Strukturänderungsstufe 0 ins Verhältnis gesetzt. Die Strukturänderungsstufe 0 ist die triviale Inkrementierung der Cluster-Hierarchie (es werden lediglich die neuen und geänderten Dokumente ihren Clustern zugeordnet). Wenn eine höhere Strukturänderungsstufe vom Anwender gewählt

[4]Die Güte der jeweiligen Referenz-Clusterhierarchie ist in den Tabellen in Abschnitt 7.2.1 zu finden.

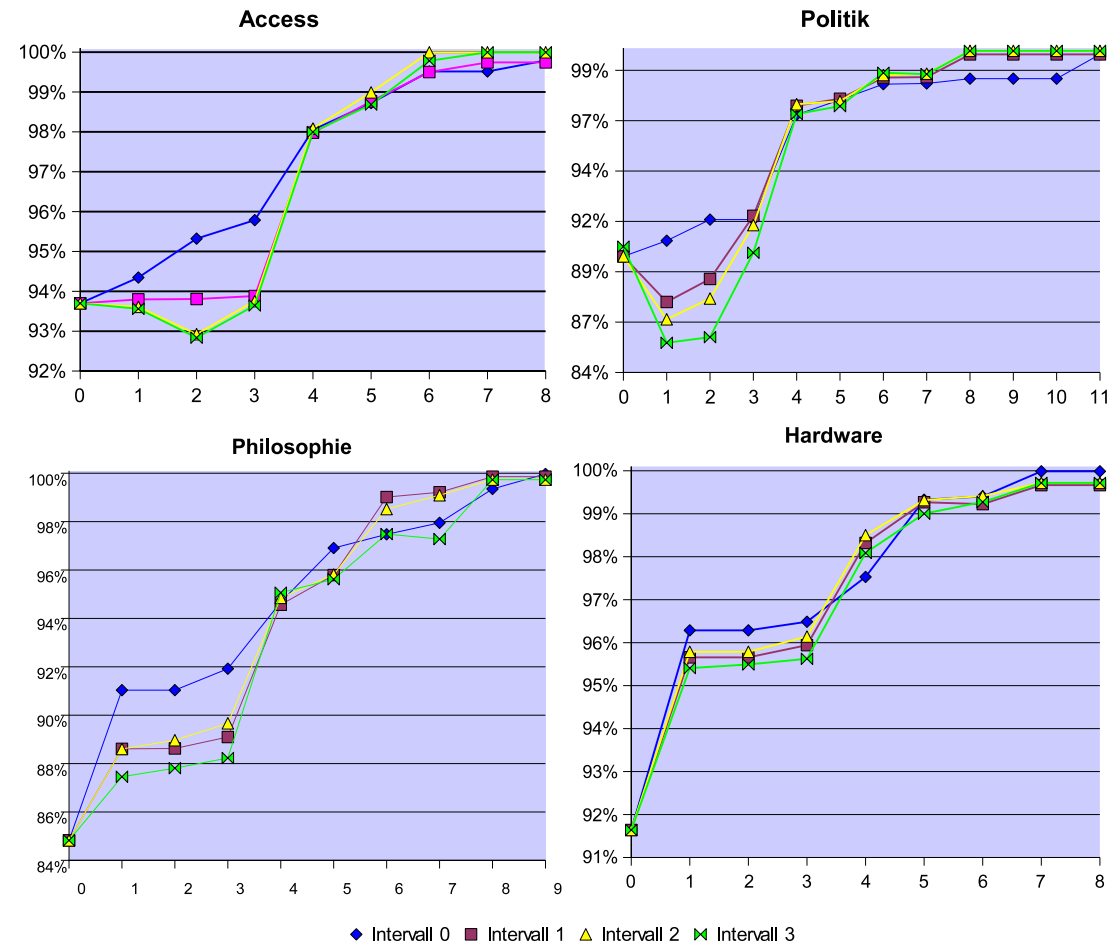


Abbildung 7.1: Intervall-Inkrementierung der Newsgroups

wird, ist seine Erwartungshaltung an die Inkrementierung, dass die Güte mindestens gleich gut wie bei der geringeren Stufe ist. Bei der trivialen Inkrementierung erwartet er also das schlechteste Ergebnis in Bezug auf die Güte. Daher sind die in den Diagrammen 7.2 und 7.3 gezeigten Verläufe der Strukturänderungen, als prozentuale Verbesserungen gegenüber der *schlechtesten* Inkrementierungsvariante zu interpretieren.

Im Widerspruch zur Erwartung sinken die Gütewerte bei der Politik Newsgroup für die Strukturänderungsstufen 1 und 2, ab der sechsten Iteration unter den Gütewert der trivialen Inkrementierung und bleiben auch während aller Iterationen darunter. Bei der Access Newsgroup sinkt die Güte für die Strukturänderungsstufen 1, 2 und 3 bei der vierten bzw. fünften Iteration unter die Güte der trivialen Inkrementierung. Die Erklärung für dieses Verhalten liegt in der Art der Inkrementierung. An einem Beispiel lässt sich dies verdeutlichen.

