Universität Paderborn

Institut für
Informatik

Studienarbeit

AG Wissensbasierte Systeme

Vektorraummodell-basierte versus Suffix-Baum-basierte Kategorisierung

vorgelegt von
Martin Potthast

Paderborn, Januar 2005

Erstprüfer:	Dr. Benno Stein
Zweitprüfer:	Prof. Dr. Gregor Engels
Betreuer:	Dr. Benno Stein Sven Meyer zu Eißen
Abgabedatum:	28.01.2005

Anschrift:	Martin Potthast Padergasse 10 33098 Paderborn
Matrikel:	6154050
Studiengang:	Diplom-Informatik
Fachsemester:	7
E-Mail:	beebop@upb.de

Erklärung:

Ich habe die Arbeit selbständig angefertigt und keine anderen, als die angegebenen und bei Zitaten kenntlich gemachten Quellen und Hilfsmittel verwendet.

Paderborn, 28.01.2005

Abstract: Für die Kategorisierung von Dokumenten gibt es unterschiedliche Paradigmen bezüglich des zugrunde liegenden Modells, nämlich Vektorraummodell-basierte oder auf Suffix-Bäumen basierende. Ein Vektorraummodell-basiertes Kategorisierungsverfahren wird in dieser Ausarbeitung der Suffix-Baum-Kategorisierung gegenübergestellt. Beide werden zunächst allgemein definiert, demonstriert und anschließend theoretisch miteinander verglichen. Dabei zeigt sich zum einen, dass Suffix-Baum-Kategorisierung unvollständig kategorisiert, was bedeutet, dass nicht unbedingt jedes Dokument einer Menge, die mit diesem Verfahren kategorisiert wird, einer Kategorie zugeordnet wird. Darauf aufbauend wird zum anderen gezeigt, dass die Kategorisierungsergebnisse der Suffix-Baum-Kategorisierung unter bestimmten Voraussetzungen durch die der Vektorraummodell-basierten Kategorisierung approximiert werden können. Es wird außerdem gezeigt, dass Suffix-Bäume mittels geeigneter Ähnlichkeitsmaße auch losgelöst von der Suffix-Baum-Kategorisierung als Dokumentmodell verwendet werden können.

Inhaltsverzeichnis

1 Einleitung

Die Kategorisierung von Dokumenten betrifft all jene Problemstellungen, in denen eine große Menge von Dokumenten durch einen Anwender durchsucht oder verarbeitet werden muss. Ziel dabei ist es, die Dokumente ihrem Inhalt entsprechend in Kategorien einzuordnen und so den Anwender bei seiner Arbeit zu unterstützen. Verfahren zur Kategorisierung können in zwei Gruppen eingeteilt werden, nämlich solche, die Kategorien automatisch erzeugen oder solche, die aus einer Menge von bestehenden Kategorien auswählen. Man spricht hier auch von unüberwachter bzw. überwachter Kategorisierung. Der Anwender wird mit ihrer Hilfe in die Lage versetzt, eine weitaus größere Menge von Dokumenten als mit konventionellen Mitteln zu verarbeiten. Dies geschieht allein anhand der Kategorie, in der sie sich befinden, ohne dass jedes Dokument einzeln betrachtet werden muss.

Als zu kategorisierende Dokumente kommen hier alle Arten natürlichsprachlicher Texte in Frage. Sie können unter anderen in Form von Textdateien, E-Mails oder Webseiten vorliegen. Das Format der Dokumente spielt für ihre Kategorisierung keine große Rolle, da sie in der Regel zuvor einer Vorverarbeitung unterzogen werden. Deren Ziel ist unter anderem, die Dokumentstruktur jedes Dokuments zu vereinheitlichen.

Ein Kategorisierungsverfahren lässt sich in zwei Schritte unterteilen, die Modellbildung und die Fusionierung. Zunächst werden die zu kategorisierenden Dokumente auf Grundlage eines Dokumentmodells indiziert. Ein Dokumentmodell erlaubt die Darstellung von Dokumenten auf einer abstrakteren Ebene. Die Dokumente werden darin durch eine Reihe von Merkmalen repräsentiert. Das ermöglicht die weitere Verarbeitung und den Vergleich der Dokumente.

Des Weiteren ist es in einem Dokumentmodell möglich, Aussagen über die Ähnlichkeit zweier Dokumente aufgrund ihrer Merkmale zu treffen. Dabei bezieht sich „Ähnlichkeit“ auf ihre inhaltliche Übereinstimmung. Dazu wird in den meisten Fällen ein Ähnlichkeitsmaß definiert, das einen Wert im Intervall [0;1] errechnet. Liegt der Wert nahe 0, so stimmen die Dokumente inhaltlich kaum, bei einem Wert nahe 1 dagegen sehr stark überein.

Mit Hilfe des Ähnlichkeitsmaßes werden dann, im zweiten Schritt, die im Modell indizierten Dokumente durch einen geeigneten Algorithmus kategorisiert. Das Endergebnis dieser Fusionierung ist eine Menge von Dokumentkategorien, mit der Eigenschaft, dass die Vereinigung aller Kategorien der Ausgangsmenge von Dokumenten entspricht.

Darüber hinaus ist die Generierung einer Benennung für jede Kategorie ein Folgeschritt bei der unüberwachten Kategorisierung von Dokumenten. Sie soll dazu dienen, den Inhalt der in den Kategorien enthaltenen Dokumente kurz und treffend wiederzugeben. Das ermöglicht es einem Anwender dieser Verfahren, der auf der Suche nach Dokumenten zu einem bestimmten Thema ist, im Idealfall, die für ihn interessanten Kategorien aufgrund ihrer Kennzeichnung auszuwählen. Bei der überwachten Kategorisierung ist dies nicht notwendig, da hier aus bereits bestehenden Kategorien, also vom Anwender selbst erstellten, ausgewählt wird.

Ein Anwendungsbereich der Kategorisierung von Dokumenten in Form von Webseiten ist die Suche nach Informationen im Internet mittels Suchmaschinen. Anstatt einer Liste von Ausschnitten aus Webdokumenten, so genannte Schnipsel (engl. Snippets), sollen dem Anwender hier Kategorien als Suchergebnis präsentiert werden. Tatsächlich geschieht die Kategorisierung hier nicht auf Grundlage vollständiger Webseiten, son-

dern vielmehr auf Grundlage der Schnipsel. Suchmaschinen wie AIsearch[1] oder Vivísimo[2] sind dazu beispielsweise in der Lage.

Die vorliegende Ausarbeitung betrifft zwei unüberwachte Kategorisierungsverfahren. Eines davon basiert auf einem Vektorraummodell, das auch in erstgenannter Suchmaschine eingesetzt wird [11]. Verfahren, die auf diesem Dokumentmodell beruhen, sind sehr verbreitet. Bei dem anderen Verfahren handelt es sich um Suffix-Baum-Kategorisierung, ein verhältnismäßig junger Ansatz, von dem angenommen wird, dass er in der zweitgenannten Suchmaschine Verwendung findet [17]. Es basiert auf der Datenstruktur Suffix-Baum als Dokumentmodell.

Die zu klärenden Fragestellungen, die sich insbesondere auf die Dokumentmodelle beider Verfahren beziehen, sind, ob sich das Dokumentmodell Suffix-Baum auch losgelöst von dem im Fusionierungsschritt der Suffix-Baum-Kategorisierung eingesetzten Algorithmus verwenden lässt sowie welcher Ansatz der stärkere in Bezug auf das zu erwartende Ergebnis und die Laufzeit ist.

Zu diesem Zweck beinhaltet das Folgekapitel eine ausführliche Definition und Beispiele zu beiden Ansätzen. Kapitel 3 befasst sich mit einem theoretischen Vergleich beider Verfahren und zieht Schlussfolgerungen aus den Anwendungsergebnissen. Im 4. Kapitel wird die Austauschbarkeit der beiden Dokumentmodelle untereinander durch analytische Betrachtungen diskutiert. Das letzte Kapitel fasst die erarbeiteten Ergebnisse zusammen und gibt einen Ausblick auf mögliche weitere Studien.

[1] www.aisearch.de
[2] www.vivisimo.com

2 Definition

In den folgenden Abschnitten werden die zu vergleichenden Kategorisierungsverfahren gemäß dem im ersten Kapitel vorgestellten Schema Vorverarbeitung, Modellbildung, Fusionierung und Generierung von Benennungen Schritt für Schritt definiert.

Der Schritt der Vorverarbeitung der Dokumente vor der eigentlichen Kategorisierung ist beiden Verfahren gemein. Sie dient der ersten Vereinfachung der Dokumentstruktur. Jedes Dokument enthält Wörter, die in allen natürlichsprachlichen Texten sehr häufig enthalten sind oder ausschließlich dem Satzgefüge dienen, aber keinen Einfluss auf den Inhalt haben. Diese Wörter werden auf einer zuvor definierten Stopp-Wort-Liste verzeichnet und aus den zu kategorisierenden Dokumenten entfernt. Die verbleibenden Wörter werden anschließend in ihre jeweilige Stammform gebracht, so dass unterschiedliche Deklinationen ein und desselben Wortstamms nicht als verschiedene Wörter erkannt werden. Das ist sinnvoll, da andernfalls verschiedene Formen von inhaltlich gleichbedeutenden Wörtern als Diskriminierungskriterien zwischen Dokumenten dienen würden und nicht als Indizien für ihre Ähnlichkeit.

Das Herz eines jeden Kategorisierungsverfahrens sind die Schritte Modellbildung und Fusionierung. Allgemein betrachtet geht es in diesen Schritten darum, Strukturen in den im Modell abgebildeten Objekten zu suchen. Man spricht hier auch von Clustering. Ein Kategorisierungsverfahren basiert also auf Clusteringverfahren.

2.1 Vektorraummodell-basierte Kategorisierung

Das Vektorraummodell (engl. Vector-Space-Model, kurz VSM) basiert auf dem Gedanken, dass Dokumente auch als eine Ansammlung von Wörtern betrachtet werden können. Dabei steht jedes Wort je für einen Eintrag in einem Vektor, so dass ein Dokument als Vektor aus den in ihm enthaltenen Wörtern repräsentiert werden kann. Informationen über die ursprüngliche Reihenfolge der Wörter im Dokument selbst sind in dieser Repräsentationsform nicht enthalten.

Um mehrere Dokumente in ein und denselben Vektorraum abzubilden, müssen die Einträge der jeweiligen Vektoren einander angeglichen werden. Das bedeutet, dass ein Dokumentvektor auch Einträge für Wörter enthalten muss, die im durch ihn repräsentierten Dokument nicht enthalten sind. Zu diesem Zweck kann ein Wörterbuch angelegt werden, dass die Vereinigungsmenge aller in einer Menge von Dokumenten enthaltenen Wörter enthält. Dieses Wörterbuch dient dann als Basis, um einen Dokumentvektor für ein Dokument der Menge aufzustellen.

Um die Einträge in einem Dokumentvektor zu quantifizieren, ist ein Maß notwendig, dass zu jedem Wort einen bestimmten Zahlenwert errechnet. Ein solches Maß heißt Termgewichtsmaß, da es jedem Term ein spezifisches Gewicht zuordnet. Zum Beispiel ist die Verwendung boolescher Werte, die das Vorhandensein eines Wortes im jeweiligen Dokument anzeigen, denkbar.

In Kategorisierungsverfahren, die auf dem Vektorraummodell basieren, ist das Termgewichtsmaß tfidf sehr weit verbreitet, weshalb in der Literatur auch oftmals von tfidf-basierter Kategorisierung (engl. tfidf-based Clustering, kurz TBC) gesprochen wird, wobei implizit die Verwendung eines Vektorraummodells mit diesem Termgewichtsmaß als Dokumentmodell vorausgesetzt wird.

2.1.1 tfidf und Kosinusähnlichkeit im Vektorraummodell

tfidf ist ein Gewichtsmaß, das zur Feststellung der Relevanz eines Wortes in Bezug auf den Inhalt eines Dokuments dient. Je relevanter ein Wort ist, desto größer sein tfidf-Wert. Grundlegend geht es auf Forschungen von H. P. Luhn und G. Salton zurück

[4,10]. tfidf setzt sich aus zwei verschiedenen Ansätzen zur Feststellung der Relevanz eines Wortes zusammen, der Termhäufigkeit (engl. Term-Frequency, kurz *tf*) und der inversen Dokumenthäufigkeit (engl. Inverse-Document-Frequency, kurz *idf*).

Luhn fasst seine Idee wie folgt zusammen: „It is here proposed that the frequency of word occurrence in an article furnishes a useful measurement of word significance." ([4], S. 160). Er beschreibt, dass die Häufigkeit, mit der ein Wort in einem Dokument auftritt, ein Maß dafür ist, inwieweit das Wort den Inhalt des Dokuments repräsentieren kann. Je nachdem, welcher Sachverhalt in einem Dokument geschildert wird, treten jeweils andere Wörter vermehrt auf, da sie in der Argumentation des Sachverhalts benötigt werden. Auf diese Weise ist es möglich, Dokumente, die je unterschiedliche Beschreibungen desselben Sachverhalts enthalten, aufgrund des vermehrten Vorkommens der dazu benötigten Wörter, automatisiert miteinander zu verknüpfen. Die Auswertung solcher inhaltlicher Verknüpfungen ermöglicht die Kategorisierung einer Menge von Dokumenten. Diese Gedanken bilden die Grundlage für den Begriff der Termhäufigkeit.

Definition: Termhäufigkeit
Sei d ein Dokument, |d| seine Länge in Worten, $\forall i \in \{1...|d|\}$: w_i das i-te Wort aus d und w ein Wort. Sei ferner $\delta_{w_i,w} = 1$, falls $w_i = w$ gilt und ansonsten 0. Dann ist

$$tf(w,d) = \sum_{i=1}^{|d|} \delta_{w_i,w}$$

die Termhäufigkeit des Wortes w im Dokument d.

