KI – Künstliche Intelligenz
Special Issue on Adaptive Multimedia Retrieval
Volume 4, pp. 11-17, November 2004. ISSN 0933-1875.

Automatische Kategorisierung für Web-basierte Suche

— Einführung, Techniken und Projekte —

Benno Stein und Sven Meyer zu Eißen

Suchmaschinen wie Google, Overture oder Altavista haben sich zu den Killeranwendungen des Internets gemausert. Sie indizieren bis zu 4 Milliarden Seiten und ermöglichen einen extrem schnellen Zugriff hierauf. Dabei bilden Einzelwortsuchanfragen (keyword search) den Einstiegspunkt für fast alle Benutzer – und hier beginnt auch das Problem der meisten Suchenden: Nicht selten werden zu einer Anfrage mehrere Tausend Dokumente geliefert, und es hängt von der Erfahrung des Benutzers ab, durch eine Anfrageverfeinerung die Suche zu fokussieren.
Einen Ausweg aus diesem Problem stellt die kategorisierende Suche dar. Mittels ihr gelingt eine Einschränkung des Suchraums und, an Stelle des zufallsgetriebenen Stocherns in der Dokumentliste kann sogar eine Benutzerführung treten. Voraussetzung hierfür ist, dass die von den Suchmaschinen gelieferten Dokumente bestimmten Kategorien zugeordnet sind, und dass ein kategoriebasierter Zugriff in der Benutzerschnittstelle realisiert ist. Dieses Potential haben sowohl die Anwender als auch die Entwickler von Suchmaschinen erkannt: Kategorisierende Suchmaschinen sind im Kommen. Dieser Beitrag widmet sich dem Thema; er stellt zugrundeliegende Konzepte vor, diskutiert vewandte Fragestellungen und gibt einen Überblick über entsprechende Suchmaschinenprojekte.

1 Einführung

Kategorisierung in Zusammenhang mit Web-basierter Suche bedeutet die automatische Sortierung einer Menge von Dokumenten aus dem Korpus „Internet“. Die für eine solche Sortierung in Frage kommenden Kategorien sind entweder unbekannt oder sie müssen aus einem Katalog ausgewählt werden, der in der Größenordnung 10^5 Kategorien enthält. Aus Akzeptanzgründen sollte die Einordnung der Dokumente in passende Kategorien den Suchprozess eines Anwenders kaum verzögern—also quasi auf Knopfdruck geschehen.
Dass eine solche „ad-hoc-Kategorisierung“ möglich ist und dass sie sogar eine gewisse Alltagstauglichkeit erreicht hat, demonstriert das Vivísimo-Projekt. Abbildung 1 zeigt die Ergebnisseite einer Suche mit Vivísimo für die Anfrage „Turing Test“. Links auf dieser Seite sind ca. 200 Dokumentausschnitte, sogenannte „snippets“, innerhalb eines Kategoriebaums einsortiert; die rechte Seite zeigt die übliche Listensicht, die hier gemäß der einzelnen Kategorien sortiert ist.
Suchmaschinen, die besondere Suchleistungen wie z. B. eine Kategorisierung oder andere Ergebnisanalysen durchführen, sind oft als Meta-Suchmaschinen realisiert. D. h., sie pflegen keinen eigenen Index der Internet-Dokumente, sondern benutzen (und bezahlen) die Such-Services der großen Suchmaschinenfirmen wie Google, Inktomi, Fast, Microsoft, etc. Der gesamte Ablauf einer kategorisierenden Suche, angefangen von der Eingabe des Anfragewortes bis hin zur Ergebnisdarstellung, gliedert sich in fünf Schritte:

1. Analyse der Anfrage und deren Weiterleitung an registrierte Suchmaschinen.
2. Indizierung der erhaltenen Dokumentausschnitte auf Basis eines bestimmten Dokumentmodells.
3. Struktursuche, d. h., Analyse der Beziehungen zwischen den indizierten Dokumentausschnitten hinsichtlich eines Ähnlichkeitsmaßes.
4. Konstruktion von Kurzüberschriften (Kategorienamen, Labels) für Gruppen ähnlicher Dokumentausschnitte.
5. Darstellung der gefundenen Kategoriestruktur und der Dokumentausschnitte.

Die Schritte 2 und 3 bilden das Herz jedes Kategorisierungsprozesses und werden im nächsten Abschnitt behandelt. Man unterscheidet hierbei zwischen überwachten Ansätzen, die auf einem

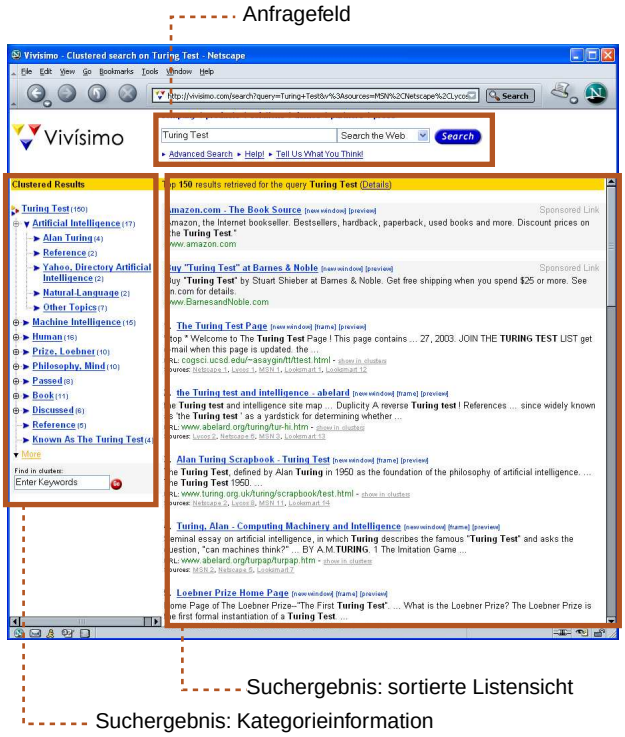

Abbildung 1: Snapshot der Vivísimo-Benutzerschnittstelle. Links befindet sich ein Baum mit den automatisch konstruierten Kategorien. Auf der rechten Seite sind die Suchergebnisse in einer klassischen, nach Kategorien sortierten Listensicht aufgeführt.

existierenden, festen Kategorisierungsschema, einem sogenannten „directory“ aufsetzen und den unüberwachten Ansätzen, die auf einer Clusteranalyse basieren. Der Abschnitt 3 behandelt Aspekte von Schritt 4 und insbesondere die Schlüsselfrage: „Wie gut sind automatisch erzeugte Kategorien?“. Der letzte Abschnitt gibt Beispiele zu Schritt 5 sowie eine Übersicht über kommerzielle und wissenschaftliche Suchmaschinenprojekte, in denen Kategorisierungstechnologie implementiert ist.