Beispiel 7.3.2. Für das Beispiel sei die Strukturänderungsstufe zwei angenommen und eine Serie von fünf Iterationen.

Im Verlaufe der zweiten Iteration hat der Cluster *bush,liber* den Kandidaten *bush,-liber,polit* in die Menge seiner gewählten Cluster aufgenommen, da dadurch seine glo-

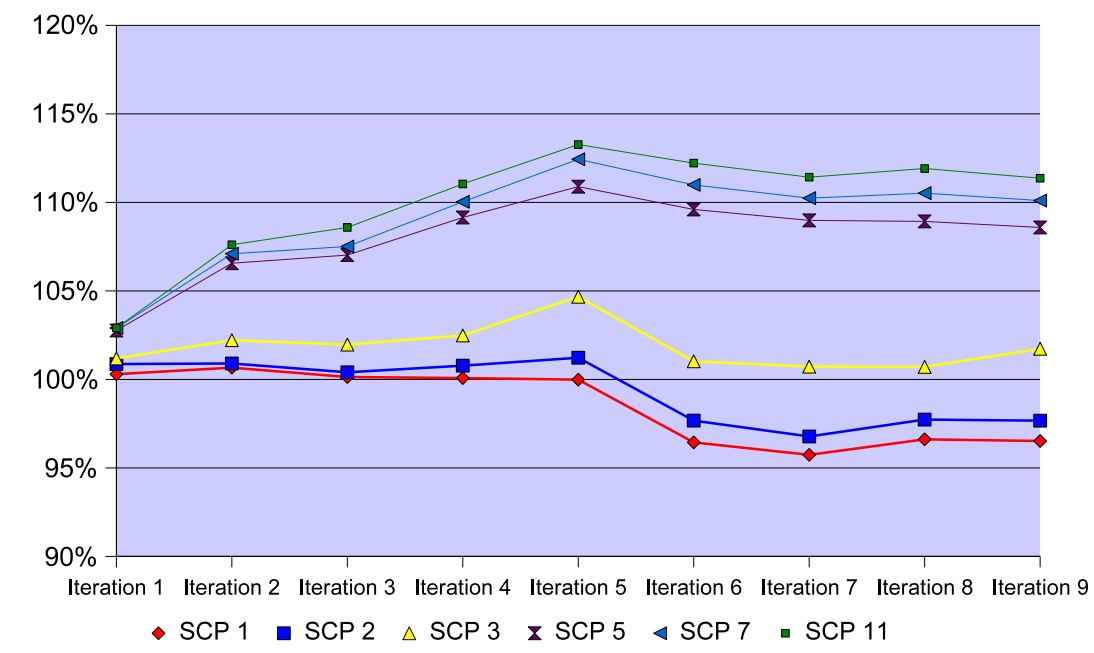


Abbildung 7.2: Iterationen der Politik Newsgroup im Intervall 2

balen Kosten gesunken sind. *bush,liber,polit* bleibt während der fünf Iterationen valide (seine Dokumentenanzahl sinkt nicht unter den MinSupport). Im Verlaufe der nachfolgenden beiden Iterationen werden die Überlappungen des Clusters *bush,liber,polit* mit den anderen Cluster-Kindern von *bush,liber* immer größer, so dass die lokalen Kosten steigen und mit ihnen die globalen Kosten. Angenommen das Entfernen des Clustes *bush,liber,polit* würde in der fünften Iteration zu einer Senkung der globalen Kosten seines Eltern-Clusters führen, so kann er doch nicht entfernt werden, da keine validen Cluster aus der Cluster-Hierarchie entfernt werden dürfen. Dies ist erst ab Stufe drei erlaubt. Das heißt, durch die Wahl des Kandidaten *bush,liber,polit* in Iteration zwei hat der *bush,liber* seine Kosten senken können. Doch im weiteren Verlauf erweist sich diese Wahl als ungünstig, doch sie ist nicht mehr rückgängig zu machen.

Das Beispiel kann auch auf höhere Strukturänderungsstufen übertragen werden. So ist die Wahl eines Kandidaten A mit einem SCP-Wert von 7 bei einer gesetzten Strukturänderungsstufe von 6 auch nur dann während einer nachfolgenden Iteration rückgängig zu machen, wenn der SCP-Wert von A gesunken ist.

Der Algorithmus der Inkrementierung ist in zeitlicher Hinsicht also ein *greedy* Algorithmus. Er wählt einen Kandidaten, der zu diesem Zeitpunkt das Ergebnis verbessert, ohne darauf zu achten, ob diese Wahl in der Zukunft rückgängig gemacht werden kann oder nicht. Wird eine recht hohe Strukturänderungsstufe gewählt, fällt diese *gierige* Vorgehensweise des Verfahrens kaum mehr ins Gewicht, wie die Diagramme in den Abbildungen 7.1, 7.2 und 7.3 zeigen.