Um die Relevanz eines Wortes in Bezug auf den Inhalt einer Menge von Dokumenten zu ermitteln, wird die Dokumenthäufigkeit (engl. Document-Frequency, kurz *df*) verwendet. Sie ist analog zur Termhäufigkeit wie folgt definiert:

Definition: Dokumenthäufigkeit und inverse Dokumenthäufigkeit
Sei D eine Menge von Dokumenten, $\forall i \in \{1...|D|\}$: d_i das i-te Dokument aus D und w ein Wort. Sei ferner $\delta_{d_i,w} = 1$, falls d_i w enthält und ansonsten 0. Dann ist

$$df(w,D) = \sum_{i=1}^{|D|} \delta_{d_i,w}$$

die Dokumenthäufigkeit und

$$idf(w,D) = \log\left(\frac{|D|}{df(w,D)}\right)$$

die inverse Dokumenthäufigkeit des Wortes w in der Dokumentenmenge D.

Mit steigender Dokumenthäufigkeit eines Wortes in einer Menge von Dokumenten sinkt seine Relevanz in Bezug auf den Inhalt eines einzelnen Dokuments. Zwar würde dieses Wort durch die Termhäufigkeit noch stark gewichtet, jedoch könnte mit seiner Hilfe nicht mehr zwischen Dokumenten der Menge unterschieden werden, die denselben bzw. einen anderen Sachverhalt darlegen. Die Diskriminationskraft des Wortes wäre insgesamt also niedrig.

Beispielsweise hätte das Wort „Atom" in einer Menge von Dokumenten aus dem Themenbereich Chemie wahrscheinlich eine hohe Dokument- und Termhäufigkeit, was bedeutet, dass eine große Anzahl von Dokumenten dieses Wort jeweils sehr häufig beinhaltet. Das ist einleuchtend, da in der Chemie das Wort „Atom" zum allgemeinen

Sprachgebrauch zählt. Es eignet sich damit kaum zur Unterscheidung inhaltlich verschiedener Dokumente. In einer Menge von geisteswissenschaftlichen Dokumenten hätte dieses Wort wahrscheinlich eine niedrige Dokument- und Termhäufigkeit. Gesetzt den Fall, dass es darin einige wenige Dokumente gäbe, deren Termhäufigkeit für „Atom“ hoch wäre, so wäre dieses Wort dagegen bestens geeignet, um diese Dokumente von anderen abzugrenzen.

Von Interesse bei der Kategorisierung von Dokumenten sind hier insbesondere diejenigen Terme, die eine hohe Relevanz, sowohl im Sinne der Term- als auch der Dokumenthäufigkeit, aufweisen, so genannte Indexterme. Zunächst gilt es daher, die beiden Konzepte in Einklang zu bringen. Da die rechnerische Relevanz eines Wortes bei der Dokumenthäufigkeit gegenläufig zur Termhäufigkeit ist, also mit steigender Häufigkeit fällt, wird die inverse Dokumenthäufigkeit verwendet. Sie wird logarithmiert, da kleine Dokumenthäufigkeiten sonst einen zu starken Einfluss auf das Gesamtergebnis hätten. Da der Logarithmus für Beträge kleiner 1 negativ ist, wird anstatt 1 die Anzahl aller Dokumente durch die Dokumenthäufigkeit geteilt. Die Kombination beider Konzepte geschieht dadurch, dass die Termhäufigkeit mit der inversen Dokumenthäufigkeit gewichtet wird, indem beide miteinander multipliziert werden.

Daraus ergibt sich das für die Quantisierung der Einträge von Dokumentvektoren im Vektorraummodell verwendete Termgewichtsmaß tfidf. Ein tfidf-Vektor ist dann ein Dokumentvektor, dessen Einträge mit tfidf berechnet werden.

Definition: tfidf-Vektor
Sei D eine Menge von Dokumenten, d ein Dokument, W die Menge aller in den Dokumenten aus D enthaltenden Wörter und $\forall i \in \{1...|W|\}$*:* w_i *das i-te Wort aus W. Dann ist*

$$\mathbf{d} = (tf(w_1,d) \cdot idf(w_1,D),\ldots,tf(w_{|W|},d) \cdot idf(w_{|W|},D))^T$$

ein tfidf-Vektor, der d in einen |W|-dimensionalen Vektorraum abbildet.

Um den Grad der inhaltlichen Übereinstimmung zweier Dokumente zu ermitteln, wird im Vektorraummodell ein Ähnlichkeitsmaß verwendet. Allgemein handelt es sich dabei um eine Funktion, die zwei Dokumentvektoren auf einen reellen Wert im Intervall [0;1] abbildet. Die am häufigsten im Zusammenhang mit dem Vektorraummodell verwendete Ähnlichkeitsfunktion ist die Kosinusähnlichkeit, die nachfolgend erläutert wird. Jedoch hängt die Wahl des Ähnlichkeitsmaßes auch von dem im Modell verwendeten Termgewichtsmaß ab. Für ein boolesches beispielsweise eignet sich das Jaccard-Maß, das einen Ähnlichkeitswert auf Grundlage der von den Dokumentvektoren geteilten Einträge ermittelt. Andere Maße sind zum Beispiel das Pseudo-Kosinusmaß, das Dice-Maß oder das Überlappungsmaß [2].

Die Kosinusähnlichkeit basiert auf dem aus der linearen Algebra bekannten Skalarprodukt zweier Vektoren. Das Skalarprodukt selbst ist bereits ein Ähnlichkeitsmaß, dessen Werte allerdings auch größer als 1 werden können. Die Ähnlichkeit zweier Vektoren steigt mit der Anzahl der Merkmale, für die das verwendete Termgewichtsmaß in beiden Vektoren einen von 0 verschiedenen Wert errechnet, und deren Größe.

Hier liegt auch die Problematik der alleinigen Verwendung des Skalarprodukts als Ähnlichkeitsmaß. Je länger ein Dokument ist, desto größer ist auch die errechnete Ähnlichkeit zu einem anderen Dokument, da die Anzahl der Merkmale oder deren Wert steigt. Anschaulich ausgedrückt, steigt die rechnerische Ähnlichkeit zweier Dokumente mit der euklidischen Länge der verrechneten Vektoren. So ist es zum Beispiel möglich, die Ähnlichkeit zweier Dokumente zu steigern, indem eines der Dokumente vervielfacht wird. Das würde der Multiplikation des entsprechenden Dokumentvektors mit einem

Skalar entsprechen und folglich seiner Verlängerung. Die inhaltliche Übereinstimmung ist allerdings durch die Vervielfachung des Ausgangsdokuments nicht gestiegen.

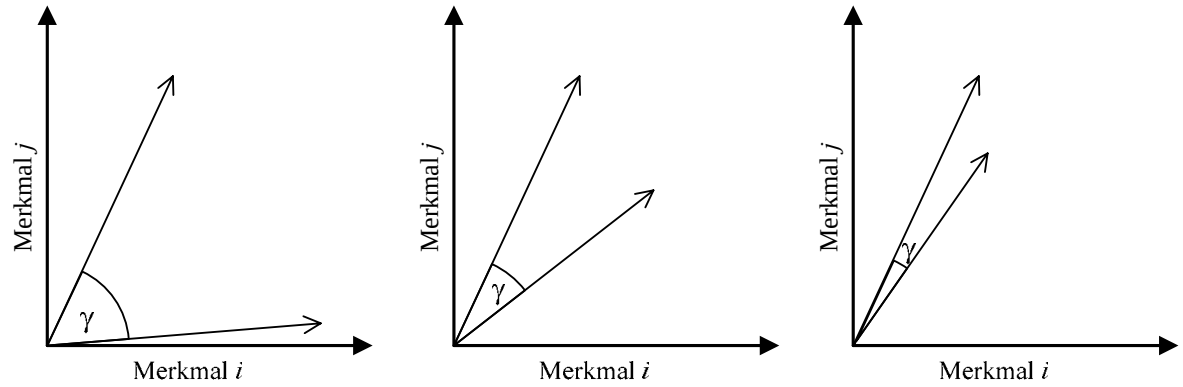

Abbildung 1: Illustration der Kosinusähnlichkeit. Dargestellt werden zwei Dokumentvektoren, die je zwei Einträge *i* und *j* aufweisen. Der Kosinus des Winkels γ zwischen ihnen quantifiziert ihre Ähnlichkeit.

Aus diesem Grund werden bei dem hier vorgestellten Ähnlichkeitsmaß die Dokumentvektoren normiert, was zur Folge hat, dass die Länge eines Dokumentvektors keinen Einfluss mehr auf die rechnerische Ähnlichkeit hat. Diese Berechnung, also das Skalarprodukt zweier normierter Vektoren, entspricht dabei dem Kosinus des Winkels zwischen ihnen. Aus diesem Grund heißt dieses Ähnlichkeitsmaß auch Kosinusähnlichkeit (engl. Cosine-Similarity, kurz φ). Abbildung 1 zeigt beispielhaft drei Diagramme, in denen Dokumentvektoren dargestellt werden, die eine kleine, mittlere bzw. große Ähnlichkeit aufweisen.

Definition: Kosinusähnlichkeit
Sei D eine Menge von Dokumenten und seien d_i und d_j Dokumente mit den zugehörigen Dokumentvektoren $\mathbf{d}_i$ und $\mathbf{d}_j$. Dann ist

$$\varphi(\mathbf{d}_i, \mathbf{d}_j) = \frac{\langle \mathbf{d}_i, \mathbf{d}_j \rangle}{||\mathbf{d}_i|| \cdot ||\mathbf{d}_j||}$$

ein Maß für die inhaltliche Übereinstimmung von d_i und d_j.

2.1.2 Fusionierung mittels Ähnlichkeitsgraphen

Es gibt zahlreiche Algorithmen, die der Verarbeitung der aus einem tfidf-basierten Vektorraummodell gewonnenen Informationen dienen. Sie unterteilen sich unter anderen in Klassen wie hierarchische, iterative und dichtebasierte Algorithmen [13]. Das hier vorgestellte Verfahren ist Single-Link-Clustering, kurz SLC, das zu den hierarchischen Kategorisierungsverfahren zählt. Gegenüber heute üblichen Kategorisierungsverfahren liefert dieses in der Regel qualitativ schlechtere Kategorisierungen. Verfahren, wie zum Beispiel das dichtebasierte MajorClust, schneiden in Evaluierungen hier deutlich besser ab [6].

Dennoch fiel die Wahl auf SLC, da das an dieser Stelle verwendete Verfahren für die im Folgenden gewonnenen Erkenntnisse nicht von Belang ist. Es erschien daher sinnvoll, ein Verfahren vorzustellen, das dem in Suffix-Baum-Kategorisierung verwendeten entspricht.

Es ist graphbasiert, was bedeutet, dass zur Erzeugung von Kategorien ein Graph aufgestellt und ausgewertet werden muss. Der Graph heißt Ähnlichkeitsgraph und ist wie folgt definiert:

Definition: Ähnlichkeitsgraph
Sei D eine Menge von Dokumenten und $\forall i \in \{1...|D|\}$*:* d_i *die i-te Dokument aus D. Sei* $\alpha \in [0;1]$ *ein Schwellwert. Ein Ähnlichkeitsgraph G = (D, E) ist ein ungerichteter Graph, für den D die Knotenmenge ist und für die Kantenmenge* $E = \{\{d_i,d_j\} \mid d_i \neq d_j \wedge sim(d_i,d_j,\alpha) = 1\}$ *gilt.*

Die Knoten dieses Graphen sind Dokumente. Die Kanten eines Ähnlichkeitsgraphen werden über eine Kantenfunktion *sim* bestimmt, die das Vorhandensein einer Kante zwischen zwei Dokumenten berechnet. Die in SLC verwendete *sim*-Funktion basiert auf einem Ähnlichkeitsmaß für Dokumentvektoren, zum Beispiel der Kosinusähnlichkeit, und ist wie folgt definiert:

$$sim_{SLC}(d_i,d_j,\alpha) = \begin{cases} 1 & \varphi(\mathbf{d}_i,\mathbf{d}_j) > \alpha \\ 0 & sonst \end{cases}$$

Der Schwellwert α erlaubt die Einstellung der Dichte im Ähnlichkeitsgraphen. Wird ein Wert gewählt, der nahe 0 liegt, so wird der Graph sehr dicht sein, da auf diese Weise auch Knoten, respektive Dokumente, miteinander verbunden werden, die inhaltlich nur wenig Ähnlichkeit besitzen. Entgegengesetzt dazu führt ein Wert nahe 1 dazu, dass der Graph dünn ist, da nur sehr ähnliche Dokumente miteinander verbunden werden. Welcher Wert für α sinnvoll ist, hängt stark von den zugrunde liegenden Dokumenten ab. Im Allgemeinen ist α so zu wählen, dass der Ähnlichkeitsgraph eine mittlere Dichte aufweist. Das hat zur Folge, dass die durch SLC erzeugte Kategorisierung weder zu ungenau noch zu detailliert ist. Die konkrete Belegung von α spielt hier aber keine große Rolle.

SLC ermittelt die endgültigen Kategorien für eine Dokumentenmenge nach der Aufstellung eines Ähnlichkeitsgraphen durch die Berechnung aller darin enthaltenen Zusammenhangskomponenten. Jede Zusammenhangskomponente ist Grundlage für eine Kategorie, die aus der Vereinigung aller Knoten beziehungsweise Dokumente der Komponente besteht. Die Vereinigung aller endgültigen Kategorien ergibt dabei die Ausgangsmenge der Dokumente. Die Idee dieses Kategorisierungsverfahrens ist, dass, respektive des Schwellwerts α, alle Dokumente im Ähnlichkeitsgraph, die miteinander verknüpft sind, eine ausreichende inhaltliche Übereinstimmung aufweisen, so dass sie in einer Kategorie zusammengefasst werden können.

Algorithmisch betrachtet gibt es zwei grundlegend verschiedene Ansätze, um mit SLC zu kategorisieren, nämlich agglomerative und divisive. Erstere stellen Knotenmenge des Ähnlichkeitsgraphen auf und fügen schrittweise Kanten zwischen ihnen ein, bis die Ähnlichkeitswerte unterhalb des Schwellwertes liegen oder, als weitere mögliche Abbruchbedingung, eine bestimmte Anzahl von Zusammenhangskomponenten erzeugt wurde. Letztere hingegen gehen von einem Ähnlichkeitsgraphen aus, in dem jeder Knoten mit allen übrigen verknüpft ist, und entfernen schrittweise Kanten, bis nur noch Dokumente verbunden sind, die einen Ähnlichkeitswert oberhalb des Schwellwerts haben.