2 Kategorisierungsansätze

Menschliche Editoren benutzen bewusst und unbewusst eine Reihe von Regeln, um Dokumente in Kategorien einzusortieren. Aus Klassifikationssicht kann ein solches Regelwerk sehr leistungsfähig sein, aus Sicht der Wissensakquisition und -pflege, oder in Punkto Erweiterbarkeit um neue Kategorien ist dieser Ansatz problematisch. Vor allem aus Kostengründen haben Ansätze aus dem Bereich des maschinellen Lernens die regelbasierten Ansätze weitgehend verdrängt, und der Fokus dieses Textes liegt auf den letzteren.

Automatische Dokumentenkategorisierung kann überwacht oder unüberwacht erfolgen. Im ersten Fall liegt eine typische Klassifikations- bzw. Auswahlsituation vor, bei der Dokumente in ein existierendes Schema einzuordnen sind. Im zweiten Fall gilt es, eine in der Dokumentenmenge latent vorhandene Struktur – das Kategorisierungsschema – durch eine Clusteranalyse zu identifizieren. Die hierbei gefundenen Dokumentgruppen werden als Kategorien interpretiert.
Ein wesentlicher Schwachpunkt des überwachten Ansatzes ist der Aufbau und die Pflege eines Kategorieschemas durch menschliche Editoren. Auch die Konstruktion eines entsprechenden Klassifizierers ist nicht unproblematisch, denn nur ein Bruchteil aller indizierten Internetseiten kann zu Lernzwecken verwendet werden. Die Folge ist eine mäßige Klassifikationsqualität, insbesondere dann, wenn nur Dokumentausschnitte die Eingangsinformation des Klassifizierers bilden. Ein Cluster-Ansatz arbeitet quasi anfragespezifisch, da er nur die Dokumente der Anfrage berücksichtigt; die Erfahrung zeigt zudem, dass die kurzen Dokumentausschnitte ausreichend sind, um eine Struktur zwischen den Dokumenten einer Anfrage aufzudecken. Die größten Schwächen von Cluster-basierten Ansätzen resultieren aus dem Problem, aussagekräftige Kategoriebezeichner zu finden, und der Schwierigkeit, die gefundenen Kategorien in einem ontologischen Sinne zueinander in Beziehung zu setzen.
Zur Zeit ist es schwer abzuschätzen, mit welchem der beiden Ansätze sich auf Dauer die leistungsfähigeren Ergebnisse produzieren lassen, beide haben ihre Vor- und Nachteile (siehe Tabelle 1). Klar ist, dass bei den kategorisierenden Suchmaschinen zur Zeit die Cluster-basierten Ansätze dominieren, wobei Vivísimo die Nase vorn hat.

	schemabasiert (überwacht)	Cluster-basiert (unüberwacht)
sinnvolle Kategoriebezeichner	+	○
Unabhängigkeit bzgl. Domänen und Sprachen	–	+
Klassifikationsqualität	○	○
Kategoriestruktur passt zum Ontologieverständnis	+	–
anfragespezifische Kategorien	○	+

Tabelle 1: Schemabasierte versus Cluster-basierte Kategorisierung: Vor- und Nachteile aus Sicht der Web-basierten Suche.

2.1 Indexing und Dokumentenmodelle

Die von den Suchmaschinen für eine Anfrage gelieferten Dokumentausschnitte werden nicht in ihrer originalen Form klassifiziert oder geclustert, sondern zunächst in eine kanonische Form überführt. Dieser Abstraktionsprozess heißt Indexing und beinhaltet u.a. Parsen, Stoppworteliminierung, Stammwortreduktion und die Berechnung eines Merkmalsvektors gemäß eines Dokumentmodells.

Während des Parsens werden die HTML-Auszeichnungen, die Bilder sowie Scripting-Code entfernt. Worte, die sehr häufig auftreten, werden als Stoppworte bezeichnet; sie tragen wenig Information zur Unterscheidung zwischen Texten bei und werden auf Basis bekannter Stoppwortlisten gelöscht. Algorithmen zur Stammwortreduktion dienen zur Entfernung von Wortteilen, die Deklinations-, Konjugations- oder Numerus-anzeigend sind; d. h., sie reduzieren ein Wort auf dessen Stamm-Morpheme [1].
Ein Dokumentmodell ist ein Konzept, das beschreibt, wie eine aussagekräftige Menge von Merkmalen $\mathbf{d}$ von den vorverarbeiteten Worten eines Dokuments zu berechnen sind. „Aussagekräftig“ heißt, dass sich eine Funktion φ angeben läßt, die von den Merkmalsmengen $\mathbf{d_1}$ und $\mathbf{d_2}$ zweier Dokumente d_1, d_2 auf das Intervall $[0;1]$ abbildet und die folgende Eigenschaft hat: Falls $\varphi(\mathbf{d_1}, \mathbf{d_2})$ nahe bei Eins ist, so sind die Dokumente d_1 und d_2 ähnlich zueinander – bzw. umgekehrt: Ein Wert nahe bei Null steht für eine große Unähnlichkeit. Grundsätzlich lassen sich alle Dokumentmodelle hinsichtlich folgender vier Dimensionen einordnen:

1. Dem Termkonzept, das die Granularität der Texteinheiten definiert, die als Merkmal verwendet werden: Worte, n-Gramme, Nominalphrasen, Halbsätze, etc.
2. Dem Termgewichtungsschema, das eine Vorschrift zur Beurteilung der Bedeutung von Termen definiert.
3. Der Verwendung linguistischer Statistiken wie Konzept-, Satzbau- und Wortklassenanalysen (latent semantic indexing, syntactic group analysis, part-of-speech analysis).
4. Der Verwendung einfacher Textstatistiken und präsentationsbezogener Analysen.

Die beiden ersten Dimensionen bilden die Grundlage der meisten Dokumentmodelle, die zur Textkategorisierung eingesetzt werden; die anderen Dimensionen spielen eine wichtige Rolle im Bereich der Text-Genre-Analyse oder der Textsynthese.