Insgesamt ist bei der Entwicklung der Güte der Cluster-Hierarchie keine klare Form (konkav oder konvex) erkennbar. Also kann hier kein klarer Schluss gezogen werden, wel-

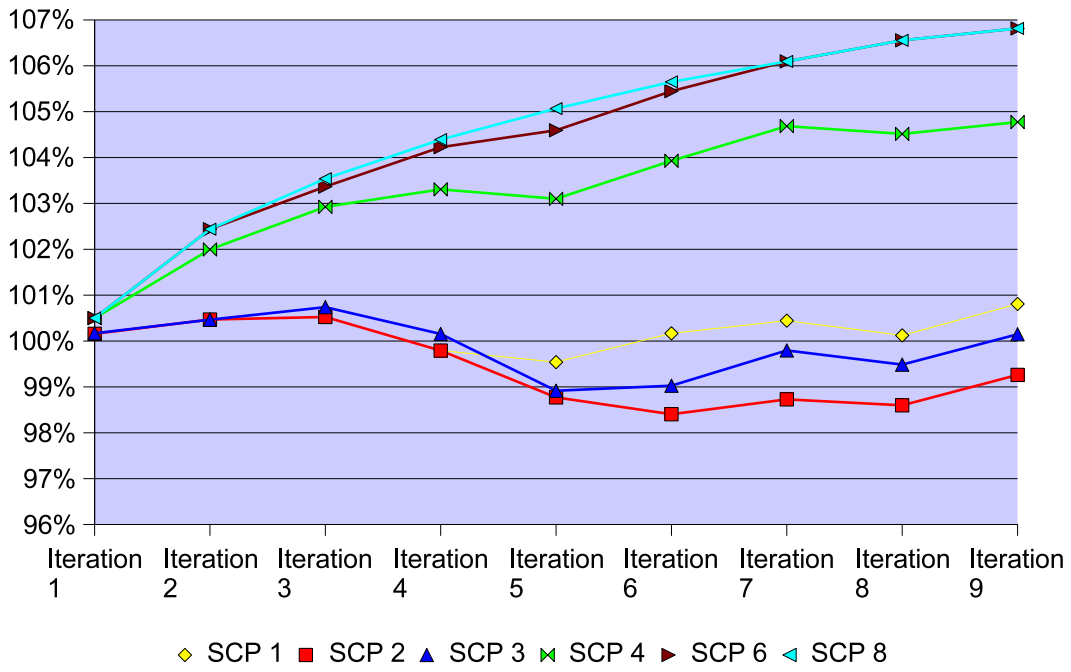


Abbildung 7.3: Iterationen der Access Newsgroup im Intervall 2

che Strukturänderungsstufe eine hohe Güte mit der geringer Gefahr von großen Strukturänderungen mit sich bringt.

Als Abschluss der Betrachtungen der intervallbasierten Inkrementierung mit Standardgewichtung, werden in den Tabelle 7.16 die Anzahl der entfernten (-) und der zugefügten (+) Cluster(-kandidaten) bei Inkrementierung mit dem *Intervall 0* aufgelistet.

	Access		Politik		Philosophie		Hardware	
	-	+	-	+	-	+	-	+
SCP 1	0	12	0	13	0	29	0	13
SCP 2	0	12	1	14	0	29	0	13
SCP 3	2	14	1	14	3	34	5	18
SCP 4	8	23	8	23	10	46	9	23
SCP 5	12	25	14	29	11	40	15	30
SCP 6	13	27	14	29	10	40	12	27
SCP 7	13	27	18	33	12	40	13	24
SCP 8	14	27	19	34	8	31	13	24
SCP 9	-	-	19	34	11	25	-	-
SCP 10	-	-	19	34	-	-	-	-
SCP 11	-	-	22	26	-	-	-	-

Tabelle 7.16: Durchgeführte Strukturänderungen bei Inkrementierung mit Intervall 0

Allgemein lässt sich beobachten, dass sowohl die Anzahl der Cluster, die aus der Hierar-

chie entfernt, als auch die Anzahl der Cluster, die zur Hierarchie hinzugefügt werden, mit größer werdender Strukturänderungsstufe steigt. Nur bei den letzten Stufenerhöhungen kommt es wieder zu einer Verringerung der Anzahl. Dieser Effekt ist durch die Tatsache bedingt, dass bei steigender Strukturänderungsstufe die Teilbäume, die entfernt werden dürfen, größer werden und Änderungen, die in diesem Teilbäumen stattgefunden haben, nicht mehr aufgeführt werden (weil der Teilbaum ja nicht mehr zur Cluster-Hierarchie gehört).