2.1.3 Generierung von Kategoriebezeichnern

Für die Erzeugung von Kategoriebezeichnern gibt es unterschiedliche Ansätze [12]. Sie lassen sich beispielsweise in Verfahren unterteilen, die die Dokumentstruktur analysieren bzw. auf dem Dokumentinhalt basieren. Die Struktur eines Dokuments, also Titel, Abschnittsüberschriften und ähnliches, vermag Aufschluss über seinen Inhalt zu geben. Die Analyse des Inhalts eines Dokuments, also in diesem Fall hauptsächlich der Wörter, aus denen es besteht, zum Beispiel durch die Berechnung der Indexterme, bietet einen anderen Ansatz. Kombinationen hieraus sind ebenfalls möglich.

Inhaltsbezogene Ansätze lassen sich darüber hinaus anhand der Art der Kategorien unterteilen, für die Bezeichner generiert werden sollen. Das sind einerseits solche Kategorisierungen, deren Kategorien nach einem einzigen Entscheidungskriterium erstellt wurden (engl. monothetic clusters) und andererseits solche, deren Kategorien auf verschiedenen Kriterien gleichzeitig beruhen (engl. polythetic clusters). Erstere sind algorithmisch schwer zu erzeugen, weshalb die meisten Kategorisierungsverfahren, so auch TBC, zu letzterem tendieren.

Eingesetzt werden unter anderen Verfahren, die auf Bayes-Klassifizierern, Neuronalen Netzen oder tfidf basieren [7]. Hier jedoch wird ein Suffix-Baum verwendet, um Bezeichner für eine Menge von Dokumenten zu generieren. Dieser Ansatz wurde bisher nur in Suffix-Baum-Kategorisierung selbst verwendet, kann aber auch losgelöst davon angewandt werden. Darüber hinaus sorgt die Verwendung von Suffix-Bäumen in TBC für eine bessere Vergleichbarkeit beider Verfahren.

Im Anschluss an die Fusionierung in TBC wird für jede Kategorie ein Suffix-Baum erzeugt, in dem jedes Dokument der jeweiligen Kategorie indiziert wird. Da alle Dokumente einer Kategorie nach Voraussetzung eine hinreichende Ähnlichkeit aufweisen, gibt es wenigstens einige Wörter, die von den Dokumenten geteilt werden. Nicht zwangsläufig, aber möglicherweise teilen die Dokumente überdies auch Wortketten, was durch die Indizierung in einem Suffix-Baum automatisch erkannt wird. Der entsprechende Suffix-Baum enthält daher eine Reihe innerer Knoten. Durch die Ermittlung der Menge aller von den Dokumenten geteilten Wörter oder Wortketten, die sich aus der Verknüpfung der Kantenbeschriftungen auf den Pfaden ergeben, die zu inneren Knoten des Suffix-Baums führen, werden mögliche Bezeichner für die Kategorie gewonnen. Eine Auswahl daraus, die sich insbesondere an der Länge der Wortketten orientieren kann, wird dann als Bezeichner der Kategorie gewählt. Auf diese Weise gelingt es, ähnlich aussagekräftige Bezeichner zu erzeugen wie Suffix-Baum-Kategorisierung.

2.1.4 Laufzeit

Die Laufzeit von TBC ist die Summe der Laufzeiten der vier für die Kategorisierung notwendigen Einzelschritte Vorverarbeitung, Aufstellung des Dokumentmodells, Fusionierung und Generierung von Kategoriebzeichnern. Sei n die Anzahl der Dokumente, die kategorisiert werden sollen.

Die Vorverarbeitung betrachtet jedes Wort eines Dokuments einzeln. Es wird vorausgesetzt, dass die Identifikation eines Wortes als Stoppwort bzw. andernfalls die Umformung in seine Stammform jeweils in konstanter Zeit abläuft. Die Dokumente können also sequentiell, Wort für Wort, durchlaufen werden. Damit steigt die Vorverarbeitungszeit für ein Dokument linear mit der Anzahl der Wörter, aus denen es besteht. Unter der Voraussetzung, dass alle Dokumente eine konstante Länge aufweisen, steigt die Laufzeit der Vorverarbeitung linear mit der Anzahl der Dokumente, also in $O(n)$.

Für die Aufstellung des tfidf-Vektorraummodells ist in erster Linie das Zählen aller in den Dokumenten vorkommenden Wörter notwendig. Dazu reicht ein sequentieller Durchlauf des Dokuments aus. Die Menge aller in den Dokumenten vorkommenden Wörter bedingt dabei die Anzahl der Dimensionen der tfidf-Vektoren. Sie kann aller-

dings auch als konstant groß angenommen werden, da statistisch gesehen nicht alle Wörter gleich häufig Verwendung finden. Alles in allem resultiert eine Laufzeit von *O(n)* für die Aufstellung des Dokumentmodells.

Während der Fusionierung entspricht jedes Dokument einem Knoten. Für die Aufstellung des Ähnlichkeitsgraphen müssen alle mit allen verglichen werden. Die Ermittlung aller Zusammenhangskomponenten kann dann zum Beispiel mittels Tiefensuche geschehen. Es ergibt sich also eine Laufzeit von $O(n^2)$.

Für die Generierung von Kategoriebezeichnern werden für alle Kategorien Suffix-Bäume erzeugt. Ein Suffix-Baum für ein Dokument kann linear in seiner Länge erzeugt werden. Bei konstant langen Dokumenten ergibt sich eine Initialisierungszeit, die linear mit der Größe einer Kategorie steigt. Da die Kategorien nicht überlappend sind, resultiert *O(n)* für die Erzeugung von Suffix-Bäumen für alle Kategorien. Aus allen Suffix-Bäumen müssen nun die möglichen Bezeichner extrahiert werden. Da die Anzahl der Knoten in einem Suffix-Baum ebenfalls linear mit der Anzahl der Dokumente steigt und ferner maximal *n* Suffix-Bäume erzeugt werden, gelingt auch dieser Schritt in *O(n)*.

2.1.5 Beispiel zu Vektorraummodell-basierter Kategorisierung

Zur Veranschaulichung von TBC werden drei Dokumente zunächst durch tfidf-Vektoren in ein Vektorraummodell projiziert und anschließend mit SLC kategorisiert. Bei den Dokumenten handelt es sich jeweils um ein Zitat einer berühmten Persönlichkeit. Sie sind in Tabelle 1 gegenübergestellt.

Dokument d_1:	Dokument d_2:	Dokument d_3:
„Manche Menschen haben einen Gesichtskreis vom Radius Null und nennen ihn ihren Standpunkt.“ David Hilbert (1862 – 1943)	*„Der Horizont vieler Menschen ist ein Kreis mit Radius Null und das nennen sie ihren Standpunkt.“* Albert Einstein (1879 – 1955)	*„Ein Gesichtspunkt ist ein geistiger Horizont mit dem Radius Null.“* Bertrand Russel (1872 – 1970)

Tabelle 1: Beispieldokumente[3] zur Demonstration von Kategorisierungsverfahren.

Zunächst erfolgt die Vorverarbeitung der Dokumente. Es werden also aus allen Dokumenten eventuell vorhandene Zeichensetzung und diejenigen Wörter entfernt, die auf einer Stopp-Wort-Liste[4] verzeichnet sind. Tabelle 2 zeigt die Dokumente nach der Vorverarbeitung.

Dokument d_1:	Dokument d_2:	Dokument d_3:
Manche Menschen haben Gesichtskreis Radius Null nennen Standpunkt	Horizont vieler Menschen Kreis Radius Null nennen Standpunkt	Gesichtspunkt geistiger Horizont Radius Null

Tabelle 2: Dokumente aus Tabelle 1 nach der Vorverarbeitung.

[3] Leider war der tatsächliche Urheber dieser, oder eines sinnverwandten Ausspruchs nicht zu ermitteln. Allerdings erscheint das von R. Merton geprägte Matthäus-Prinzip (Science, 56. Jg. 1968, Heft 159) zuzutreffen, das besagt, dass ein wissenschaftlicher Ausspruch immer dem berühmtesten aller wahrscheinlichen Kandidaten zugeschrieben wird. Merton bezieht sich dabei auf eine Bibelstelle, in der es heißt: „Denn wer da hat, dem wird gegeben werden, und er wird die Fülle haben; wer aber nicht hat, dem wird auch, was er hat, genommen werden“ (Matthäus, Kap. 25,29) [19].

[4] Eine solche Stopp-Wort-Liste für - unter anderen - deutschsprachige Wörter ist unter www.ranks.nl/stopwords/ zu finden.

Als nächstes muss für jedes Wort eines Dokuments die Termhäufigkeit und für die Menge aller Wörter jeweils die Dokumenthäufigkeit ermittelt werden. Exemplarisch für das Wort „Menschen“ ergibt sich hier für Dokument d_1 und d_2 je ein *tf*-Wert von 1 und für Dokument d_3 ein *tf*-Wert von 0. Die Dokumenthäufigkeit von „Menschen“ ist demzufolge 2, da das Wort in zwei Dokumenten enthalten ist. Für „Radius“ hingegen ist der *tf*-Wert in jedem Dokument 1 und es ergibt sich damit ein *df*-Wert von 3. Analog werden die *tf*- und *df*-Werte für alle übrigen Wörter berechnet. Daraus werden die tfidf-Vektoren ermittelt. Tabelle 3 zeigt diese Vektoren für alle Dokumente der Menge $D = \{d_1,d_2,d_3\}$.

Dimensionen:	tfidf-Vektor $\mathbf{d}_1$:	tfidf-Vektor $\mathbf{d}_2$:	tfidf-Vektor $\mathbf{d}_3$:
Geistiger	0	0	.48
Gesichtskreis	.48	0	0
Gesichtspunkt	0	0	.48
Haben	.48	0	0
Horizont	0	.18	.18
Kreis	0	.48	0
Manche	.48	0	0
Menschen	.18	.18	0
Nennen	.18	.18	0
Null	0	0	0
Radius	0	0	0
Standpunkt	.18	.18	0
vieler	0	.48	0

Tabelle 3: tfidf-Vektoren für die Beispieldokumente aus Tabelle 1.

Mit Kenntnis der tfidf-Vektoren ist es möglich, SLC anzuwenden. Dazu muss ein Ähnlichkeitsgraph aufgestellt werden. Die *sim*-Funktion des Graphen errechnet zu diesem Zweck die Kosinusähnlichkeit zwischen allen Dokumenten.

Abbildung 2 zeigt diesen Graphen, wobei jede Kante zusätzlich mit ihrem jeweiligen φ-Wert beschriftet ist. Es ist zu sehen, dass die Dokumente d_1, d_2 und d_3, untereinander einen je unterschiedlichen Grad inhaltlicher Übereinstimmung aufweisen.

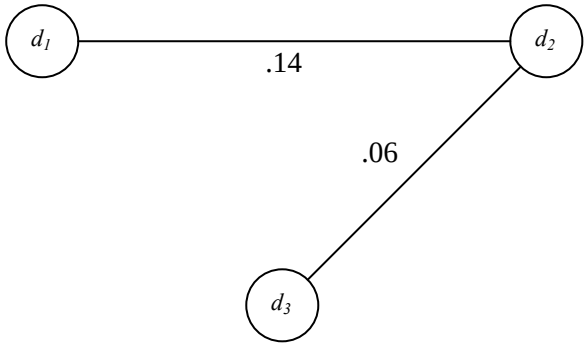

Abbildung 2: Ähnlichkeitsgraph für die Beispieldokumente aus Tabelle 1 mit $\alpha = 0$.

Bei einem Schwellwert von $\alpha = 0$ sind zwischen den Dokumenten d_1 und d_2 sowie d_2 und d_3 Kanten vorhanden, zwischen d_1 und d_3 hingegen nicht. Das ist darauf zurückzuführen, dass die Kosinusähnlichkeit zwischen Dokument 1 und 3 Null beträgt.

Dieser Schwellwert macht in Bezug auf das Ziel, die Ausgangsdokumente in Kategorien einzuteilen, keinen Sinn, da schon geringste Ähnlichkeiten zweier Dokumente ausreichen, so dass eine Kante zwischen ihnen existiert. Ein Wert von 0.5 wäre dafür beispielsweise eher geeignet. Dann besteht das Ergebnis aus drei Kategorien, eine für jedes Dokument, da kein Knoten im Ähnlichkeitsgraph eine Kante zu einem anderen aufweist.

Dennoch wirft dieses Ergebnis einige Fragen bezüglich des Verfahrens auf. Obwohl die Beispieldokumente offenkundig eine sehr starke inhaltliche Übereinstimmung aufweisen, sind die rechnerischen Ähnlichkeitswerte sehr gering. Die Kategorisierung in eine einzige Kategorie gelingt hier nur durch die Absenkung des Schwellwertes im Ähnlichkeitsgraphen. Es sei vorausgeschickt, dass dieses Ergebnis nicht der eigentlichen Praktikabilität Vektorraummodell-basierter Kategorisierung widerspricht, alles weitere wird dagegen im folgenden Kapitel diskutiert.

2.2 Suffix-Baum-Kategorisierung

Suffix-Baum-Kategorisierung (engl. Suffix-Tree-Clustering, kurz STC) ist ein, im Verhältnis zu TBC, relativ junger Ansatz für die Kategorisierung von Dokumenten. Erstmals wurde STC von O.E. Zamir und O. Etzioni vorgestellt [17,18]. Sie verfolgen den Ansatz, ein Dokument nicht als eine Menge von Wörtern zu betrachten, wie es im Vektorraummodell der Fall ist, sondern die Beziehungen der Wörter untereinander zu berücksichtigen. Die inhaltliche Übereinstimmung von Dokumenten wird daher anhand von Wortketten ermittelt werden, die die zu verarbeitenden Dokumente jeweils teilen.

Das von ihnen verwendete Dokumentmodell ist eine Datenstruktur, die es erlaubt, Dokumente unter diesen Voraussetzungen zu indizieren, der Suffix-Baum (engl. Suffix-Tree).