Sei n die Anzahl der Dokumente für eine gegebene Anfrage und sei m die Gesamtzahl der verschiedenen Worte nach der Vorverarbeitung, auch Wörterbuch oder „Bag-of-Words“ genannt. Dann ist ein Dokumentmodell d in der einfachsten Form ein Vektor der Länge m, dessen i-ter Eintrag anzeigt, ob der i-te Term des Wörterbuches in d vorkommt. Weil die Dokumente zu Vektoren im m-dimensionalen Raum aller Wörterbucheinträge abstrahiert sind, heißt dieses Dokumentmodell auch Vektorraummodell (siehe Abbildung 2). Durch die Einführung von Termgewichten an Stelle Boolescher Werte lässt sich das einfache Vektorraummodell deutlich verbessern [2, 3]: Die am weitesten verbreitete Variante kombiniert die normalisierte Termhäufigkeit tf mit der inversen Dokumenthäufigkeit idf. Hierbei bezeichnet $tf(i,j)$ die Häufigkeit von Term i in Dokument j, und $idf(i)$ ist definiert als $\log(\frac{n}{df(i)})$, wobei $df(i)$ der Anzahl derjenigen Dokumente entspricht, die Term i enthalten. Die Idee hinter dem idf-Termgewichtungsschema ist, dass Terme, die nur in wenigen Dokumenten Verwendung finden, sich besonders zur Diskriminierung eignen.
Man beachte, dass das Vektorraummodell typischerweise keine Information über die Reihenfolge enthält, in der die Worte in einem Dokument auftauchen. Ein in dieser Hinsicht wesentlich anspruchsvolleres Dokumentmodell basiert auf Suffix-Bäumen, die in Abschnitt 2.4 vorgestellt werden.
Dokumentmodelle und Ähnlichkeitsfunktionen φ bedingen einander: Das Vektorraummodell und seine Varianten spielen sehr gut mit dem Kosinus-Ähnlichkeitsmaß (= normalisiertes Skalarprodukt) zusammen, es kann aber auch mit dem Euklidischen Abstandsmaß, einem Überdeckungsmaß oder anderen Abstandskonzepten benutzt werden. Das Suffix-Baum-Dokumentmodell hingegen benötigt ein Maß, das die Ähnlichkeit zwischen zwei Gra-

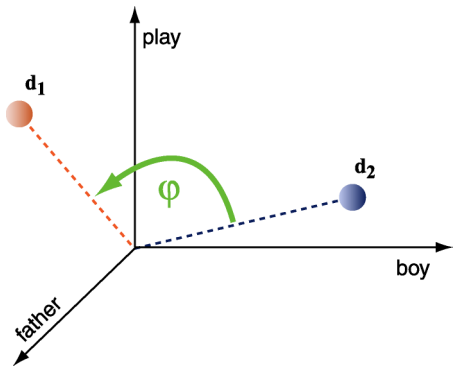

Abbildung 2: Illustration des Vektorraummodells: Die zwei Punkte stehen für Dokumente im Vektorraum des Wörterbuchs *{boy, father, play}*. Als Ähnlichkeitsfunktion φ dient die Kosinus-Ähnlichkeit, die dem Kosinus des Winkels zwischen $\mathbf{d_1}$ und $\mathbf{d_2}$ entspricht.

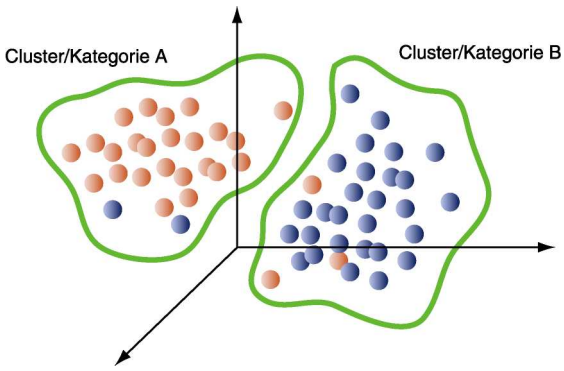

Abbildung 3: Cluster-Algorithmen versuchen, Cluster von Dokumenten zu identifizieren mit dem Ziel, die Intra-Cluster-Ähnlichkeit im Vergleich zu der Inter-Cluster-Ähnlichkeit möglichst hoch zu gestalten. Die Cluster entsprechen den gesuchten Kategorien.

phen abschätzt; in der Praxis ist diese Ähnlichkeitsberechnung Teil des Algorithmus zum Suffix-Baum-Clustern.

2.2 Verzeichnisbasierte Kategorisierung

Hinsichtlich der Kategorisierung von Dokumenten in ein festes Klassifikationsschema existieren umfangreiche Forschungsarbeiten, die ihren Ursprung in den Sechziger Jahren haben. Die zum Einsatz kommenden Verfahren des maschinellen Lernens basieren auf dem Vektorraummodell und verwenden probabilistische Methoden, Entscheidungsbäume, Regression, neuronale Netze, Support-Vektor-Maschinen oder Benutzerprofile [4, 5].
Im Zusammenhang mit Web-basierter Suche ist das Prinzip der verzeichnisbasierten Kategorisierung jedoch die Ausnahme: Hier ist nur das Yahoo!-Planet-Projekt bekannt geworden. Die Ursache dafür liegt in den bereits angesprochenen Schwierigkeiten, die aus der Wissensakquisition und -pflege oder dem maschinellen Lernen resultieren. Verzeichnisbasierte Kategorisierung ist erfolgreich in eingegrenzten Domänen, wie sie in Form der Reuters-Datenbank mit Wirtschaftsnachrichten oder der Ohsumed-Textkollektion aus dem Bereich der Medizin vorliegen (vgl. Tabelle 3).
Ein Klassifizierer für die „allgemeinste Domäne Internet“ müsste in der Lage sein, Dokumente einzuordnen, die aufgrund beliebiger Anfragen geliefert werden. D. h., als Kategorieschema wäre eine Art allgemeine Ontologie notwendig, vergleichbar dem Yahoo!-Verzeichnis.
Obwohl es einige starke Argumente gibt, die für eine verzeichnisbasierte Kategorisierung von Web-Suchergebnissen sprechen (siehe Tabelle 1), bilden Cluster-Algorithmen die Schlüsseltechnologie für alle kategorisierenden Suchmaschinen, die zur Zeit online sind.

2.3 Clustern unter dem Vektorraummodell

Definition (Clustering) *Sei D eine Menge von indizierten Dokumenten. Ein (exklusives) Clustering $\mathcal{C} = \{C \mid C \subseteq D\}$ von D ist eine Aufteilung von D in paarweise disjunkte Teilmengen mit $\bigcup_{C_i \in \mathcal{C}} C_i = D$.*

Abbildung 3 zeigt ein Clustering mit zwei Clustern bzw. Kategorien. Informell gesagt, ein Clustering bzw. eine Kategorisierung ist „gut“ hinsichtlich eines Ähnlichkeitsmaßes φ, falls die durchschnittliche Ähnlichkeit der Dokumente innerhalb eines Clusters groß ist, verglichen mit der durchschnittlichen Ähnlichkeit von Dokumenten, die aus verschiedenen Clustern stammen. Cluster-Algorithmen sind speziell entwickelte Strategien, um ein globales Optimum hinsichtlich dieser Ähnlichkeitswünsche zu finden. Aus algorithmischer Sicht werden die n Elemente einer Dokumentkollektion D oft als Knoten eines gewichteten Graphen $G = \langle V, E, \varphi \rangle$ interpretiert, dessen Kantengewichte als die Ähnlichkeiten der zugehörigen Dokumente definiert sind.
Es ist schwer zu beurteilen, welcher der existierenden Cluster-Algorithmen für die Aufgabe der Dokumentenkategorisierung am besten geeignet ist. Der folgende Text nennt die wichtigsten Klassen von Cluster-Algorithmen, die in Zusammenhang mit dem Vektorraummodell Verwendung finden (siehe auch Abbildung 4) und stellt ein neues, dichtebasiertes Verfahren vor.