Zwischen den durchgeführten Strukturänderungen (Tabelle 7.16) und den Güteverläufen bei Variierung der Strukturänderungsstufe (Abbildung 7.1) kann eine Beziehung abgelesen werden. Ein Sprung im Verlauf der Güte, geht auch mit einem überdurchschnittlichen Ansteigen der durchgeführten Strukturänderungen einher. So ist in der Access Newsgroup beim Übergang von Stufe drei auf Stufe vier ein starkes Ansteigen der Güte zu sehen. Gleichfalls steigen die durchgeführten Strukturänderungen von insgesamt 16 auf 31. Ähnliches ist bei der Politik Newsgroup zu beobachten: ein Sprung zwischen Stufe drei und vier, einhergehend mit einer Zunahme der durchgeführten Strukturänderungen von 15 auf 31.

7.3.2 Alternative Gewichtungen

Für alle bisherigen Test des Clusterings wurde der Standardgewichtungsfaktor für die Formel der globalen Kosten (siehe Abschnitt 5.2.4) in Form der Anzahl der Dokumente angenommen. Wie bereits erwähnt, sind aber auch andere Gewichtungsfaktoren denkbar. Die Evaluation in diesem Abschnitt soll zeigen, ob die Wahl des Gewichtungsfaktors einen Einfluss auf die Inkrementierung hat. Es wurde die Access Testkollektion exemplarisch mit zwei weiteren Gewichtungsfaktoren geclustert. Der erste Gewichtungsfaktor ist der Logarithmus der Dokumentanzahl ($\log(|D|)$) und der zweite die Summe der Anzahl der Kandidaten und der Anzahl der gewählten Cluster in dem jeweiligen Cluster(-kandidaten) ($(|N|+|M|)$). Die Strukturstatistiken für die Cluster-Hierarchien befinden sich im Anhang in Abschnitt A.1.

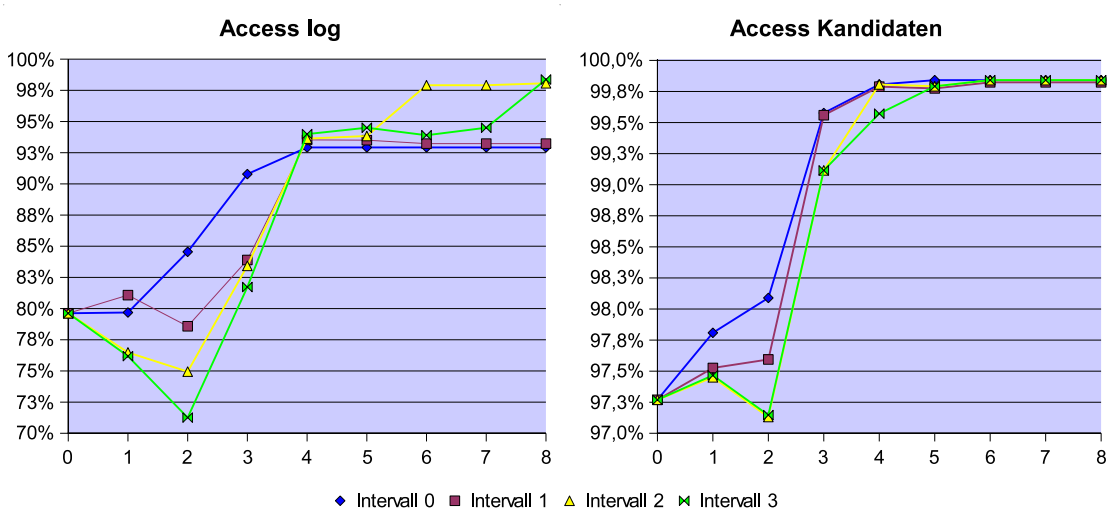


Abbildung 7.4: Intervall-Inkrementierung der Access Newsgroup mit alternativen Gewichtungen

Die Kurven der Inkrementierungen dieser beiden Gewichtungsfaktoren weisen eine signifikant andere Form auf als die der Inkrementierung mit Standardgewichtung. Bei beiden hat besonders die Inkrementierung mit dem Intervall 0 einen konvexen Verlauf, erreicht also schon bei niedriger Strukturänderungsstufe einen hohen Grad an Güte. Allerdings ist der Abstand zum optimalen Clustering selbst bei der höchsten Strukturänderungsstufe noch recht hoch, verglichen mit den Ergebnissen der Inkrementierung mit Standardgewichtung.

Der schnelle Anstieg bei den niedrigeren Strukturänderungsstufen lässt sich dadurch erklären, dass die beiden Gewichtungsfaktoren den Einfluss der globalen Kosten der Kind-Cluster auf die globalen Kosten des Eltern-Clusters stärken. Und wie aus den Strukturstatistiken für die Standardgewichtung aus Abschnitt 7.2.1 hervorgeht, sind in den unteren Ebenen der Cluster-Hierarchie (näher zu den Blättern) die durchschnittlichen globalen Kosten geringer als in den oberen Ebenen (Diese Beobachtung gilt auch für die alternativen Gewichtungen.). Das hat zur Folge, dass bei einer stärkeren Gewichtung der Kind-Cluster Änderungen, die in den unteren Regionen der Hierarchie passieren, größeren Einfluss auf die Güte der gesamten Cluster-Hierarchie haben. Die Veränderungen an der Hierarchie treten schon bei nicht all zu hoher Strukturänderungsstufe auf.