2.2.1 Suffix-Bäume als Dokumentmodell

Die Datenstruktur Suffix-Baum ist weithin bekannt und diente ursprünglich für effiziente Suchen in großen Datenmengen. P. Weiner stellte ihn im Zusammenhang seiner Lösung für lineare Suche in Texten erstmalig vor [5, 8, 14, 16]. Ziel der Datenstruktur ist es, einen Index zu erzeugen, der den effizienten, direkten Zugriff auf jede Teilzeichenkette des Eingabedokuments ermöglicht. Ein Suffix-Baum kann in linearer Zeit in der Länge der Eingabe erzeugt werden und verbraucht entsprechend linearen Platz. Die Frage, ob ein Dokument eine Zeichenkette enthält, kann mit einem Suffix-Baum, der es indiziert, linear in der Länge der Zeichenkette beantwortet werden.

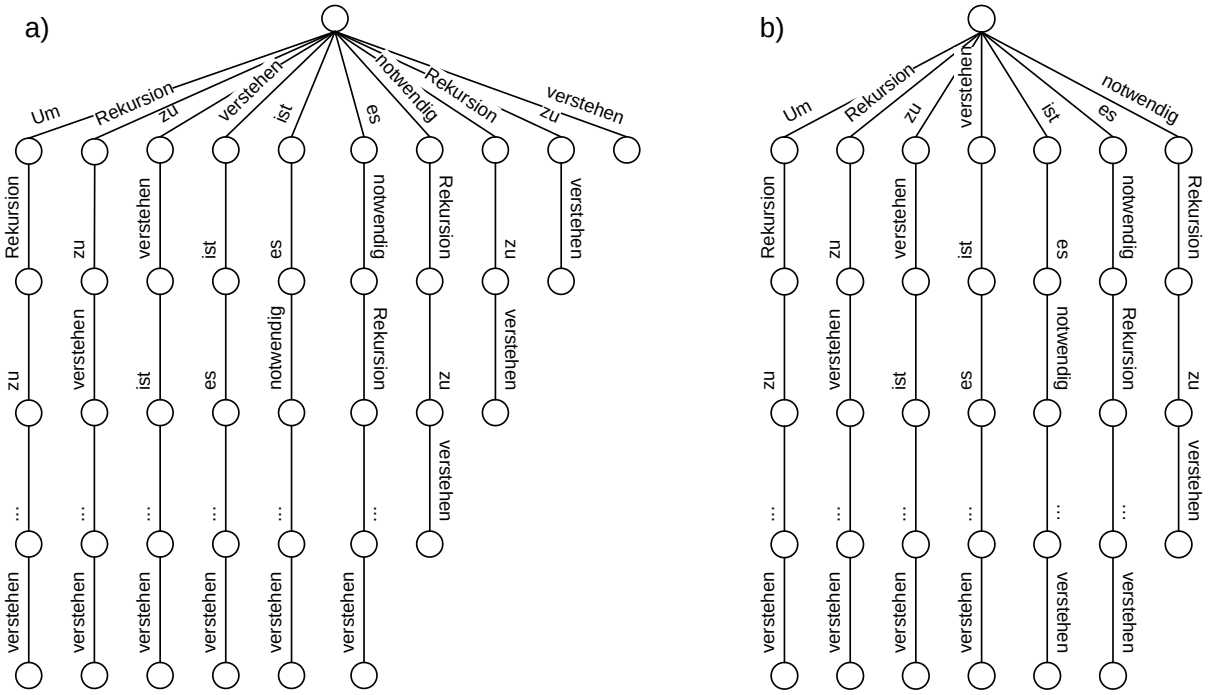

Abbildung 3: Darstellung zweier Vorstufen eines Suffix-Baums für den Text „Um Rekursion zu verstehen ist es notwendig, Rekursion zu verstehen“.

Die Indizierung eines Dokuments geschieht, wie der Name der Datenstruktur schon suggeriert, über seine Suffixe. Das *i*-te Suffix des Dokuments ist dabei die Wortkette, die beim Wort *i* beginnt und bis zu seinem Ende reicht. Wird die Menge aller Suffixe eines Dokuments betrachtet, so ist zu bemerken, dass jedes Wort im Dokument das erste Wort eines seiner Suffixe ist.

Jedes Suffix eines Dokuments sei nun durch einen von der Wurzel ausgehenden Pfad in einem Wurzelbaum repräsentiert, wobei jede Kante des Pfads mit je einem Wort des Suffixes Reihenfolge erhaltend beschriftet ist. Abbildung 3a zeigt einen solchen Baum für den Beispieltext „Um Rekursion zu verstehen ist es notwendig, Rekursion zu verstehen“. Diese Struktur indiziert den Text jedoch auf eine ineffiziente Art und Weise.

Erster Ansatzpunkt für eine Verbesserung dieser Struktur ist, zu erkennen, dass es mehrere Suffixe geben kann, die mit demselben Wort oder mit derselben Wortkette beginnen. Dieses längste gemeinsame Präfix einer Teilmenge der Suffixe tritt hier allerdings von der Wurzel ausgehend gleich mehrfach auf, nämlich für jeden Pfad, der eines der betreffenden Suffixe repräsentiert, einmal. Diese Redundanz kann vermieden werden, indem das längste gemeinsame Präfix aller Teilmengen der Suffixe eines Dokuments durch einen gemeinsamen Pfad repräsentiert wird, von dem aus die restlichen Teilsuffixe in eigene Teilbäume verzweigen. Bei jeder neuerlichen Verzweigung findet dieses Prinzip dann analog für alle verbliebenen Teilsuffixe Verwendung. In Abbildung 3b ist dies für den oben vorgestellten Text zu sehen. Das Suffix „Rekursion zu verstehen“ ist zum Beispiel vollständig in einem anderen Suffix enthalten.

Der zweite Ansatz zur Effizienzsteigerung ist, alle Knoten, die einen Ausgangsgrad von 1 haben, einzusparen. Diese Knoten verzweigen nicht in neue Teilbäume, sondern sind Teile eines Präfixes eines oder mehrerer Suffixe. Der Knoten an sich enthält daher keine zusätzliche Information, die für weitere Suchen notwendig wäre, da der traversierbare Pfad im Baum von ihm aus immer eindeutig ist. Die Einsparung des Knotens ist dadurch zu bewerkstelligen, dass die Kantenbeschriftungen der Eingangs- und Ausgangskante verknüpft eine gemeinsame Kante beschriften, die den Vater- und den

Sohn-Knoten verbindet. Auf diese Weise ist mit allen Knoten zu verfahren, deren Ausgangsgrad 1 ist.

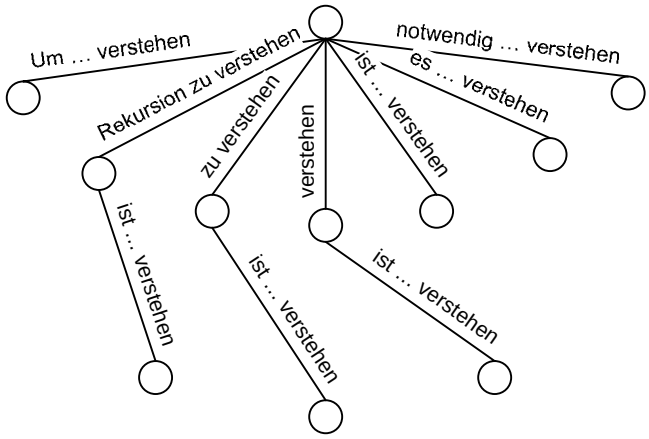

Abbildung 4: Suffix-Baum für den Text „Um Rekursion zu verstehen ist es notwendig, Rekursion zu verstehen".

Die Indizierung eines Dokuments, nach dem obigen Schema, stellt einen Suffix-Baum für das Dokument dar. Abbildung 4 zeigt einen Suffix-Baum für das obige Beispiel. Seine Definition lautet wie folgt:

Definition: Suffix-Baum
Sei d ein Dokument und S die Menge aller Suffixe von d. Dann heißt ein Wurzelbaum ST Suffix-Baum für d, genau dann wenn:

- *die Verknüpfung der Beschriftungen der Kanten auf einem beliebigen Pfad von der Wurzel bis zu einem Blattknoten ein Suffix aus S ist.*
- *für alle Suffixe aus S ein Pfad von der Wurzel bis zu einem beliebigen Knoten k existiert.*
- *die Beschriftungen aller ausgehenden Kanten eines Knotens paarweise verschieden sind.*
- *jeder Knoten, der nicht Wurzel oder Blattknoten ist, mindestens den Ausgangsgrad 2 hat.*

Auch die Indizierung einer Menge von Dokumenten ist mit einem Suffix-Baum möglich. Dazu ist es sinnvoll, aber nicht notwendig, jedem Dokument je ein eindeutiges Terminalwort anzuhängen, das ansonsten nicht in der Dokumentenmenge auftauchen darf. Die Menge aller Suffixe der Dokumentenmenge ist dann die Vereinigung der Mengen aller Suffixe jedes Dokuments. Der Suffix-Baum wird mit dieser Menge entsprechend der Definition aufgebaut.

Die Terminalwörter vermeiden, dass, falls mindestens zwei Dokumente ein Suffix teilen, dieses im resultierenden Suffix-Baum durch nur einen Pfad repräsentiert wird. Das geschieht dadurch, dass die eindeutigen Terminalwörter jedes Suffix mit seinem Herkunftsdokument kennzeichnen, und es damit in der Menge der Suffixe aller Dokumente unterscheidbar machen. Dokumente, die ohne Terminalwörter ein Suffix teilen, teilen mit Terminalwörtern ein längstes gemeinsames Präfix, welches sich als gemeinsamer Pfad im Suffix-Baum wieder findet. Folglich garantieren die Terminalwörter, dass jedes Suffix im Suffix-Baum durch einen Pfad von der Wurzel zu einem Blatt repräsentiert wird, und nicht durch einen Pfad von der Wurzel zu einem beliebigen Knoten. Diese Eigenschaft eines Suffix-Baums lässt sich wie folgt definieren:

Definition: blattabschließender Suffix-Baum
Sei d ein Dokument, S die Menge aller Suffixe aus d, ST ein Suffix-Baum basierend auf d und P die Menge aller Pfade von der Wurzel zu einem Blattknoten in ST. ST heißt blattabschließend, wenn eine bijektive Abbildung S → P existiert, so dass jedem Suffix genau ein Pfad in ST zugeordnet wird.

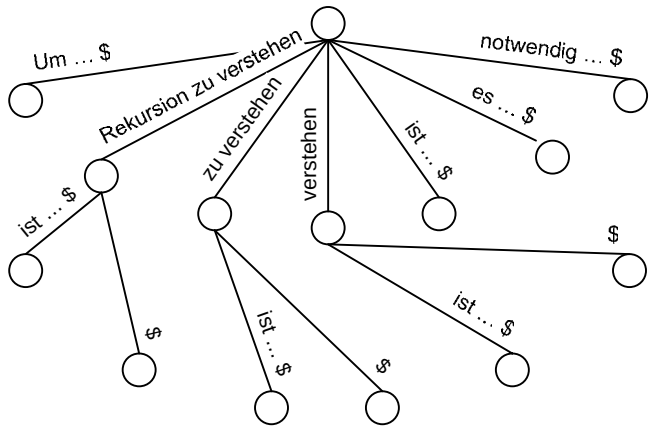

Abbildung 5: Blattabschließender Suffix-Baum für den Text „Um Rekursion zu verstehen ist es notwendig, Rekursion zu verstehen $".

Abbildung 5 zeigt einen blattabschließenden Suffix-Baum analog zu dem aus Abbildung 4. Es wurde das Wort „$" als Terminalwort eingefügt.

Wenn ein Suffix-Baum blattabschließend ist, dann existieren im Baum genau so viele Blätter, wie das Dokument oder die Menge der Dokumente, auf denen er beruht, Suffixe hat. Umgekehrt lassen sich bei jedem Suffix-Baum, über den bekannt ist, dass er blattabschließend ist, die Anzahl der Suffixe, aus denen das ursprüngliche Dokument besteht, ablesen, nämlich die Anzahl der Blätter im Baum. Der Knoten, der alle Blätter des Suffix-Baums vereint, ist die Wurzel des Baums selbst. Diese Beobachtung dem Wurzelknoten zuzuordnen ist also legitim. Es stellt sich dann jedoch die Frage, in welchem Zusammenhang die übrigen Knoten eines Suffix-Baums mit dieser Beobachtung stehen bzw. welche Aussage sich analog treffen lässt.

Satz:
Sei ST ein blattabschließender Suffix-Baum und k ein beliebiger Knoten aus ST, nicht aber seine Wurzel. Sei n_k ferner die Anzahl der Blattknoten im von k aufgespannten Teilbaum von ST und z die Zeichenkette, die sich aus der Verknüpfung aller Kantenbeschriftungen auf dem Pfad von der Wurzel bis nach k ergibt. Dann ist n_k gleich der Anzahl der Vorkommen von z im Ausgangsdokument, auf dem ST basiert.

Sei *d* ein Dokument und *z* eine Zeichenkette, die n_z mal in *d* enthalten ist. Daraus folgt, dass es genau n_z Suffixe gibt, die *z* als Präfix haben. Aus der Definition des Suffix-Baums folgt, dass in einem Suffix-Baum ST für *d* insbesondere für diese n_z Suffixe ein Pfad existieren muss. Ist *ST* blattabschließend, so schließen diese Pfade jeweils immer mit einem Blattknoten ab. Weiterhin gilt, da die Kantenbeschriftungen jeder Kante, die von einem bestimmten Knoten ausgehen paarweise verschieden beginnen müssen, dass alle Pfade der n_z Suffixe bis zu einem Knoten *k* einen Pfad in *ST* teilen. Die Zeichenkette *z* ist entweder gleich, oder ein Präfix der Verknüpfung aller Kantenbeschriftungen dieses Pfades bis *k*. Von *k* ausgehend verzweigt sich ein Teilbaum in *ST*, der für alle n_z Suffixe je einen Blattknoten enthält. Es können nicht weniger als n_z Blattknoten sein, da *z* genau n_z Mal im Ausgangstext vorkommt, aber auch nicht mehr als n_z, da dies voraussetzen würde, dass ein weiteres Suffix in *d* existiert, das *z* als Präfix hat.