Hierarchische Algorithmen. Hierarchische Algorithmen erzeugen einen Baum mit Knotenteilmengen, in dem sie nach und nach die Knoten des gegebenen Ähnlichkeitsgraphen entweder aufteilen oder zusammenfassen. Um ein eindeutiges Clustering zu erhalten, ist ein zweiter Schritt notwendig, der festlegt, wo innerhalb des Baumes der Aufteilungs- oder Zusammenfassungsprozess gestoppt wird. Die hierfür eingesetzten Heuristiken analysieren die k nächsten Nachbarn, die Cluster-Distanzen und -Durchmesser, die Cluster-Varianzen (Ward), minimale Spannbäume oder Kantenschnitte.

Iterative Algorithmen. Cluster-Algorithmen diesen Typs versuchen, ein existierendes Clustering stufenweise zu verbessern; sie lassen sich weiter einteilen in exemplarbasierte und austauschbasierte Verfahren. Die erstgenannten unterstellen für jeden zukünftigen Cluster einen Repräsentanten – genauer: einen Zentroiden für geometrische Graphen oder einen Medoiden für andere Distanzgraphen – dem die Knoten gemäß ihrer Entfernung zugeordnet werden. Austauschbasierten Verfahren starten mit einem vorläufigen Clustering und tauschen solange Knoten zwischen Clustern aus, bis ein gegebenes Qualitätsmaß erfüllt ist.

Meta-Suchverfahren. Unter diesem Begriff fasst man Algorithmen zusammen, die ein Cluster-Problem als generische Optimierungsaufgabe betrachten. Aus Sicht einer Cluster-Aufgabe bietet diese Herangehensweise ein Maximum an Flexibilität, ist in der Realität aber mit Laufzeiten verbunden, die meist weit über den Laufzeiten anderer Cluster-Algorithmen liegen.

Dichtebasierte Algorithmen. Dichtebasierte Algorithmen versuchen, Cluster gleichmäßiger Dichte zu erzeugen. Im Idealfall bestimmen sie die optimale Cluster-Anzahl automatisch und machen keine Einschränkung bzgl. Cluster-Form und Größenverteilung [6].

Ein solcher dichtebasierter Clustering-Algorithmus ist MajorClust [7]; er hat sich im Bereich der Dokumentkatgorisierung bewährt und findet u. a. in der Meta-Suchmaschine Alsearch Verwendung. Da es sich um ein relativ neues Verfahren handelt und sich sein

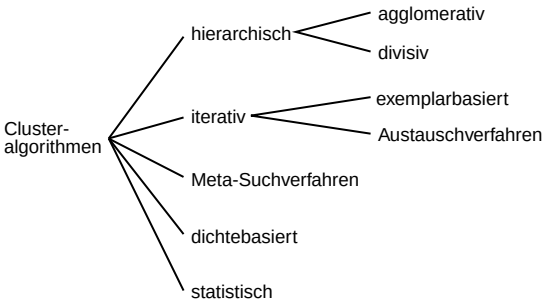

Abbildung 4: Einteilung existierender Cluster-Algorithmen hinsichtlich ihres algorithmischen Prinzips.

Prinzip algorithmisch gut beschreiben läßt, ist er nachfolgend vorgestellt. MajorClust ist ein heuristisches Verfahren zur Maximierung von Λ, dem gewichteten partiellen Zusammenhang eines Ähnlichkeitsgraphen G.

Definition (Λ) *Sei $\mathcal{C} = \{C_1, \ldots, C_k\}$ ein Clustering der Knotenmenge V eines gewichteten Graphen $G = \langle V, E, \varphi \rangle$.*

$$\Lambda(\mathcal{C}) := \sum_{i=1}^{k} |C_i| \cdot \lambda_i,$$

wobei λ_i den gewichteten Kantenzusammenhang des induzierten Subgraphen $G(C_i)$ bezeichne. Der gewichtete Kantenzusammenhang λ eines Graphen $G = \langle V, E, \varphi \rangle$ ist als $\min \sum_{\{u,v\} \in E'} \varphi(u, v)$ definiert, mit der Bedingung, dass $G' = \langle V, E \setminus E' \rangle$ nicht zusammenhänged ist. λ wird auch als die Kapazität eines minimalen Cuts von G bezeichnet.

Zu Anfang entspricht jeder Knoten seinem eigenen Cluster. In den nachfolgenden Reorganisationsschritten wird ein Knoten v demjenigen Cluster zugeordnet, der augenblicklich die höchste Kantengewichtssumme bezüglich v besitzt (siehe auch Abbildung 5). Der Algorithmus terminiert, wenn kein Knoten mehr seine Cluster-Zugehörigkeit ändert.

Algorithmus. MajorClust.
Input. Graph $G = \langle V, E, \varphi \rangle$.
Output. Funktion $c : V \to \mathbf{N}$, die Knoten auf Cluster abbildet.

(1) $n = 0$, $t =$ *false*
(2) $\forall v \in V$ **do** $n = n + 1$, $c(v) = n$ **end**
(3) **while** $t =$ *false* **do**
(4) $\quad t =$ *true*
(5) $\quad \forall v \in V$ **do**
(6) $\quad\quad c^* = \underset{i,\ i=1,\ldots,n}{\operatorname{argmax}} \left(\sum_{\substack{\{u,v\} \in E \\ \wedge\ c(u)=i}} \varphi(u, v) \right)$
(7) $\quad\quad$ **if** $c(v) \neq c^*$ **then** $c(v) = c^*, t =$ *false*
(8) $\quad$ **end**
(9) **end**

2.4 Clustern mittels Suffix-Bäumen

Suffix-Baum-Clustern, im Folgenden mit STC abgekürzt, unterscheidet sich grundlegend von den vektorraumbasierten Ansätzen [8]. STC definiert Dokumentähnlichkeit mittels Wort- bzw. Satzüberdeckungen und Wortreihenfolge. Folgende Eigenschaften liegen dem STC zugrunde:

1. Je mehr Satzfragmente zwischen Dokumenten übereinstimmen, desto ähnlicher sind die Dokumente.
2. Die natürliche Reihenfolge der Worte innerhalb eines Dokumentes ist wichtig für die Ähnlichkeitsberechnung.
3. Ein Dokument kann mehrere Themen beinhalten und kann daher in mehreren Clustern auftauchen.