Die Wahl des Gewichtungsfaktors hat also offensichtlich einen Einfluss auf das Verhalten des Clusterverfahrens. Eine genauere Betrachtung der Unterschiede würde allerdings den Rahmen dieser Diplomarbeit sprengen.

8 Fazit und Ausblick

In der Einleitung und im Kapitel 3 wurden Kriterien festgelegt, welche die Clusterhierarchie und das sie erstellende Clusterverfahren erfüllen sollten. In diesem Kapitel wird überprüft, inwiefern diese Zielsetzungen von dem in dieser Diplomarbeit entwickelten Verfahren erreicht wurden. Auch werden Ansatzpunkte für weitere Forschungsmöglichkeiten aufgezeigt.

In Kapitel 3 wurden verschiedene Anforderungen an das Clusterverfahren gestellt. Es soll eine *Quasihierarchie* erstellen. In einer solchen Hierarchie darf das Dokument eines Eltern-Clusters in mehreren seiner Kind-Clustern auftauchen. Die Cluster dürfen sich also überlappen. Weiterhin soll ein Cluster einen zumindest groben Hinweis auf seinen Inhalt bereitstellen. Diese Kriterien wurden von dem gewählten Ansatz ([FWE03]) bereits erfüllt. Die Cluster(-kandidaten) (*frequent term sets*) weisen eine überlappende Struktur auf und ihre Terme bieten einen stichwortartigen Eindruck ihres Inhaltes. Allerdings ist ein Nachteil des Verfahrens [FWE03], dass sehr breite, *buschige* Clusterstrukturen entstehen. Wie der Vergleich in Abschnitt 7.2.3 gezeigt hat, wurde dieser Nachteil durch das in dieser Diplomarbeit entwickelten Verfahren behoben. Ein Eltern-Cluster hat nun eine weitaus übersichtlichere Anzahl von Kind-Clustern.

Das Hauptaugenmerk dieser Diplomarbeit liegt auf der inkrementellen Erweiterung einer bestehenden Clusterhierarchie. Sie soll es ermöglichen, der Hierarchie neue Beiträge hinzuzufügen und geänderte Dokumente, wenn dies erforderlich ist, weiteren Clustern zuzuordnen. Dabei sollen auch Strukturänderungen durchgeführt werden können, welche die *Güte* der Clusterhierarchie gewährleisten sollen. Die Frage, die sich stellt, ist die nach den minimal notwendigen Strukturänderungen zur Sicherstellung einer akzeptablen Güte. Wie kann also der Zielkonflikt zwischen *maximaler Güte* und *minimalen Strukturänderungen* gelöst werden? Wie die Evaluation der inkrementellen Erweiterung gezeigt hat (siehe 7.3.1), kann keine klare Antwort auf diese Frage gegeben werden. Je mehr Strukturänderungen zugelassen sind, desto größer ist zwar im Allgemeinen auch die Güte der Hierarchie (welche durch die Inkrementierung entsteht), aber es gibt folgende problematische Gesichtspunkte:

1. Die Güte der Hierarchie kann durch Strukturänderungen im Verlaufe mehrerer Inkrementierungen auch sinken[1], wenn nicht die maximal möglichen Strukturänderungen ausgeführt werden.

2. Der Verbesserungsverlauf der Güte bei Inkrementierung mit verschiedenen Strukturänderungsstufen weist Sprünge auf und ist ansonsten eher linear.

Der Grad der Verbesserung verändert sich also in nicht klar voraussagbarer Weise mit den zulässigen Strukturänderungen. Es kann nur eine klare Aussage getroffen werden: Wenn die maximal möglichen Strukturänderungen zugelassen werden, ist davon auszugehen, dass die Abweichung der Güte vom Optimum minimal ist. Doch sind hier auch die durchgeführten Strukturänderungen am größten.

[1] Im Vergleich zur trivialen Erweiterung, wenn lediglich die neuen oder geänderten Dokumente ihren Clustern zugeordnet werden und keine Strukturänderungen erlaubt sind.

Dem Administrator einer Newsgroup kann also keine klare Empfehlung gegeben werden, mit welchen Einstellungen (in Bezug auf die zulässigen Strukturänderungen) die Inkrementierung ausgeführt werden sollte. Eine mögliche Lösung ist die Inkrementierung interaktiv mit dem Anwender durchzuführen. Hierbei könnten in einem ersten Schritt von der Applikation die *Sprungpunkte* ausfindig gemacht werden. Mit *Sprungpunkten* sind die Strukturänderungsstufen gemeint, bei denen eine überdurchschnittliche Steigerung der Güte im Vergleich zur vorhergehenden Stufe zu verzeichnen ist. Die Veränderungen in der Clusterhierarchie, die an diesen Punkten durchgeführt werden müssten, können dem Anwender mitgeteilt werden. Der Anwender kann dann in einem zweiten Schritt entscheiden, welche Strukturänderungsstufe verwendet werden soll. So können unliebsame Überraschungen (wie zum Beispiel das Entfernen als *wichtig* empfundener Cluster) vermieden werden.