Sei *ST* ein blattabschließender Suffix-Baum, *k* ein beliebiger Knoten aus *ST*, der nicht Wurzel ist, *z* die Verknüpfung aller Kantenbeschriftungen von der Wurzel bis *k* und n_k die Anzahl aller Blattknoten, die im durch *k* aufgespannten Teilbaum enthalten sind. Jeder Pfad von der Wurzel über *k* zu einem der n_k Blattknoten repräsentiert per Definition ein Suffix eines Dokuments *d*. Von jedem dieser Suffixe ist *z* jeweils Präfix. Da die Pfade der n_k Suffixe mit verschiedenen Blattknoten abschließen, sind die Suffixe selbst paarweise verschieden voneinander. Jedes der Suffixe ist ein Suffix des Dokuments *d*, auf dem *ST* beruht, was bedeutet, dass *z* genau n_k mal in *d* enthalten sein muss.

Die Quantisierung inhaltlicher Übereinstimmung zweier Dokumente wird im STC nicht benötigt. Daher ist kein Ähnlichkeitsmaß für dieses Dokumentmodell explizit definiert worden. Stattdessen werden während der Fusionierung die im Suffix-Baum kodierten Informationen über Wortketten ausgewertet, die mindestens zwei Dokumente teilen.

2.2.2 Fusionierung mittels Ähnlichkeitsgraphen

Die Kategorisierung einer Menge von Dokumenten *D* erfolgt im STC ebenfalls mit Hilfe eines Ähnlichkeitsgraphen und SLC. Die Knotenmenge des Ähnlichkeitsgraphen entspricht hier jedoch nicht der Dokumentenmenge, sondern wird aus den Knoten des Suffix-Baums errechnet. Es handelt sich dabei um Mengen von Dokumenten, die als Basis für die endgültigen Kategorien dienen. Sie werden daher auch Basiskategorien genannt. Es handelt sich dabei um die Menge aller inneren Knoten des Suffix-Baums, abgesehen von seiner Wurzel. Jeder dieser Knoten repräsentiert je eine Basiskategorie. Jede Basiskategorie enthält wiederum alle Dokumente, aus denen Suffixe in dem durch den entsprechenden Knoten aufgespannten Teilbaum enden. Daraus folgt, dass die Menge der Basiskategorien, die aus einem Suffix-Baum extrahiert werden, Teilmenge der Potenzmenge von *D* ist. Mit Hilfe der folgenden *sim*-Funktion wird für zwei Basiskategorien b_i und b_j bestimmt, ob eine Kante zwischen ihnen im Ähnlichkeitsgraphen existiert:

$$sim_{STC}(b_i, b_j, \alpha) = \begin{cases} 1 & \frac{|b_i \cap b_j|}{|b_i|} > \alpha \wedge \frac{|b_i \cap b_j|}{|b_j|} > \alpha \\ 0 & sonst \end{cases}$$

Die endgültigen Kategorien ergeben sich hier ebenfalls aus den Vereinigungsmengen der Basiskategorien der Zusammenhangskomponenten des Ähnlichkeitsgraphen.

Es werden in STC jedoch im Ähnlichkeitsgraphen nicht alle Kanten mittels sim_{STC} berechnet, sondern eine Auswahl der vielversprechendsten Basiskategorien getroffen und von jedem seine Kanten zu allen übrigen berechnet. Auf diese Weise wird eine in der Anzahl der Dokumente quadratische Laufzeit reduziert auf eine quadratische Laufzeit in der Anzahl der ausgewählten Basiskategorien.

Zu diesem Zweck wird eine Bewertungsfunktion eingesetzt, die jeder Basiskategorie auf Grundlage der Anzahl enthaltener Dokumente und der Länge der durch seine Dokumente geteilten Wortkette einen spezifischen Wert zuweist. Die Wortkette ergibt sich dabei aus der Verknüpfung aller Kantenbeschriftungen von der Wurzel des Suffix-Baums zum Knotenrepräsentanten der betrachteten Basiskategorie. Die Bewertung ergibt sich aus der Multiplikation der Satzlänge mit der Anzahl der in der Basiskategorie enthaltenen Dokumente. Die vielversprechendsten Basiskategorien sind diejenigen mit der höchsten Bewertung.

2.2.3 Generierung von Kategoriebzeichnern

Die Generierung der Bezeichner für die endgültigen Kategorien ist in STC kein eigenständiger Algorithmus, sondern sowohl mit dem Dokumentmodell als auch mit dem Fusionierungs-Algorithmus verwoben.

Während der Fusionierung erhalten alle aus dem Suffix-Baum extrahierten Basiskategorien bereits eine eigene Bezeichnung. Es handelt sich dabei um die Wortkette, die sich aus der Verknüpfung der Kantenbeschriftung von der Wurzel des Suffix-Baums zum Knotenrepräsentanten einer Basiskategorie ergibt. Im Ähnlichkeitsgraphen sind die Basiskategorien während der Fusionierung Teil einer Zusammenhangskomponente. Die endgültige Kategorie, die sich aus einem solchen Teilgraph ergibt, erhält daraufhin die Vereinigung der Bezeichner seiner Basiskategorien als Bezeichnung. Dabei werden längere Wortketten bevorzugt, da sie aussagekräftiger sind.

Wichtigstes Merkmal dieser Form der Kategoriebezichner ist, dass hier nicht eine Sammlung von Einzelwörtern, sondern eine oder mehrere Wortketten den Inhalt einer Kategorie charakterisieren. Da diese Wortketten außerdem von mindestens zwei der in der Kategorie enthaltenen Dokumente geteilt werden, erscheint diese Form der Benennung plausibel.

2.2.4 Laufzeit

Die Laufzeit von STC setzt sich Zusammen aus den Laufzeiten der Vorverarbeitung, der Initialisierung des Dokumentmodells und der Fusionierung. Da die Generierung der Kategoriebezeichner mit den letztgenannten Schritten verwoben ist, entfällt auf diesen Schritt keine Laufzeit. Der Schritt der Vorverarbeitung der Dokumente benötigt, analog zu TBC, $O(n)$, wobei n die Anzahl der zu kategorisierenden Dokumente ist.

Die Initialisierung eines Suffix-Baums für ein Dokument geschieht mit Ukkonens Algorithmus linear in der Länge des Dokuments in Worten [14]. Unter der Voraussetzung von konstant langen Dokumenten folgt daraus eine konstante Initialisierungszeit. Damit beträgt die Laufzeit, einen Suffix-Baum für n Dokumente zu initialisieren, $O(n)$.

Die Auswertung des Suffix-Baums nach Basiskategorien und deren Bezeichnern geschieht mittels eines einmaligen Durchlaufs durch den Baum. Da die Anzahl der Knoten und Kanten im Suffix-Baum ebenfalls linear in der Anzahl der Dokumente ist, gilt dasselbe für diesen Schritt. Die Bewertung und Auswahl der vielversprechendsten Basiskategorien geschieht in $O(n)$. Da ab hier nur noch m Basiskategorien weiterverarbeitet werden, geschehen die Aufstellung des Ähnlichkeitsgraphen und seine Auswertung in $O(m^2)$.

2.2.5 Beispiel für Suffix-Baum-Kategorisierung

Um STC zu veranschaulichen, werden auch hier die drei Dokumente aus Tabelle 1 kategorisiert. Dabei müssen die Dokumente ebenfalls einer Vorverarbeitung unterzogen werden. Daher kann hier direkt mit den Dokumenten, wie sie in Tabelle 2 aufgeführt sind, fortgefahren werden.

Der nächste Schritt ist die Aufstellung eines Suffix-Baums für die drei Dokumente. Da hier eine Menge von Dokumenten durch einen blattabschließenden Suffix-Baum indiziert werden soll, ist es sinnvoll, dass jedem Dokument ein einzigartiges Terminalwort hinzugefügt wird, das sonst nicht in einem der Dokumente enthalten ist. Hier dient „$$_i$“ als Terminalwort, wobei $i \in \{1,2,3\}$ der Index desjenigen Dokuments ist, in dem das Terminalwort eingefügt wurde. Tabelle 4 zeigt die durch Terminalwörter ergänzten Dokumente.

Dokument d_1:	Dokument d_2:	Dokument d_3:
Manche Menschen haben Gesichtskreis Radius Null nennen Standpunkt $\$_1$	Horizont vieler Menschen Kreis Radius Null nennen Standpunkt $\$_2$	Gesichtspunkt geistiger Horizont Radius Null $\$_3$

Tabelle 4: Die Beispieldokumente aus Tabelle 2. Jedem Dokument wurde ein einzigartiges Terminalwort hinzugefügt.

Abbildung 6 zeigt den Suffix-Baum für die Dokumente aus Tabelle 4. Ein Suffix setzt sich aus der Verknüpfung der Kantenbeschriftungen auf einem beliebigen Pfad von der Wurzel zu einem Blattknoten zusammen. Das Terminalwort lässt erkennen, aus welchem Dokument das jeweilige Suffix stammt. Falls längere Wortketten als Kantenbeschriftung dienen, so sind diese durch das erste Wort, gefolgt von drei Punkten und dem letzten Wort der Kette, angedeutet.

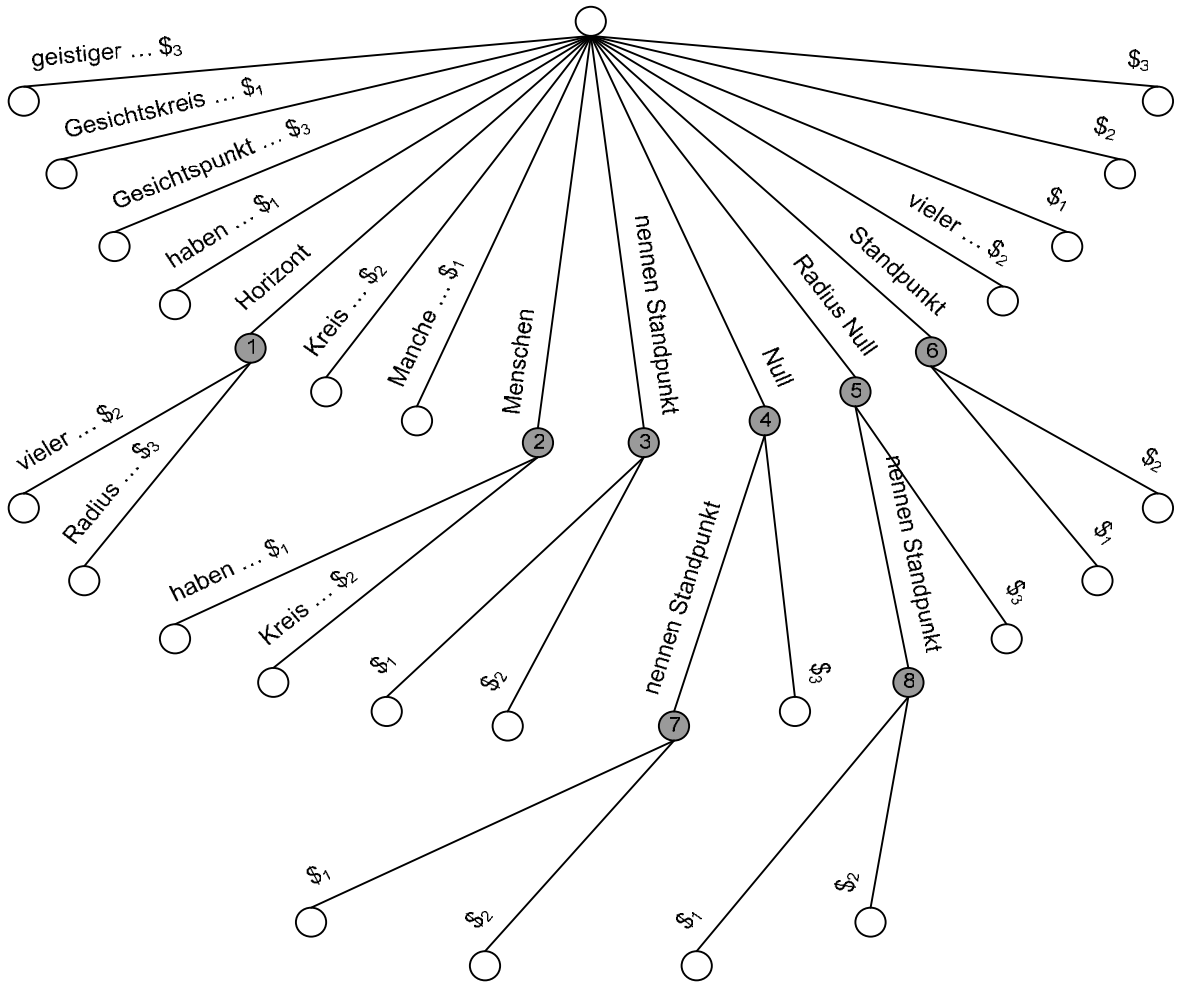

Abbildung 6: Suffix-Baum für die Beispieldokumente aus Tabelle 4

Aus diesem Suffix-Baum werden nun die Basiskategorien extrahiert. Diese werden in Abbildung 6 durch die grau markierten und nummerierten Knoten repräsentiert. Jeder der acht Basiskategorien steht für eine Menge von Dokumenten. Beispielsweise repräsentiert der mit 1 beschriftete Knoten eine Basiskategorie b_1, die alle Dokumente enthält, welche Suffixe enthalten, deren längstes gemeinsames Präfix das Wort „Horizont" ist, nämlich $b_1 = \{d_2,d_3\}$. Aus den so gewonnenen Basiskategorien lässt sich mittels der sim_{STC}-Funktion ein Ähnlichkeitsgraph aufstellen, der in Abbildung 7 gezeigt wird.