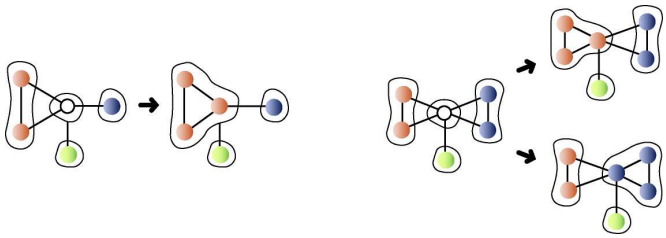

Abbildung 5: Illustration von Schritt 6 und 7 in MajorClust: Eine definite Mehrheitsentscheidung (links) und ein Unentschieden (rechts) bei der Zuweisung des mittleren Knotens zu einem Cluster.

Diese Eigenschaften können mittels eines Suffix-Baumes implementiert werden. Hierfür wird ein Dokument als Folge von Worten $\mathbf{d} = w_1 \ldots w_m$ betrachtet. Der i-te Suffix eines Dokumentes ist diejenige Teilfolge von $\mathbf{d}$, die mit dem Wort w_i anfängt. Ein Suffix-Baum einer Wortfolge ist ein Baum, der jeden Suffix der Wortfolge auf einem Pfad enthält, wobei die Kanten des Pfades mit den Worten des Suffix markiert sind. Die Konstruktion eines Suffix-Baumes geschieht wie folgt: Bei der Einfügung des i-ten Suffix in den Baum wird geprüft, ob eine Kante mit der Markierung w_i inzident zur Wurzel ist. In diesem Fall wird die Kante traversiert, und es wird geprüft, ob eine nachfolgende Kante die Markierung w_{i+1} trägt usw. Trifft man auf einen Knoten ohne eine passend markierte Kante, so wird ein neuer Knoten als Nachfolger generiert und mit einer entsprechend markierten Kante verbunden. Abbildung 6 zeigt einen Suffix-Baum für drei kleine Dokumente mit den Inhalten „boy plays chess“, „boy plays bridge too“ und „father plays chess too“.
Offensichtlich realisiert diese Struktur die eingangs beschriebenen Eigenschaften. Z. B. beinhaltet Knoten a („boy plays“) eine Teilwortfolge der ersten beiden Dokumente; Knoten c („chess“) steht für Dokument eins und drei, und Knoten b („plays“) enthält eine Teilwortfolge aller Dokumente. In der STC-Terminologie werden Knoten, die Teilwortfolgen von mehr als einem Dokument enthalten, Basis-Cluster (base cluster) genannt. Abbildung 6 zeigt die fünf Basis-Cluster der drei Dokumente.
Aus der Menge der Basis-Cluster werden nun die größten ausgewählt. Sie entsprechen den Knoten eines Ähnlichkeitsgraphen, wobei zwei Basis-Cluster mit einer Kante verbunden werden, falls sie mehr als die Hälfte ihrer Dokumente gemeinsam haben (siehe Abbildung 7). Die Zusammenhangskomponenten dieses Graphen bilden die gesuchten Kategorien.

3 Angrenzende Fragestellungen

Cluster-Algorithmen sind die Grundlage der meisten Kategorisierungsansätze. Die Auswahl eines Cluster-Algorithmus und seine Parametrisierung ist jedoch eine Kunst für sich: Der Algorithmus muß effizient sein, und natürlich sollen die generierten Cluster möglichst gut die menschliche Vorstellung einer Kategorisierung nachbilden. Weiterhin benötigt man Verfahren, um kurz und treffend den Inhalt eines Clusters zusammenfassen und ihn hinsichtlich anderer Cluster abgrenzen. Beide Aspekte sind Gegenstand aktueller Forschung und werden in diesem Abschnitt näher beleuchtet.

3.1 Validierung und Benchmarks

Hierarchische, exemplarbasierte oder dichtebasierte Cluster-Algorithmen werden auf breiter Basis zur Dokumentkategorisierung eingesetzt [9, 10]. Obwohl die Leistungsfähigkeit der Algorithmen signifikant von ihren Parametern abhängt (siehe Tabelle 2),

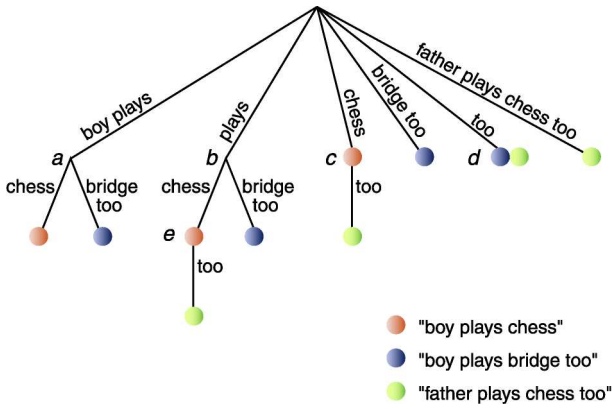

Abbildung 6: Ein Suffix-Baum für die Dokumente „boy plays chess“, „boy plays bridge too“ und „father plays chess too“. Die eingefärbten Knoten repräsentieren die Dokumente, die den Suffix beinhalten, der an dem jeweiligen Knoten endet.

ist deren automatische Bestimmung im Zusammenhang mit der Dokumentkategorisierung wenig erforscht. Eine Möglichkeit, um geeignete Parameter zu finden, sind Clustering-Qualitätsmaße. Sie bilden – gemäß ausgewählter Eigenschaften – ein Clustering auf eine reelle Zahl ab. Hat man nun eine Menge von Clusterings, die mit verschiedenen Parametern erzeugt wurden, so lässt sich auf Basis des Qualitätsmaßes eine Güterangfolge definieren.
Was macht ein gutes Clustering aus? Traditionelle Qualitätsmaße wie der Dunn-Index oder der Davies-Bouldin-Index definieren Qualität mittels Varianzen innerhalb von Clustern, Distanzen zwischen Clustern und Distanz-Durchmesser-Verhältnissen. Diese Eigenschaften beziehen sich auf die strukturelle Natur der Cluster und sollen deren Homogenität und Separierung messen. Es bleibt die Frage, ob ein vom Menschen erstelltes Clustering einen hohen Wert unter einem solchen Qualitätsmaß erzielt – oder anders gefragt, ob es ein Clustering-Qualitätsmaß gibt, das dem menschlichen Informationsbedürfnis (information need) im Dokumenten-Retrieval Rechnung tragen kann.
Abbildung 8 zeigt ein Streudiagramm, in dem jeweils die Güte eines Clusterings bezüglich einer menschlichen Kategorisierung sowie der zugehörige Dunn-Index-Wert gegenübergestellt sind. Schwächen im Dunn-Index zeigen sich in den eingekreisten Clusterings: Die alleinige Orientierung an diesem Index führt nicht zur Auswahl des besten Clusterings. Grundlage solcher und ähnlicher Analysen sind von Menschen erstellte Testkollektionen; Tabelle 3 nennt einige bekannte.
Es wird nun ein graphbasiertes Qualitätsmaß vorgestellt, das keine expliziten Abstands- und Streukriterien verwendet [11]. Dazu folgende Vorüberlegung: Ein Graph $G = \langle V, E\rangle$ heißt dünn, falls $|E| = \Theta(|V|)$; er heißt dicht, falls $|E| = \Theta(|V|^2)$. D. h., man kann ein Maß für die Dichte θ eines Graphen aus der Gleichung $|E| = |V|^\theta$ ableiten. Für gewichtete Graphen $G = \langle V, E, \varphi\rangle$ lässt sich diese Beobachtung verallgemeinern:

$$\varphi(G) = |V|^\theta \;\Leftrightarrow\; \theta = \frac{\ln(\varphi(G))}{\ln(|V|)},$$

wobei $\varphi(G) := |V| + \sum_{e\in E}\varphi(e)$. Offensichtlich kann θ dazu

Cluster-Algorithmen	Parameter
exemplarbasiert	Anzahl von Kategorien, initiale Wahl von Stellvertretern, initiale Clusterings
hierarchisch	Agglomerations- bzw. Divisionslevel
dichtebasiert	Dichteschwellwert

Tabelle 2: Einige Parameter von Cluster-Algorithmen.

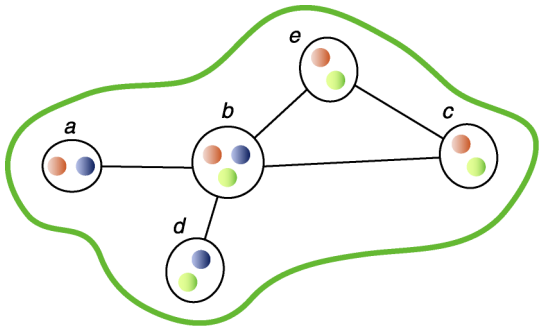

Abbildung 7: Der Ähnlichkeitsgraph für die Basis-Cluster aus Abbildung 6. Da es nur *eine* Zusammenhangskomponente gibt, werden alle Basis-Cluster zu einer Kategorie zusammengefasst. Wäre das Wort „plays“ auf einer Stoppwortliste, würden drei Kategorien entstehen, da der mittlere Basis-Cluster – und mit ihm seine vier inzidenten Kanten – verworfen würden.

benutzt werden, um die Dichte eines induzierten Subgraphen $G' = \langle V', E', \varphi'\rangle$ von G anzugeben: G' ist dünn (dicht) im Vergleich zu G, falls $\varphi(G')/|V'|^\theta$ kleiner (größer) als 1 ist. Diese Überlegung führt zum Qualitätsmaß der erwarteten Dichte, $\bar\rho$.

Definition (erwartete Dichte) *Sei $\mathcal{C} = \{C_1, \ldots, C_k\}$ ein Clustering der Knotenmenge V von $G = \langle V, E, \varphi\rangle$, und sei $G_i = \langle V_i, E_i, \varphi_i\rangle$ der von G induzierte Subgraph bezüglich C_i. Dann ist die erwartete Dichte eines Clusterings $\mathcal{C}$ wie folgt definiert:*

$$\bar\rho(\mathcal{C}) = \sum_{i=1}^{k} \frac{|V_i|}{|V|} \cdot \frac{\varphi(G_i)}{|V_i|^\theta}, \quad \textit{wobei } |V|^\theta = \varphi(G)$$

Experimente mit dem Reuters-Korpus RCV1 bestätigen die Robustheit dieses Maßes [11]; exemplarisch ist die Performance im Streudiagramm in Abbildung 9 dargestellt.

3.2 Cluster-Benennung

Die gefundenen Cluster sind mit wenigen und für ihren Inhalt treffenden Worten zu benennen, um einem Anwender einen Überblick über identifizierte Themen zu geben. Für eine Cluster-Benennung wünscht man sich unter anderem die folgenden Eigenschaften: verschiedene Cluster sollen verschiedene Bezeichnungen erhalten, die Bezeichnungen sollen zwischen den Clustern diskriminieren und die Bezeichnungen sollen möglichst genau den Inhalt der ihnen zugeordneten Dokumenten wiedergeben. Ein formaler Rahmen mit wichtigen Eigenschaften einer Cluster-Benennung ist in [12] beschrieben.
Aktuelle Ansätze zum Benennen von Kategorien und Unterkategorien in Kategoriebäumen benutzen unter anderem den χ^2-Test: Jedem Blatt des Kategoriebaumes wird die Vereinigung aller Worte der assoziierten Dokumente zugewiesen; jeder innere Knoten erhält die Vereinigung der Worte seiner Söhne. Ausgehend von der Wurzel wird für jedes Wort w ein χ^2-Test für die Hypothese „Die Wahrscheinlichkeit, dass w in einem der Nachfolger auftaucht, ist

Name	Organisation	Inhalt	#Dok.	#Kat.
Reuters-21578	flach	Nachrichten	10,000	90
Reuters RCV1	hierarchisch	Nachrichten	800,000	120
Ohsumed	hierarchisch	Medizintexte	230,000	14000
TREC	divers	divers	>20 GB	divers

Tabelle 3: Bekannte Testkollektionen. Die Anzahl von Dokumenten und Kategorien bezieht sich auf den kategorisierten Teil der Kollektion. In den Reuters-Kollektionen kann ein Text zu mehreren Kategorien zugeordnet sein.

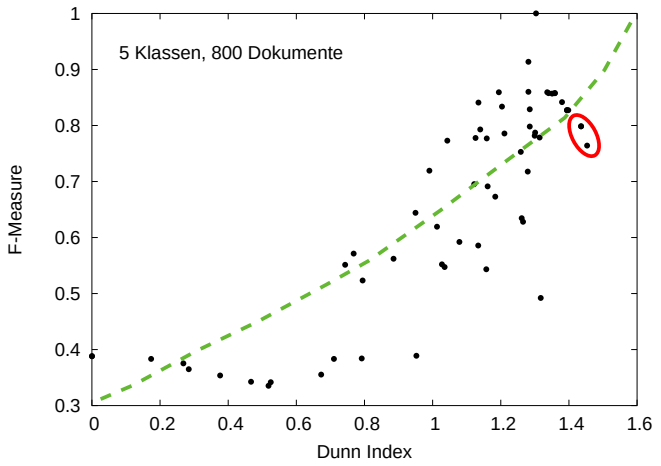

Abbildung 8: Experiment zum Dunn-Index. Die Testkollektion enthält 800 Dokumente, die in fünf Kategorien unterteilt sind. Die Ähnlichkeit jedes Clusterings bezüglich der Referenzkategorisierung wurde mit dem F-Measure quantifiziert und auf der y-Achse aufgetragen, die entsprechenden Werte des Dunn-Index auf der x-Achse. Die eingekreisten Clusterings zeigen kritische Dunn-Index Werte: die zugehörigen F-Measure-Werte sind schlechter als bei vielen anderen Clusterings; der Dunn-Index bewertet diese Clusterings besser, als sie aus Sicht des Menschen sind. Die gestrichelte Kurve zeigt den Verlauf eines idealen Qualitätsmaßes.