Das initiale Clustering der *frequent term sets* weist eine weitere Schwierigkeit auf, welche bisher noch nicht näher angesprochen wurde. Um die aus der Komplexität von $O(2^n)$ resultierenden langen Laufzeiten zu verringern und dennoch das globale Optimum zu finden, wurde ein *branch-and-bound* Algorithmus eingesetzt (siehe 5.3.2.1). Die Hoffnung war, dass sich der Suchraum so weit beschränken lässt, dass der Algorithmus effizient arbeitet. Wie die Evaluation allerdings gezeigt hat, benötigt der Algorithmus bereits ab einer Anzahl von etwa 40 initialen Termen (1-fts) eine horrende Zeitspanne um zu terminieren. Die Approximationsalgorithmen (*n-Greedy* und der *rundenbasierte* branch-and-bound) sind zwar effizienter, erzeugen mitunter aber Clusterhierarchien mit sehr geringer Güte. Wie erste Testevaluationen gezeigt haben, kann das Verfahren der inkrementellen Erweiterung (siehe 6.3) als *Verbesserungsverfahren* eingesetzt werden. Wird nach dem initialen Clustering mit einem Approximationsalgorithmus, das Verfahren der Inkrementierung auf die Clusterhierarchie angewandt (ohne Dokumente nachzufügen), so kann ihre Güte verbessert werden. Dieses Verhalten legt eine weiter Lösungsmöglichkeit des Komplexitätsproblems nahe. Das initiale Clustering wird mit einer geringen Anzahl an initialen Termen und dem *branch-and-bound* Algorithmus durchgeführt. Anschließend wird dann die Anzahl der initialen Terme heraufgesetzt und das Verfahren der inkrementellen Erweiterung auf die bestehende Clusterhierarchie mit diesen zusätzlichen initialen Termen angewandt. Die Frage ist dann: Wie gut ist dieses *inkrementell-initiale* Clustering im Vergleich zu den in dieser Diplomarbeit evaluierten Approximationsalgorithmen? Eine detaillierte Evaluation dieser Aspekte geht über den Rahmen dieser Diplomarbeit hinaus.

Das *inkrementell-initiale* Clustering und die oben angesprochene interaktive Inkrementierung bieten vielversprechende Ansatzpunkte für weitere Forschungen.

Literaturverzeichnis

[ABK99] M. Ankerts, M. M. Breuning und H.-P. Kriegel. OPTICS:Ordering Pointd To Identify the Clustering Structure. Technical report, Institute for Computer Science, University Munich, 1999.

[AK02] J. Allan und G. Kumaran. Details on Stemming in the Language Modelling Framework. Technical report, University of Massachusetts Amherst, 2002.

[AS94] R. Agrawal und R. Srikant, Hrsg. *Fast Algorithms for Mining Association Rules*, 1994.

[BEX02] F. Beil, M. Ester und X. Xu, Hrsg. *Frequent Term-Based Text Clusering*, 2002.

[CKP92] D. R. Cutting, D. R. Karger und J. O. Pedersen, Hrsg. *Scatter/Gather:A Cluster-based Approach to Browsing Large Document Collections*, 1992.

[CLA04] F. Coenen, P. Leng und S. Ahmed. Data Structure for Association Rule Mining: T-Trees and P-Trees. *IEEE Transaction on knowledge and data engineering, Vol.16, No.6*, Seiten 774–778, 2004.

[EKS96] M. Ester, H.-P. Kriegel und J. Sander. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Technical report, Institute for Computer Science, University Munich, 1996.

[Fuh03] N. Fuhr. Information Retrieval - Skriptum, Universität Dortmund, 2003.

[FWE03] B. C. M. Fung, K. Wang und M. Ester, Hrsg. *Hierarchical Document Clustering using frequent items*, San Francisco, 2003.

[Han81] D. J. Hand. *Discrimination and Classification.* John Wiley & Sons, New York, 1981.

[HdV00] D. Hiemstra und A. P. de Vries. Relating the new language models of information retrieval to the traditional retrieval models. Technical report, University of Twente, Enschede, 2000.

[HE89] Prof. Dr. J. Hartung und Dr. B. Elpelt. *Multivariate Statistik - Lehr- und Handbuch der angewandten Statistik.* R. Oldenbourg Verlag, München, 1989.

[HEK89] Prof. Dr. J. Hartung, Dr. B. Elpelt und Dr. K.-H. Klösner. *Statistik - Lehr- und Handbuch der angewandten Statistik.* R. Oldenbourg Verlag, München, 1989.