$b_1 = \{d_2,d_3\}$
$b_2 = \{d_1,d_2\}$
$b_3 = \{d_1,d_2\}$
$b_4 = \{d_1,d_2,d_3\}$
$b_5 = \{d_1,d_2,d_3\}$
$b_6 = \{d_1,d_2\}$
$b_7 = \{d_1,d_2\}$
$b_8 = \{d_1,d_2\}$

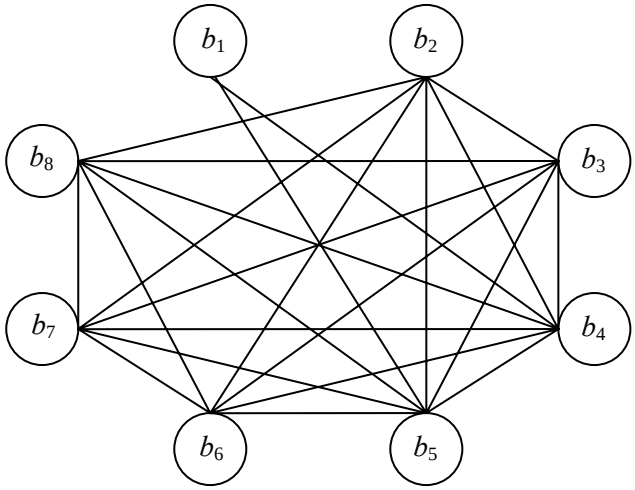

Abbildung 7: Basecluster-Graph für die aus Abbildung 6 extrahierten Basiskategorien, die im Suffix-Baum-Kategorisierung verwendete *sim*-Funktion und $\alpha = 0.5$.

Es ist leicht zu sehen, dass der Ähnlichkeitsgraph sehr dicht ist und es nur eine einzige Zusammenhangskomponente gibt. Daraus folgt, dass auch nur eine einzige Kategorie erzeugt wird, in der alle Beispieldokumente enthalten sind

Auf die Bewertung der Basiskategorien kann, aufgrund der Überschaubarkeit der in diesem Beispiel zu kategorisierenden Dokumente, verzichtet werden. Die Bezeichnung der endgültigen Kategorie ist die Vereinigung aller von den Dokumenten paarweise geteilten Wortketten.

3 Theoretischer Vergleich

TBC und STC weisen an verschiedenen Stellen Unterschiede auf, die Anlass zu einer genaueren Betrachtung geben. Dieses Kapitel betrifft die konträren Ergebnisse der gezeigten Beispiele des vorigen Kapitels. Im Anschluss daran werden einige besondere Eigenschaften, die STC gegenüber TBC auszeichnen, vorgestellt. Die Nachfolgenden Abschnitte stellen einige Nachteile von STC heraus, die die Vollständigkeit der Kategorisierung durch STC, die Gründe für die Auswahl vielversprechender Basiskategorien und die Berücksichtigung von transitiven Ähnlichkeiten betreffen.

3.1 Interpretation der Anwendungsergebnisse

Die Ergebnisse aus dem im vorigen Kapitel gezeigten Beispiel erscheinen nicht sehr plausibel. Ziel beider Kategorisierungsverfahren ist, inhaltlich übereinstimmende Dokumente einander zuzuordnen. Während STC das erwartete Ergebnis erzeugt, liefert TBC hier jedoch im Wesentlichen das Gegenteil.

Dieses zunächst gegenintuitive Verhalten basiert auf den im Dokumentmodell von TBC getroffenen Annahmen. Dort wird für jedes Wort eines Dokuments der tfidf-Wert ermittelt. Dieser Wert basiert jedoch nicht allein auf Informationen, die aus dem Dokument selbst gewonnen wurden, sondern auch auf solchen, die aus der Menge aller Dokumente, die es zu kategorisieren gilt, stammen, nämlich der inversen Dokumenthäufigkeit. Die *idf*-Werte, die mit den *tf*-Werten eines Wortes multipliziert werden, sorgen dafür, dass die Relevanz des Wortes in Bezug auf den Inhalt eines Dokuments mit seinem Vermögen, dieses Dokument von anderen abzugrenzen, gewichtet wird. Wörter, die häufig in der Dokumentenmenge enthalten sind, erhalten daher niedrigere tfidf-Werte.

Im Beispiel sind in der Ausgangsmenge drei Dokumente enthalten, die inhaltlich gleich sind. Der Argumentation von Luhn folgend, wurden also in allen Dokumenten bestimmte Wörter verwendet, die benötigt werden, das Gesagte auszudrücken. Hier stechen zum Beispiel die Wörter „Radius“ und „Null“ hervor. Sie tauchen in allen Dokumenten auf. Das führt dazu, dass die Dokumenthäufigkeit dieser Wörter gleich der Anzahl der Dokumente ist und sie somit nicht geeignet erscheinen, die Dokumente voneinander abzugrenzen. Im Gegenteil, sie sind an sich die stärksten Indikatoren dafür, dass die Dokumente inhaltlich gleich sind. Im Gegenzug werden Wörter, die nur in einem der Dokumente enthalten sind, als besonders geeignet bewertet, den Inhalt des Dokuments selbst zu repräsentieren und es von den übrigen abzugrenzen.

Insofern ist das Ergebnis im Sinne der Annahmen richtig. Dennoch stellt sich die Frage, welche Voraussetzungen erfüllt sein müssen, damit die Dokumente in derselben Kategorie einsortiert werden. Dazu ist eine größere Ausgangsmenge erforderlich, bei der die Worthäufigkeiten möglichst repräsentativ verteilt sind. In einer solchen Menge würde voraussichtlich das Wort „Radius“ einen größeren tfidf-Wert erlangen, als zum Beispiel das Wort „Manche“, das an sich in kaum einem Sinnzusammenhang mit dem Inhalt der Dokumente steht.

Bei Betrachtung des Ergebnisses von STC fällt, im Gegensatz zu dem von TBC, die große Dichte des Ähnlichkeitsgraphen auf. STC hat keine Probleme, auch bei kleinen Ausgangsmengen sich inhaltlich stark überschneidende Dokumente miteinander zu verknüpfen. Dies ist auch hier auf das verwendete Dokumentmodell zurückzuführen. Eine seiner besonderen Eigenschaften ist, dass für zwei gleiche Suffixe niemals mehr als ein Pfad von der Wurzel ausgeht. Gleiche Suffixe teilen sich also einen Pfad im Baum. Stammen nun zwei gleiche Suffixe aus verschiedenen Dokumenten, so sind alle Kanten auf ihrem gemeinsamen Pfad ein Indikator dafür, dass die beiden Ursprungsdokumente

genau dieses Suffix teilen. Das wiederum wird als Indiz dafür betrachtet, dass beide Dokumente inhaltlich übereinstimmen. Auf dieser Grundlage können mit Hilfe eines Suffix-Baums, durch Auswertung seiner inneren Knoten, die paarweisen Gemeinsamkeiten zwischen allen Dokumenten einer Menge gefunden werden. Da die Beispieldokumente, wie im entsprechenden Suffix-Baum zu sehen ist, eine Reihe von Gemeinsamkeiten aufweisen, erkennt STC sie als inhaltlich gleich.

Es spielt dabei keine Rolle, wie viele Dokumente in der Ausgangsmenge sind, da aus dem Suffix-Baum nur die Informationen über die paarweisen Gemeinsamkeiten aller Dokumente entnommen werden. STC benutzt darüber hinaus keine Informationen über die gesamte Dokumentenmenge.

Diese Eigenschaft von STC lässt sich mit dem Begriff „lokal“ fassen. Ausgehend von einer Menge zu kategorisierender Dokumente, arbeitet STC während der Indizierung immer lokal auf einem bestimmten Dokument der Menge. TBC hingegen arbeitet zu jedem Zeitpunkt global auf der gesamten Menge, da für die Indizierung eines Dokuments immer Informationen über alle Dokumente benötigt werden.

3.2 Eigenschaften von STC

STC zeichnet sich gegenüber TBC durch zwei Eigenschaften aus, die Auswirkungen auf die Effizienz der Anwendung bzw. die Qualität und Form des Ergebnisses haben. Sie gehen nicht zuletzt auf das Dokumentmodell Suffix-Baum zurück. STC besitzt diese Eigenschaften aber nicht exklusiv. Auch TBC und andere Kategorisierungsverfahren lassen sich dahingehend anpassen. Es handelt sich um Inkrementalität des Algorithmus und Überlappung der endgültigen Kategorien.

Inkrementalität bedeutet sinngemäß schrittweises Vorgehen. Algorithmen, die diese Eigenschaft aufweisen, zeichnen sich dadurch aus, dass sie, mit einer gewissen Granularität, zur Laufzeit wiederkehrende Arbeitsschritte ausführen und so das Ergebnis Stück für Stück ausbauen, also inkrementieren. Suffix-Bäume sind inkrementell initialisierbar [14].

Die Arbeitsschritte bei der Initialisierung von Suffix-Bäumen sind hier das Aufnehmen neuer Dokumente in die Datenstruktur. Nach jedem Arbeitsschritt ist ein der Definition entsprechender Suffix-Baum als Zwischenergebnis verfügbar. Daher kann zum Beispiel auch ohne vollständige Kenntnis einer zu kategorisierenden Dokumentenmenge bereits mit dem Aufbau des Dokumentmodells begonnen werden. Da nach jedem Arbeitsschritt während der Initialisierung bereits ein Suffix-Baum zur Verfügung steht, ist es außerdem möglich, das Fusionierungsverfahren hierfür in Gang zu setzen bzw. dessen Ergebnisse ständig zu aktualisieren. Somit erbt STC diese Eigenschaft von seinem Dokumentmodell.

TBC hingegen ist dazu, aufgrund seines Dokumentmodells, nicht in der Lage. Hier sorgt die globale Arbeitsweise während der Indizierung von Dokumenten dafür, dass vor der Berechnung eines tfidf-Vektors für ein Dokument alle übrigen bekannt sein müssen. Ist das nicht der Fall, so kann es im Extremfall dazu kommen, dass die *idf*-Werte sich mit der Kenntnis jedes neuen Dokuments verändern, was die Anpassung aller bisher erstellten Vektoren nötig machen würde. Falls hingegen die *idf*-Werte einer Datenbasis entnommen würden, anstatt die tatsächlichen zu verwenden, so wäre auch TBC inkrementell.

Die Ergebnisse, die von STC generiert werden, weisen die Eigenschaft auf, überlappend zu sein. Das bedeutet, dass Dokumente nicht ausschließlich einer einzigen Kategorie zugeordnet werden, sondern möglicherweise mehreren. Die Grundannahme dahinter ist, dass ein Dokument mehrere Sachverhalte beinhalten kann, so dass eine ausreichende Ähnlichkeit zu mehr als einer Kategorie gegeben ist. Es ist sinnvoll dieses Dokument

dann in beiden Kategorien abzulegen, da das Verfahren nicht wissen kann, welche der Anwender letztendlich auswählen wird, ihm aber keine möglicherweise wichtigen Informationen vorenthalten werden sollen.

Ursächlich geht diese Eigenschaft darauf zurück, dass die für die Fusionierung durch SLC gewählten Basiskategorien bereits überlappend gewählt werden. Alle Basiskategorien werden im Suffix-Baum durch einen Knoten repräsentiert. Die Knoten, die als Grundlage für Basiskategorien ausgewählt werden, sind alle inneren Knoten. Für diese Knoten gilt, dass sie auf einem Pfad liegen, der mindestens zwei Suffixe repräsentiert. Mit steigender Anzahl der Dokumente steigt daher auch die Wahrscheinlichkeit, dass nicht beide Suffixe aus ein- und demselben Dokument stammen, was dazu führt, dass die daraus gewonnene Basiskategorie aus mehr als einem Dokument besteht. Eine zu starke Überlappung zweier Basiskategorien führt jedoch dazu, dass beide im Ähnlichkeitsgraphen in derselben Zusammenhangskomponente liegen und damit auch zu einer Kategorie vereinigt werden.

TBC ermöglicht überlappende Kategorien nicht, ohne dass ein anderer Fusionierungsalgorithmus angewendet wird. Hierzu sei außerdem auf das folgenden Kapitel verwiesen, in dem diskutiert wird, ob auf Grundlage des tfidf-Vektorraummodells die Kategorisierungsergebnisse von STC approximiert werden können.

3.3 Unvollständigkeit von STC

Es ist möglich, dass nicht alle Dokumente von STC kategorisiert werden, so dass die Vereinigung aller von STC erzeugten Kategorien nicht der Ausgangsmenge von Dokumenten entspricht. Das heißt, dass nach der Ausführung von STC nicht zwangsläufig alle eingegebenen Dokumente auch in den erzeugten Kategorien enthalten sind. Dies verletzt die Eigenschaft von Kategorisierungen, dass die Vereinigungsmenge aller Kategorien der Ausgangsmenge entspricht. Es gibt zwei Ursachen dafür, die beide im zweiten Schritt von STC begründet sind.

Das erste Problem entsteht während der Auswertung des Dokumentmodells, also der Extraktion der Basiskategorien aus dem Suffix-Baum. Dabei kann es vorkommen, dass ein Dokument in keiner Basiskategorie enthalten ist und folglich auch in keiner der endgültigen Kategorien enthalten sein kann. Dies lässt sich an einem einfachen Beispiel demonstrieren, in dem das Zitat aus Tabelle 5 Verwendung findet.

Dokument *d*:	Nach der Vorverarbeitung:
„Es irrt der Mensch, solang' er strebt." Johann Wolfgang von Goethe (1749 – 1832)	irren Mensch solange streben

Tabelle 5: Beispieldokument zur Demonstration der Unvollständigkeit von STC.

Dieses Dokument soll nun als einziges mit STC kategorisiert werden. Um das erwartete Ergebnis eines solchen Unterfangens, also der Kategorisierung eines einzigen Dokuments, vorweg zu nehmen; Das Endergebnis sollte aus einer Kategorie bestehen, die ebendieses Dokument enthält. Nach der Vorverarbeitung des Dokuments wird ein Suffix-Baum dafür erstellt.

An dieser Stelle der Fusionierung werden alle inneren Knoten des Suffix-Baums, die weder Wurzel noch Blattknoten sind, für die Erstellung der Basiskategorien ausgewertet. Allerdings gibt es im Suffix-Baum für das Dokument, der in Abbildung 8 zu sehen ist, keine solchen Knoten. Aus diesem Grund können auch keinerlei Basiskategorien aufgestellt werden, was dazu führt, dass das Endergebnis von STC leer ist.

Verallgemeinert lässt sich die Behauptung aufstellen, dass Dokumente, die nicht ein einziges Wort mit einem der anderen Dokumente aus einer zu kategorisierenden Menge teilen, nicht im Endergebnis von STC enthalten sind. Es handelt sich hierbei um einen Grenzfall, da mit steigender Anzahl oder Länge der Dokumente auch die Wahrscheinlichkeit steigt, dass jedes Dokument zumindest ein Wort mit einem anderen teilt. Das führt wiederum dazu, dass alle Dokumente jeweils wenigstens einer Basiskategorie zugeordnet sind und somit auch Teil des Endergebnisses sein können.