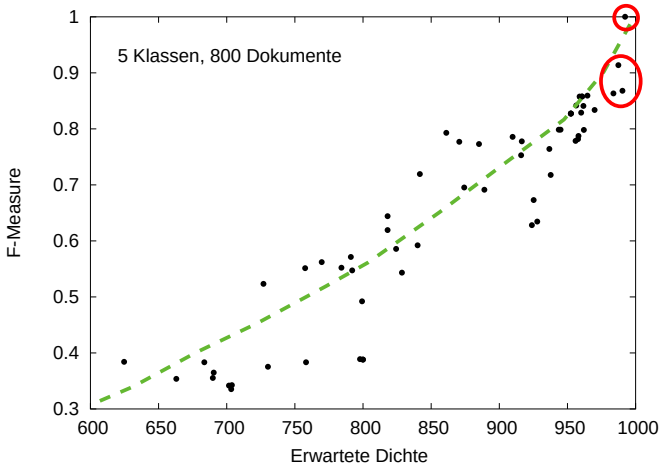

Abbildung 9: Experiment zur erwarteten Dichte $\bar{\rho}$ auf den gleichen Daten wie im Experiment zum Dunn-Index in Abbildung 8. Die vier besten Clusterings werden als solche erkannt, und das Maximum von $\bar{\rho}$ identifiziert hier die ideale (menschliche) Kategorisierung.

gleich.“ durchgeführt. Kann die Hypothese nicht verworfen werden, so wird w als Bestandteil des gemeinsames Vokabulars des aktuellen Unterbaumes interpretiert und aus dem Vokabular aller Nachfolger gestrichen. Die häufigsten Worte aus dem Vokabular eines jeden Knotens bilden dann die Cluster-Benennung.
Weighted Centroid Covering (WCC) ist eine effiziente Methode zur Cluster-Benennung, die auf nicht-hierarchische Kategorisierungen angewandt werden kann. Sie versucht unter Berücksichtigung der Cluster-Größe möglichst fair die wichtigsten Worte im virtuellen Wortvektor des Cluster-Zentroiden zu überdecken [12].

4 Kategorisierende Suchmaschinen

Tabelle 4 zeigt eine Projektübersicht zu kategorisierenden Suchmaschinen. Nur noch wenige der Forschungsprojekte sind online; kommerzielle Projekte dominieren die Szene. Jedes der Projekte folgt unterschiedlichen Paradigmen was Benutzer-Interaktion, Kategoriebildung, Visualisierung und – vom softwaretechnischem Standpunkt aus – Aufteilung von Funktionalität in Client und Server angeht.
Vivísimo (siehe Abbildung 1) ist das bislang erfolgreichste Projekt. Es begann im Jahr 2000 als Forschungsprojekt der Carnegie Mellon Universität und beantwortet mittlerweile über eine Million Anfragen pro Monat. Nach der Formulierung einer Anfrage startet Vivísimo eine Meta-Suche auf konventionellen Suchmaschinen. Obwohl über die zugrundeliegende Cluster-Technologie auf Vivísimos Web-Seiten nichts zu erfahren ist, liegt die Vermutung nahe, dass eine Variante des Suffix-Baum-Clusterns zum Einsatz kommt. Die gefundenen Kategorien werden in einem Kategoriebaum angezeigt: Die Cluster-Benennungen der größten Kategorien sind auf der linken Seite des Browsers dargestellt, während der Inhalt einer Kategorie im rechten Teilfenster als HTML-Liste zu sehen ist. Der Kategoriebaum kann dabei ähnlich wie beim Windows-Explorer expandiert werden: Vivísimo erzeugt serverseitig HTML/JavaScript-Code und überlässt dem Browser das Aktualisieren des Baumes und der HTML-Liste. Vivísimos Cluster-Technologie wird seit 2003 auch von MetaCrawler, WebCrawler und DogPile eingesetzt.
Turbo10 bietet die Option, eine Meta-Suche auf bis zu 10 von insgesamt ca. 1.600 Suchschnittstellen durchzuführen [13]. Die meisten davon sind Datenbank-Frontends; dementsprechend wirbt Turbo10 damit, Informationen aus dem „Deep Web“, also von Standardsuchmaschinen nicht-indizierte Seiten, systematisch nutzbar zu machen. Das Clustern findet asynchron statt: Nachdem die Suchergebnisse an den Browser übermittelt wurden, erstellt JavaScript-Code immer genau 10 Cluster. Jeder dieser Cluster ist mit einem Schlüsselwort bezeichnet und ähnlich wie in Vivísimo dargestellt.
Bei Alsearch (siehe Abbildung 10 und [14]) werden Eingaben während der Anfrageformulierung mit dem SmartSpell-Algorithmus syntaktisch geprüft. Alsearch führt eine Meta-Suche auf mehreren Suchmaschinen aus und analysiert ca. 200 Suchergebnisse hinsichtlich ihrer thematischen Ähnlichkeit. Die gefundenen Kategorien werden durch einen Graphen dargestellt, wobei verwandte Kategorien räumlich näher angeordnet sind, als nicht verwandte. Der Graph ist in einem hyperbolisch verzerrten Raum dargestellt: Sobald eine Kategorie mit der Maus in Richtung Zentrum bewegt

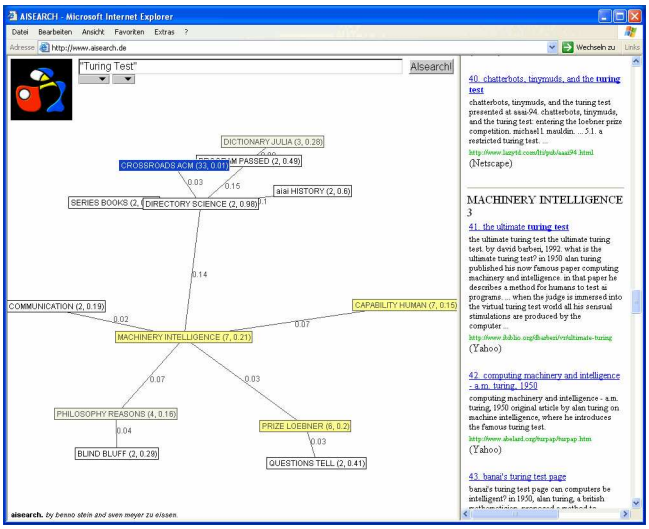

Abbildung 10: Snapshot der Alsearch-Benutzerschnittstelle. Auf der linken Seite ist ein Graph mit den gefundenen Kategorien zu sehen. Ein Klick auf einen Cluster zentriert diesen und stellt seinen Inhalt auf der rechten Seite des Fensters dar.