[HK01] J. Han und M. Kamber, Hrsg. *Data Mining - Concepts and Techniques.* Academic Press, San Diego, 2001.

[HPY00] J. Han, J. Pei und Y. Yin, Hrsg. *Mining frequent patterns without candidate generation*, Dallas, 2000.

[HTF01] T. Hastie, R. Tibshirani und J. H. Friedman. *The Elements of Statistical Learning.* Springer Verlag, Berlin, 2001.

[JMF99] A. K. Jain, M. N. Murty und P. J. Flynn. Data Clustering - A Review. *ACM Computing Surveys, Vol. 31*, Seiten 264–323, 1999.

[Kuh90] R. Kuhlen. Zum Stand pragmatischer Forschung in der Informationswissenschaft. In *Pragmatische Aspekte beim Entwurf und Betrieb von Informationssystemen*, Seiten 13–18. Proceedings des 1.Internationalen Symposiums für Informationswissenschaft, 1990.

[Por80] M. F. Porter. An algorithm for suffix stripping. Program (Automated Libary and Information Systems), 1980, Seiten 130–137.

[SCZ98] G. Sheikholeslami, S. Chatterjee und A. Zhang, Hrsg. *WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial*, New York, 1998.

[SKK00] M. Steinbach, G. Karypis und V. Kumar. A Comparison of Document Clustering Techniques. Technical report, Department of ComputerScience and Engineering, University of Minnesota, 2000.

[vR79] C. J. van Rijsbergen. *Information Retrieval.* Butterworth, London, 1979.

[WXL99] K. Wang, C. Xu und B. Liu, Hrsg. *Clustering Transactions using Large Items*, 1999.

[ZCC] Q. Zou, H. Chiu und W. W. Chu. Using Pattern Decomposition Methods for Finding All Frequent Patterns in Large Datasets.

[ZE89] O. Zamir und O. Etzioni, Hrsg. *Web Document Clustering: A Feasibility Demonstration*, 1989.

A Ergebnisstabellen

A.1 Initiales Clustering

	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5	Gesamt
1.FTS	-	40	381	616	215	2	1254
2.Cluster	1	11	48	52	19	1	132
3.mit Kand.	1	11	44	11	0	0	67
4.GC (av)	0,5891	0,4224	0,1621	0,1275	0	0	0,2055
5.LC (av)	2,0982	1,3586	0,2278	0,1275	0	0	0,4249
6.OV (av)	15358,0	2341,3	100,4	0	0	0	679,6
7.UP (av)	5742,0	1073,9	181,1	115,4	0	0	399,9
8.Kinder (av)	11,0	4,4	1,2	1,0	0	0	1,8
9.Kand. (av)	29,0	13,4	2,1	1,1	0	0	4,2
10.Dok. (av)	10056,0	2195,5	968,1	770,8	691,4	618,0	1019,0
11.Dok. ohne	49,0	20,6	75,9	64,5	-	-	64,5
Kand (av)	0,49%	0,94%	7,84%	8,36%	-	-	6,33%
12.Dok. in	1263,0	312,5	193,9	175,9	-	-	226,3
Misc (av)	12,56%	14,23%	20,03%	22,82%	-	-	22,21%

Tabelle A.1: Access Newsgroup: Referenz-Clustering mit Gewichtungsfaktor $\log(|D|)$

	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5	Gesamt
1.FTS	-	40	381	616	215	2	1254
2.Cluster	1	13	29	28	7	0	78
3.mit Kand.	1	13	18	6	0	0	38
4.GC (av)	0,9741	0,6644	0,4740	0,3427	0	0	0,5316
5.LC (av)	2,0096	0,7794	0,5023	0,3427	0	0	0,6116
6.OV (av)	13995,0	816,5	227,3	80,3333333	0	0	767,9
7.UP (av)	6214,0	617,7	421,8	294,2	0	0	621,1
8.Kinder (av)	13,0	2,2	1,5	1,2	0	0	2,0
9.Kand. (av)	27,0	10,3	4,9	2,5	0	0	7,0
10.Dok. (av)	10056,0	1744,0	1042,1	776,5	741,9	0	1152,4
11.Dok. ohne	49,0	26,7	60,1	49,2	-	-	46,7
Kand (av)	0,49%	1,53%	5,77%	6,33%	-	-	4,05%
12.Dok. in	1379,0	235,7	265,6	229,5	-	-	278,9
Misc (av)	13,71%	13,51%	25,48%	29,56%	-	-	24,21%

Tabelle A.2: Access Newsgroup: Referenz-Clustering mit Gewichtungsfaktor $(|N| + |M|)$