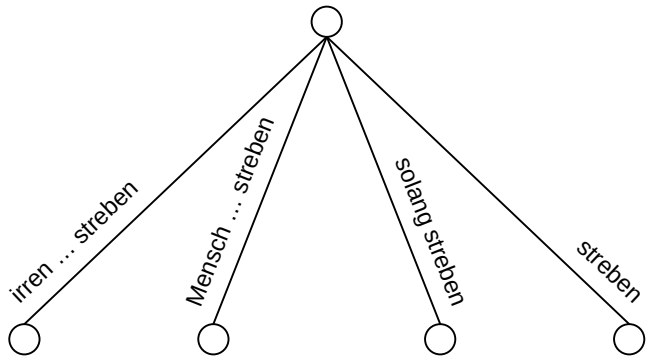

Abbildung 8: Suffix-Baum für das Dokument aus Tabelle 5.

An dieser Stelle kommt allerdings die zweite Ursache für die Unvollständigkeit zum tragen, die es auch in größeren Dokumentenmengen möglich macht, dass nicht alle Dokumente kategorisiert werden.

Es handelt sich um die nach der Basiskategorie-Extraktion getroffene Auswahl der vielversprechendsten Basiskategorien. Von allen aus einem Suffix-Baum gewonnenen Basiskategorien wird eine bestimmte Anzahl der durch eine Bewertungsfunktion am höchsten bewerteten Basiskategorien ausgewählt. Aus dieser Auswahl werden dann die endgültigen Kategorien gewonnen.

Dieses Vorgehen hat zur Folge, dass Dokumente, die nur geringe Ähnlichkeiten zu allen übrigen aufweisen, unter Umständen nicht kategorisiert werden. Das ist dann der Fall, wenn sie in nur wenigen oder ausschließlich niedrig bewerteten Basiskategorien vertreten sind. Die Auswahl der vielversprechendsten Basiskategorien polarisiert sozusagen das Ergebnis, da hauptsächlich diejenigen Dokumente berücksichtigt werden, die große Ähnlichkeiten zu mindestens einem anderen aufweisen. Einzigartige Dokumente werden in STC hingegen vernachlässigt.

Es ist nicht klar, inwieweit dieser Effekt in der Praxis zum Tragen kommt, da hier insbesondere die Zusammensetzung der zu kategorisierenden Dokumentenmenge einen großen Einfluss hat. Nichtsdestoweniger sind dies starke Indizien, die darauf hindeuten, dass STC unvollständig kategorisiert.

3.4 Gründe für die Auswahl vielversprechender Basiskategorien

Unter dem Gesichtspunkt der Unvollständigkeit stellt sich die Frage, wie sich die Arbeitsweise von STC verändert, wenn auf die Auswahl der vielversprechendsten Basiskategorien verzichtet wird. Dazu ist es zunächst notwendig zu klären, wie die Basiskategorien, die aus einem Suffix-Baum gewonnen werden, beschaffen sind.

Die Basiskategorien, die aus Knoten der ersten Ebene gewonnen werden, enthalten in der Praxis sehr viele Dokumente. Das erschließt sich daraus, dass bei einer hinreichend großen Anzahl von Dokumenten, jedes Dokument eine größere Anzahl von Einzelwörtern mit anderen teilt, da statistisch gesehen nur ein begrenzt großer Wortschatz verwendet wird. Daraus folgt, dass die Dokumente, die in Basiskategorien

zusammengefasst sind, die sich aus Knoten auf der ersten Ebene des Suffix-Baums ergeben, nicht mehr als ein einzelnes Wort teilen. Es ist sehr unwahrscheinlich, dass mehrere Dokumente eine längere Wortkette teilen, ohne dass andere Dokumente einzelne Wörter aus dieser Kette beinhalten.

Die Basiskategorien, die sich aus Knoten der zweiten Ebene ergeben und deren Dokumente daher mindestens zwei aufeinander folgende Wörter teilen, sind sehr viel kleiner, da die Wahrscheinlichkeit, dass eine große Anzahl von Dokumenten zwei aufeinander folgende Wörter teilen wesentlich geringer ist. Dasselbe gilt für Basiskategorien aus tieferen Ebenen, die nochmals kleiner sind als die der zweiten.

Die Menge aller Basiskategorien unterteilt sich also in diejenigen, die aus der ersten Ebene des Baums stammen und sehr viele Dokumente umfassen, und diejenigen, die nur wenige Dokumente enthalten und aus tieferen Ebenen gewonnen werden. In einem Ähnlichkeitsgraph wären diese beiden Sorten nicht miteinander verbunden. Das folgt aus der sim_{STC}-Funktion, die eine Kante zwischen zwei Basiskategorien genau dann ermöglicht, wenn sie sich wenigstens zu einem durch α bestimmten Prozentsatz überlappen. Zamir und Etzioni geben in ihrem Papier hierfür einen Wert von 0.5 vor [17]. Es ist anzunehmen, dass die Basiskategorien tieferer Ebenen sich zu weniger als 50 Prozent mit denen der ersten Ebene überlappen.

Wenn von dieser Beschaffenheit der Basiskategorien ausgegangen wird, unterteilt sich der Ähnlichkeitsgraph in Zusammenhangskomponenten, die aus großen, sich stark überlappenden, Basiskategorien bestehen und solche, die aus kleinen Basiskategorien bestehen. Es würden sich also wenige große Kategorien ergeben, die sehr viele Dokumente umfassen und viele kleine.

Die großen Kategorien sind für einen Anwender wenig hilfreich, da sie zu unscharf sind, und nur auf der Basis von durch die Dokumente geteilten Einzelwörtern erstellt wurden. Die kleineren hingegen wurden auf Basis von geteilten Wortketten ermittelt und sind damit von größerem Wert. Durch die Auswahl der vielversprechendsten Basiskategorien, bei der die Basiskategorien der ersten Ebene herausfallen, geschieht in STC daher eine zusätzliche Stopp-Wort-Elimination, bei der Einzelwörter unberücksichtigt bleiben, die in zu vielen Dokumenten enthalten sind. Das entspricht den Grundannahmen, auf denen auch die inverse Dokumenthäufigkeit beruht.

Bei diesen Erwägungen wird davon ausgegangen, dass mit STC vollständige Dokumente kategorisiert werden, also zum Beispiel wissenschaftliche Abhandlungen oder Webseiten. Bei der Kategorisierung von letzteren jedoch, die zum Beispiel im Rahmen von Websuchen geschieht, erhalten Kategorisierungsverfahren in der Regel nur Schnipsel der Webseiten, die kleine Ausschnitte des Textes um die gefundenen Suchbegriffe sind. Hier verhält sich STC nicht wie oben beschrieben.

Die Basiskategorien auf der ersten Ebene eines Suffix-Baums sind bei der Kategorisierung von Schnipseln kleiner, als wenn vollständige Dokumente kategorisiert werden. Das folgt aus der Tatsache, dass Schnipsel einen sehr kleinen Ausschnitt eines Dokuments um einen gefundenen Suchbegriff umfassen. Die Wahrscheinlichkeit, dass sehr viele davon in nächster Umgebung des Suchbegriffs andere Wörter teilen ist hier wesentlich geringer. Zudem sinkt bei der Verwendung von Schnipseln die Wahrscheinlichkeit, dass längere Wortketten geteilt werden.

Diese Überlegungen bedeuten daher eine Aufwertung der Basiskategorien, die aus der ersten Ebene eines Suffix-Baums gewonnen werden. Sie sind bei der Kategorisierung von Schnipseln von größerer Relevanz als in allen übrigen Fällen. Deshalb erscheint hier der Verzicht auf die Auswahl der vielversprechendsten Basiskategorien sinnvoll. Damit bleibt ferner die Vollständigkeit der Kategorisierung gewahrt.

3.5 Einbeziehung transitiver Ähnlichkeit von Dokumenten in STC

Es ist für STC nicht klar, unter welchen Bedingungen zwei Dokumente d_i und d_j in ein- und derselben Kategorie einsortiert werden. Bezogen auf SLC lautet die Fragestellung präziser, welche Bedingungen vorliegen müssen, damit zwei Dokumente im Ähnlichkeitsgraph in derselben Zusammenhangskomponente liegen.

Für TBC genügt dafür eine hinreichend große Parallelität der tfidf-Vektoren $\mathbf{d}_i$ und $\mathbf{d}_j$, die durch den Kosinus des Winkels zwischen ihnen quantifiziert wird. Die Parallelität steigt mit der Anzahl der Wörter, die von beiden Dokumenten geteilt werden, und deren Relevanz in Bezug auf den Inhalt der Dokumente. Ersteres bedingt die Anzahl der Einträge, die in beiden Vektoren einen von 0 verschiedenen Wert aufweisen, letzteres bedingt die Höhe des tfidf-Werts für die jeweiligen Wörter. Determiniert wird dies ausschließlich über die sim_{TBC}-Funktion.

In STC hingegen gibt es zwei Fälle, die dazu führen können, dass d_i und d_j in derselben Kategorie einsortiert werden. Der erste Fall ist, dass beide Dokumente ein oder mehrere Wörter oder Wortketten teilen. Das hat zur Folge, dass beide Dokumente zusammen in mindestens einer Basiskategorie vertreten sind und damit auch mindestens einmal in derselben Zusammenhangskomponente. Dieser Fall wird bereits mit Aufstellung des Suffix-Baums für die Dokumentenmenge determiniert.

Der zweite Fall tritt ein, wenn d_i und d_j in verschiedenen Basiskategorien enthalten sind, die sich stark genug überlappen, so dass, respektive der sim_{STC}-Funktion, eine Kante zwischen ihnen existiert. Die Dokumente dürfen also einerseits keinerlei Wörter order Wortketten teilen. Andererseits ist es notwendig, dass beide zu mindestens einem dritten Dokument d_k Gemeinsamkeiten aufweisen, da ansonsten die Basiskategorien der beiden Dokumente sich nicht überlappen würden. STC berücksichtigt hier also transitive Ähnlichkeiten zwischen d_i und d_j.

Dieser Fall tritt beispielsweise ein, wenn das Dokument d_i das Wort „Atom“ enthält, d_j das Wort „Gott“ und ein drittes Dokument d_k existiert, das beide Wörter enthält. Aus einem Suffix-Baum für diese Dokumente werden die Basiskategorien $\{d_i,d_k\}$ und $\{d_j,d_k\}$ ermittelt. Zwischen ihnen existiert respektive sim_{STC} für alle Werte von $\alpha < 0.5$ eine Kante. d_i und d_j liegen dann in derselben Zusammenhangskomponente.

Die Berücksichtigung transitiver Ähnlichkeiten macht in dieser Form jedoch keinen Sinn. Es werden Dokumente in derselben Kategorie vereint, die keinerlei Gemeinsamkeiten aufweisen. Dies macht die Kategorisierung unschärfer, da eine Kategorie auch Dokumente zu verschiedenen Themen umfassen kann. Der eigentliche Vorteil, der durch die Verwendung sich überlappender Kategorien entsteht, nämlich dass berücksichtigt wird, dass ein Dokument auch unterschiedliche Themen beinhalten kann, verkehrt sich ins Gegenteil.

Je höher der Schwellwert α gewählt wird, desto weniger kommt dieser Effekt zum Tragen, da nur noch Basiskategorien verbunden werden, die sich sehr stark überlappen. Eine solch starke Überlappung tritt am häufigsten bei Basiskategorien auf, die aus der ersten Ebene des Suffix-Baums gewonnen wurden. Die vorherige Auswahl vielversprechender Basiskategorien sorgt jedoch dafür, dass diese nicht als Knoten im Ähnlichkeitsgraph berücksichtigt werden.

Es ist also fraglich, inwieweit die Anwendung von SLC auf die aus dem Suffix-Baum gewonnenen Basiskategorien Sinn macht, beziehungsweise, ob sich die endgültige Kategorisierung sehr stark von den Basiskategorien unterscheidet. Wenn das nicht der Fall ist, dann genügt schon die Aufstellung der Basiskategorien, um eine gleichwertige Kategorisierung zu erlangen.

Zur Klärung dieser Frage sind Experimente notwendig, in denen eine Dokumentkollektion durch STC sowohl mit als auch ohne SLC kategorisiert wird. Anschließend gibt die Messung der Ähnlichkeit der Ergebnisse beider Varianten Aufschluss darüber, inwieweit diese Überlegungen zutreffen.

4 Austauschbarkeit der Dokumentmodelle

Das Dokumentmodell Suffix-Baum und, darauf aufbauend, STC ist, da keinerlei quantitatives Ähnlichkeitsmaß verwendet wird, auf den ersten Blick nicht mit Fusionierungsalgorithmen kompatibel, die hierauf aufbauen. Umgekehrt gilt dasselbe für das tfidf-Vektorraummodell, das scheinbar nicht zu dem in STC verwendeten Fusionierungsverfahren passt. In diesem Kapitel wird daher diskutiert, inwieweit die Austauschbarkeit beider Dokumentmodelle durch das jeweils andere Paradigma gegeben ist.

4.1 Quantisierung der Ähnlichkeit in den Dokumentmodellen

Im Unterschied zum Dokumentmodell Suffix-Baum wird im tfidf-Vektorraummodell ein Maß für die Ähnlichkeit zwischen zwei Dokumenten verwendet, die Kosinusähnlichkeit. Quantifiziert wird sie dadurch, dass der Kosinus des Winkels zwischen ihren tfidf-Vektoren berechnet wird. Hierbei ist allerdings zu berücksichtigen, dass dieser Wert nicht eindeutig bestimmt ist. Das heißt, dass ein Ähnlichkeitswert zwischen zwei Dokumenten nur unter bestimmten Voraussetzungen zutrifft, ansonsten aber auch einen völlig anderen Betrag haben kann.

Ursache hierfür ist die inverse Dokumenthäufigkeit. Der Betrag des Ähnlichkeitswertes zwischen zwei Dokumenten hängt stark von der Menge von Dokumenten ab, auf deren Basis der *idf*-Wert berechnet wurde. Es handelt sich also bei der Kosinusähnlichkeit zwischen zwei tfidf-Vektoren genau genommen um ein bedingtes Ähnlichkeitsmaß. Daher sind Ähnlichkeitswerte von Dokumenten aus verschiedenen Mengen auch nicht ohne weiteres vergleichbar. Um das zu gewährleisten, müsste beispielsweise eine einheitliche Datenbasis für alle *idf*-Werte verwendet werden.