Name	Stapellauf	Status	Entwicklungs-stadium	Kategorisierungs-methode	Dokumenten-modell	URL
Grouper	1997	inaktiv	Forschung	unüberwacht	Suffix-Baum	keine Demo verfügbar
Oasis	1997	inaktiv	Forschung	unüberwacht	VRM	keine Demo verfügbar
EuroSearch	1998	inaktiv	Forschung	unüberwacht	VRM	keine Demo verfügbar
Yahoo Planet	1998	inaktiv	Forschung	überwacht	VRM	http://www-ai.ijs.si/DunjaMladenic/
ZNow	1998	jetzt Endymion	kommerziell	unüberwacht	unbekannt	http://www.goendymion.com/
Vivísimo	2000	online	kommerziell	unüberwacht	unbekannt	http://www.vivisimo.com
Infonetware	2000	online	kommerziell	unüberwacht	unbekannt	http://www.infonetware.com
Lighthouse	2000	inaktiv	Forschung	unüberwacht	VRM	http://www-ciir.cs.umass.edu/~leouski
KartOO	2001	online	kommerziell	unüberwacht	unbekannt	http://www.kartoo.com
Alsearch	2002	online	Forschung	unüberwacht	VRM	http://www.aisearch.de
WebRat	2002	z. Zeit inaktiv	Forschung	unüberwacht	VRM	http://www.know-center.at/webrat/
Turbo10	2003	online	kommerziell	unüberwacht	unbekannt	http://www.turbo10.com
Mooter	2003	online	kommerziell	unüberwacht	unbekannt	http://www.mooter.com

Tabelle 4: Projekte zur kategorisierenden Dokumentsuche im Internet. Einige der frühen Forschungsprojekte sind inzwischen inaktiv, und es gibt daher keine Demonstrationswebseiten. Viele der kommerziellen Projekte machen keine Angaben über Dokumentmodell oder Cluster-Technologie. Die Abkürzung VRM steht für Vektorraummodell.

wird, bekommt diese und angrenzende Kategorien mehr Platz. Ein Klick auf eine Kategorie zeigt die hierzu gehörige Liste der gefundenen Suchergebnisse.

KartOO holt sich für eine Anfrage zwölf Dokumente von einer Suchmaschine und zeigt diese mit einem Flash-Browser-Plugin an: Jedes Dokument ist durch einen Knoten symbolisiert, der mit der URL des Dokumentes beschriftet ist. Wenn der Mauszeiger sich über eines der dargestellten Themen bewegt, werden Dokumente, die mit dem Thema assoziiert sind, kurzzeitig mit einer Hyperkante verbunden.

Mooter ist ein relativ junges Projekt, das im Jahr 2003 ins Leben gerufen wurde. Nach einer ersten Suchanfrage generiert Mooter ein Bild, in dem gefundene Kategorien sternförmig um die eingegebene Anfrage angeordnet sind. Auf Mausklick erscheint eine neue Seite, die nur die Unterkategorien der gewählten Kategorie auf der linken und eine Linkliste für den Inhalt einer selektierten Unterkategorie auf der rechten Seite zeigt.

WebRat gruppiert die Ergebnisse einer Meta-Suche und stellt das Clustering in einer „Wolkenlandschaft“ innerhalb eines Java-Applets dar [15]. Für jeden gefundenen Cluster entsteht eine mit Schlüsselworten beschriftete Wolke, deren Größe sich nach der Dokumentanzahl richtet. Der Abstand zwischen den Wolken spiegelt die Cluster-Ähnlichkeit wider.

LightHouse benutzt ebenfalls eine Meta-Suche, um Wortvektoren aus gefundenen Treffern zu erstellen [16]. Interessanterweise werden hier die Wortvektoren nicht geclustert – ihre Abstandsmatrix dient als Eingabe für einen MDS-Algorithmus, der ein Layout gemäß der Ähnlichkeit im zwei- oder dreidimensionalen Raum erstellt: Je ähnlicher sich Dokumente sind, desto geringer ist ihre Distanz. Jedes Dokument wird durch eine Kugel dargestellt, und das Clustern obliegt dem Auge des Betrachters.

Die Webseite von Infonetware ist eine Demonstration der Cluster-Technologie von Infogistics. Vergleichbar zu Vivísimo werden Meta-Suchergebnisse hierarchisch in Kategorien organisiert. Die Strategie der Kategorisierungstechnologie (RealTerm) besteht aus vier Schritten: Rechtschreibkorrektur, Identifikation wichtiger Phrasen, Clustering und Hierarchiebildung auf Basis von Wortassoziationen.

Referenzen[1]

[1] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3), 1980.

[2] G. Salton. *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley, 1988.

[3] N. Fuhr. Models for Retrieval with Probabilistic Indexing. *Information Processing and Management*, 25(1),1989.

[4] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 2002.

[5] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of ICML-97, 14th International Conference on Machine Learning*, 1997.

[6] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of KDD96*, 1996.

[7] B. Stein and O. Niggemann. On the Nature of Structure and its Identification. In *Graph-Theoretic Concepts in Computer Science*, volume 1665 of LNCS, 1999.

[8] O. Zamir and O. Etzioni. Web Document Clustering: A Feasibility Demonstration. In *Proceedings of the 21st SIGIR Conference*, 1998.

[9] N. Fuhr, N. Gövert, M. Lalmas, and F. Sebastiani. Categorisation tool, final prototype. Technical report, University of Dortmund, 1999.

[10] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. Technical Report 00-034, University of Minnesota, 2000.

[11] B. Stein, S. Meyer zu Eißen, and F. Wißbrock. On Cluster Validity and the Information Need of Users. In *Proceedings of AIA 03, Benalmádena, Spain*, 2003

[12] B. Stein and S. Meyer zu Eißen. Topic Identification: Framework and Application. In *Proceedings of I-KNOW 04, Graz*, 2004.

[13] N. Hamilton. The mechanics of a deep net metasearch engine. In *Proceedings of WWW2003*, 2003.

[14] S. Meyer zu Eißen and B. Stein. The Alsearch Meta Search Engine Prototype. *Proceedings of WITS 02, Barcelona, Spain*, 2002.

[15] V. Sabol, W. Kienreich, M. Granitzer, J. Becker, K. Tochtermann, and K. Andrews. Applications of a Lightweight, Web-Based Retrieval, Clustering and Visualisation Framework. In *4th International Conference on Practical Aspects of Knowledge Management*, 2002.

[16] A. Leuski and J. Allan. Lighthouse: showing the way to relevant information. In *2000 IEEE InfoVis '00*, 2000.

[1] Eine ausführliche Literaturliste findet sich in der Web-Version des Artikels auf www.aisearch.de.