A.2 Intervall Inkrementierung

SCP	Vollständig		Intervall 1		Intervall 2		Intervall 3	
0	1,05857	93,70%	1,05857	93,70%	1,05857	0,93699	1,05857	93,70%
1	1,05131	94,35%	1,05745	93,80%	1,05957	0,93610	1,06012	93,56%
2	1,04054	95,32%	1,05734	93,81%	1,06739	0,92924	1,06838	92,84%
3	1,03550	95,79%	1,05649	93,88%	1,05795	0,93754	1,05917	93,65%
4	1,01159	98,05%	1,01232	97,98%	1,01124	0,98084	1,01214	98,00%
5	1,00469	98,72%	1,00426	98,77%	1,00199	0,98989	1,00507	98,69%
6	0,99670	99,51%	0,99686	99,50%	0,99191	0,99995	0,99402	99,78%
7	0,99670	99,51%	0,99442	99,74%	0,99191	0,99995	0,99191	99,99%
8	0,99395	99,79%	0,99442	99,74%	0,99191	0,99995	0,99191	99,99%

Tabelle A.3: Access Newsgroup: absolute und prozentuale Werte der Inkrementierung nach Intervallen (die prozentualen Werte beziehen sich auf das Referenz-Clustering)

SCP	Vollständig		Intervall 1		Intervall 2		Intervall 3	
0	1,24137	89,76%	1,24137	89,76%	1,24137	89,76%	1,23470	90,25%
1	1,23067	90,54%	1,27344	87,50%	1,28613	86,64%	1,30362	85,48%
2	1,21659	91,59%	1,25704	88,64%	1,27100	87,67%	1,29942	85,75%
3	1,21659	91,59%	1,21411	91,78%	1,22023	91,32%	1,23884	89,95%
4	1,15119	96,79%	1,14592	97,24%	1,14503	97,31%	1,15075	96,83%
5	1,14244	97,54%	1,14181	97,59%	1,14322	97,47%	1,14614	97,22%
6	1,13342	98,31%	1,12966	98,64%	1,12824	98,76%	1,12693	98,88%
7	1,13301	98,35%	1,12941	98,66%	1,12745	98,83%	1,12770	98,81%
8	1,13028	98,58%	1,11662	99,79%	1,11466	99,97%	1,11466	99,97%
9	1,13028	98,58%	1,11662	99,79%	1,11466	99,97%	1,11466	99,97%
10	1,13028	98,58%	1,11662	99,79%	1,11466	99,97%	1,11466	99,97%
11	1,11658	99,79%	1,11662	99,79%	1,11466	99,97%	1,11466	99,97%

Tabelle A.4: Politik Newsgroup: absolute und prozentuale Werte der Inkrementierung nach Intervallen (die prozentualen Werte beziehen sich auf das Referenz-Clustering)

SCP	Vollständig		Intervall 1		Intervall 2		Intervall 3	
0	1,31329	84,82%	1,31329	84,82%	1,31329	84,82%	1,31329	84,82%
1	1,22369	91,03%	1,25721	88,61%	1,25729	88,60%	1,27378	87,46%
2	1,22369	91,03%	1,25698	88,62%	1,25218	88,96%	1,26863	87,81%
3	1,21191	91,92%	1,25030	89,10%	1,24240	89,66%	1,26261	88,23%
4	1,17667	94,67%	1,17798	94,57%	1,17448	94,85%	1,17183	95,06%
5	1,14954	96,91%	1,16295	95,79%	1,16267	95,81%	1,16503	95,62%
6	1,14284	97,48%	1,12506	99,02%	1,13074	98,52%	1,14272	97,48%
7	1,13729	97,95%	1,12282	99,21%	1,12423	99,09%	1,14515	97,28%
8	1,12127	99,35%	1,11558	99,86%	1,11695	99,73%	1,11696	99,73%
9	1,11431	99,97%	1,11558	99,86%	1,11695	99,73%	1,11696	99,73%

Tabelle A.5: Philosophie Newsgroup: absolute und prozentuale Werte der Inkrementierung nach Intervallen (die prozentualen Werte beziehen sich auf das Referenz-Clustering)

SCP	Vollständig		Intervall 1		Intervall 2		Intervall 3	
0	0,89597	91,64%	0,89597	91,64%	0,89597	91,64%	0,89597	91,64%
1	0,85274	96,29%	0,85834	95,66%	0,85718	95,79%	0,86059	95,41%
2	0,85274	96,29%	0,85834	95,66%	0,85718	95,79%	0,85982	95,49%
3	0,85096	96,49%	0,85579	95,94%	0,85402	96,14%	0,85863	95,63%
4	0,84187	97,53%	0,83509	98,32%	0,83356	98,50%	0,83706	98,09%
5	0,82663	99,33%	0,82713	99,27%	0,82671	99,32%	0,82934	99,00%
6	0,82601	99,40%	0,82749	99,22%	0,82590	99,41%	0,82707	99,27%
7	0,82117	99,99%	0,82381	99,67%	0,82339	99,72%	0,82339	99,72%
8	0,82117	99,99%	0,82381	99,67%	0,82339	99,72%	0,82339	99,72%

Tabelle A.6: Hardware Newsgroup: absolute und prozentuale Werte der Inkrementierung nach Intervallen (die prozentualen Werte beziehen sich auf das Referenz-Clustering)