STC berechnet keinen Wert, der die Ähnlichkeit zwischen zwei Dokumenten misst. Nichtsdestoweniger wäre ein solcher Wert wegen den oben genannten Kompatibilitätsgründen von großem Interesse. Dazu ist es allerdings notwendig, ein Ähnlichkeitsmaß auf Basis der zur Verfügung stehenden Daten zu definieren. Jedes Dokument ist im Suffix-Baum durch eine Reihe von Pfaden repräsentiert, die je ein Suffix repräsentieren. Zwei Dokumente werden also durch je einen Teilbaum des Suffix-Baums vollständig repräsentiert. Es ist daher ein Ähnlichkeitsmaß notwendig, das diese beiden Bäume vergleicht. Unter den in Kapitel 2, Abschnitt 1.1 aufgeführten Ähnlichkeitsmaßen ist beispielsweise das Überlappungsmaß für diese Aufgabe geeignet.

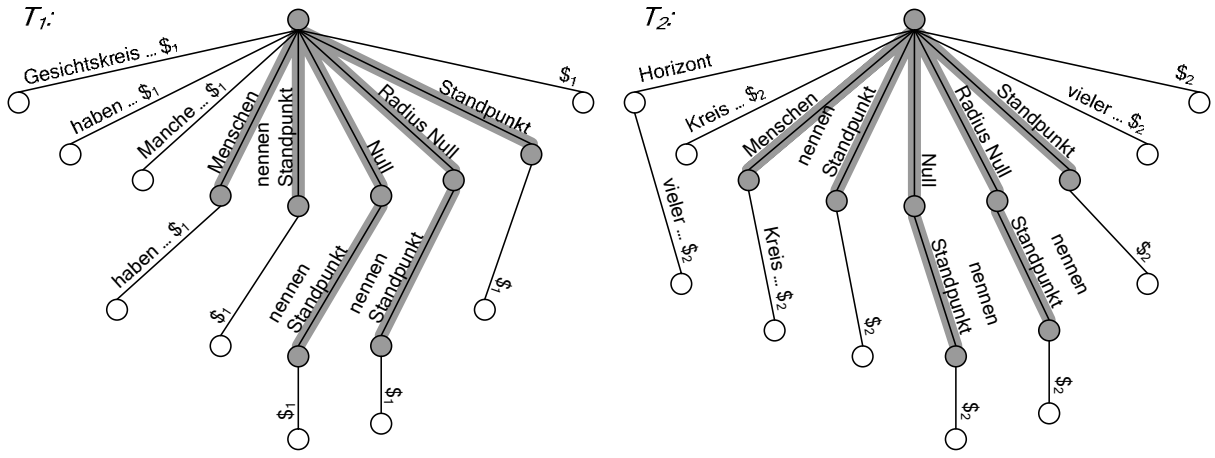

Abbildung 9: Die Teilbäume T_1 und T_2 der Dokumente d_1 und d_2 aus dem Suffix-Baum in Abbildung 6. Grau markiert sind die sich überlappenden Teile beider Bäume.

Die Idee dieses Maßes ist, wie sein Name schon sagt, die Überlappung der zu vergleichenden Bäume zu quantifizieren. Das geschieht hier beispielsweise durch die Berechnung des Verhältnisses der von den Bäumen geteilten Kanten zur minimalen Anzahl der Kanten eines der beiden Bäume. Für zwei Bäume T_i und T_j würde also folgende Formel verwendet:

$$sim(T_i, T_j) = \frac{|T_i \cap T_j|}{\min(|T_i|, |T_j|)}$$

Dieses Maß begünstigt kleine gegenüber großen Bäumen, da hiermit Bäume genau dann als identisch identifiziert werden, wenn ein Baum Teilbaum des anderen ist. Wenn anstatt des Minimums das Maximum der Kanten eines der beiden Bäume verwendet wird, so werden zwei Bäume nur dann als identisch identifiziert, wenn sie vollständig übereinstimmen.

Wird dieses Maß im Zusammenhang mit einem blattabschließenden Suffix-Baum verwendet, der auf Dokumenten beruht, denen Terminalwörter hinzugefügt wurden, so können zwei Dokumente nicht als vollständig identisch erkannt werden. Die Terminalwörter werden im Baum durch zusätzliche Kanten repräsentiert und mindern auf diese Weise die Ähnlichkeit zwischen zwei Dokumenten. Tabelle 6 stellt die mit dem Überlappungsmaß errechneten Ähnlichkeiten der Beispieldokumente aus Kapitel 2 für den in Abbildung 6 aufgestellten Suffix-Baum sowohl mit als auch ohne Berücksichtigung der Terminalwörter gegenüber. Dabei steht T_i mit $i \in \{1,2,3\}$ für den Teilbaum des Suffix-Baums, der das entsprechende Dokument d_i repräsentiert. Abbildung 9 zeigt die Teilbäume T_1 und T_2 sowie, grau markiert, die sich überlappenden Teile dieser Bäume. Die rechnerischen Ähnlichkeiten der Dokumente sind ohne Berücksichtigung der Terminalwörter erwartungsgemäß höher.

	Rechnerische Ähnlichkeit	
	mit TW	**ohne TW**
$sim(T_1,T_2)$	.44	.64
$sim(T_1,T_3)$	.22	.33
$sim(T_2,T_3)$	.33	.50

Tabelle 6: Gegenüberstellung der rechnerischen Ähnlichkeiten der Beispieldokumente aus Kapitel 2 nach Überlappungsmaß, repräsentiert durch einen Suffix-Baum jeweils mit und ohne Terminalwörter (TW).

Die Definition eines Ähnlichkeitsmaßes ermöglicht es, die in einem Suffix-Baum indizierten Dokumente mit jedem Fusionierungsalgorithmus zu kategorisieren, der auf der Verwendung solcher Maße beruht. Es bleibt zu klären, ob die Verwendung dieses Dokumentmodells zusammen mit einem Ähnlichkeitsmaß gegenüber den klassischen Modellen wesentliche Vorteile bringt, sowohl aus Effizienzerwägungen als auch aus Gründen der Aussagekraft der ermittelten Werte.

4.2 Approximation von STC mittels des Vektorraummodells

Die Austauschbarkeit des Vektorraummodells durch das in STC verwendete Paradigma lässt sich nicht über ein anderes Ähnlichkeitsmaß herstellen. Es ist jedoch möglich, unter bestimmten Voraussetzungen die Ergebnisse von STC zu approximieren. Dazu ist

eine gänzlich andere Auswertung der tfidf-Vektoren notwendig. Sie wird im folgenden Abschnitt zunächst theoretisch vorgestellt und anschließend anhand eines Beispiels demonstriert. Im dritten Abschnitt wird die Qualität der Approximation differenziert bewertet.

4.2.1 Auswertung der tfidf-Vektoren

Für eine Ausgangsmenge *D* von Dokumenten, wobei *W* die Menge aller in den Dokumenten verwendeten Wörter und *w* ein beliebiges Wort aus *W* ist, sei *V* die Menge aller tfidf-Vektoren.

Jede Dimension der Vektoren repräsentiert je ein Wort *w* aus *W*. Jedes *w* wiederum ist das erste Wort von mindestens einem Suffix der Dokumente aus *D*. Außerdem gilt für alle *w*, für die mehr als ein bzw. kein tfidf-Wert ungleich 0 ist, dass sie in mehr als einem Dokument enthalten sind. Ersterer Fall erschließt sich aus der Tatsache, dass ein tfidf-Wert für ein *w* in einem Dokumentvektor nur dann ungleich 0 sein kann, wenn das zugehörige Dokument *w* enthält. Letzterer Fall dagegen ergibt sich daraus, dass der *idf*-Wert eines Wortes, das in allen Dokumenten enthalten ist, per Definition gleich dem Logarithmus von 1, also 0, sein muss und damit auch in jedem Dokumentvektor der entsprechende Eintrag. Sind also für ein *w* in allen tfidf-Vektoren Nullen, so kommt das Wort in allen Dokumenten vor.

Daraus folgt, dass diese Wörter Präfix von mehr als einem Suffix aus unterschiedlichen Dokumenten sind. In einem blattabschließenden Suffix-Baum für *D* existiert dann ein Knoten *k* mit Tiefe 1, der kein Blattknoten ist und für den die Kantenbeschriftung seiner Eingangskante mit *w* beginnt.

Dieser Knoten wäre im STC eine Basiskategorie und enthielte all die Dokumente aus *D*, die Suffixe enthalten, welche mit *w* beginnen. Im Umkehrschluss können alle Basiskategorien, die für eine Ausgangsmenge *D* im STC identifiziert würden und deren Knotenrepräsentanten im Suffix-Baum die Tiefe 1 haben, aus den tfidf-Vektoren *V* identifiziert werden. Dokumente, die für ein beliebiges aber festes *w* mehr als einen bzw. keinen tfidf-Wert ungleich 0 haben, sind dafür zu je einer Basiskategorie zusammenzufassen.

Diese Basiskategorien dienen dann als Knotenmenge für einen Ähnlichkeitsgraph, dessen *sim*-Funktion der von STC entspricht. Mittels SLC werden daraus, analog zu STC, die endgültigen Kategorien ermittelt.

Die Auswertung der tfidf-Vektoren nach diesem Schema beruht nicht mehr auf dem Termgewichtsmaß tfidf, da für die Aufstellung von Basiskategorien ausschließlich das Vorhandensein von Wörtern in mehreren Dokumenten geprüft wird. Ein boolesches Termgewichtsmaß wäre an dieser Stelle ebenfalls verwendbar.

4.2.2 Beispiel für die Approximation von STC

Zur Demonstration werden in diesem Abschnitt die aus Kapitel 2 bekannten Dokumente nach obigem Schema kategorisiert. Indiziert werden die Dokumente daher mit dem tfidf-Vektorraummodell. Die entsprechenden Dokumentvektoren sind vollständig in Kapitel 2, Tabelle 3 und ausschnittsweise in Abbildung 10 zu sehen.

Die Abbildung zeigt verschiedene, für die Identifizierung von Basiskategorien relevante Dimensionen. Beispielsweise würde sich für die Dimension „Nennen“ eine Basiskategorie aus den Dokumenten 1 und 2 zusammensetzen, da die entsprechenden tfidf-Vektoren hier jeweils einen Wert ungleich 0 aufweisen, der Vektor für Dokument 3 jedoch nicht. Für die Dimension „Radius“ ergibt sich ebenfalls eine Basiskategorie, die alle Dokumente enthält, da hier alle Einträge der tfidf-Vektoren 0 sind. Entsprechend werden auch Basiskategorien für die Dimensionen „Null“ und „Standpunkt“ sowie – in der Abbildung nicht zu sehen – „Horizont“ und „Menschen“ gebildet.

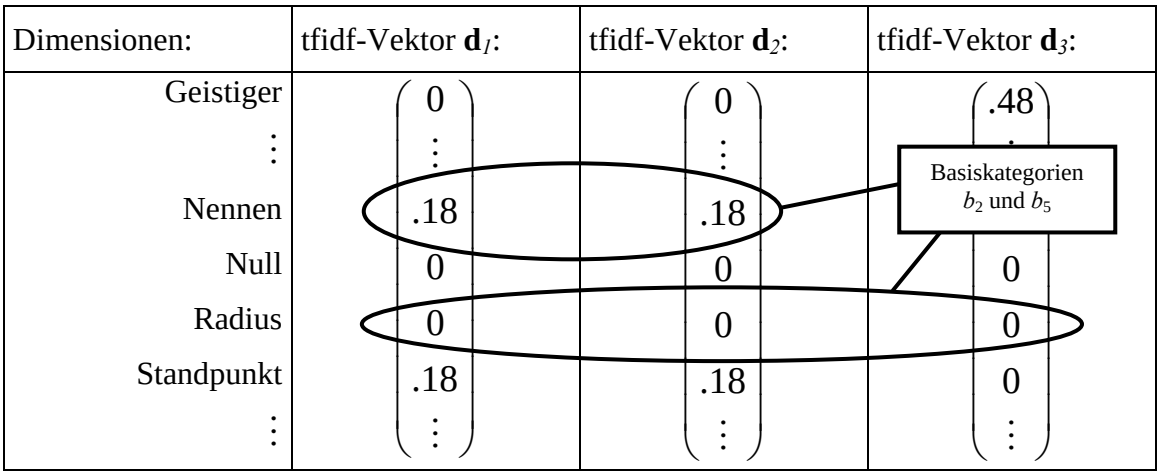

Abbildung 10: Ausschnitt aus den tfidf-Vektoren aus Kapitel 2, Tabelle 3 mit u.a. zwei elliptisch markierten Basiskategorien.

Die Knotenrepräsentanten dieser Basiskategorien wären in einem Suffix-Baum, der diese Dokumente indiziert, in Tiefe 1 zu finden. Die Basiskategorien aus tieferen Ebenen des Suffix-Baums können aus den tfidf-Vektoren jedoch nicht extrahiert werden, da sie keine Informationen über die Reihenfolge der Wörter im Dokument enthalten. Ohne Informationen über die Reihenfolge der Wörter kann allerdings keine Aussage darüber gemacht werden, ob die Dokumente nicht nur einzelne Wörter, sondern auch Wortketten teilen.

Abbildung 11 zeigt den Ausschnitt des Suffix-Baums, über den Informationen gewonnen werden können. Knoten, die sich aufgrund der extrahierten Informationen in tieferen Ebenen garantiert verzweigen, sind grau markiert. Bei allen anderen Knoten, respektive Worten, ist bekannt, dass Suffixe, die mit dem jeweiligen Wort beginnen, einmalig in *D* sind. Daher müssen es Blattknoten sein, deren Eingangskantenbeschriftungen beginnend mit dem jeweiligen Wort zum Ende des jeweiligen Dokuments reichen. Die Beschriftungen der Eingangskanten der grau markierten Knoten sind, bis auf das erste Wort, unklar, hier angedeutet durch das jeweilige Wort, gefolgt von drei Punkten und einem Fragezeichen.

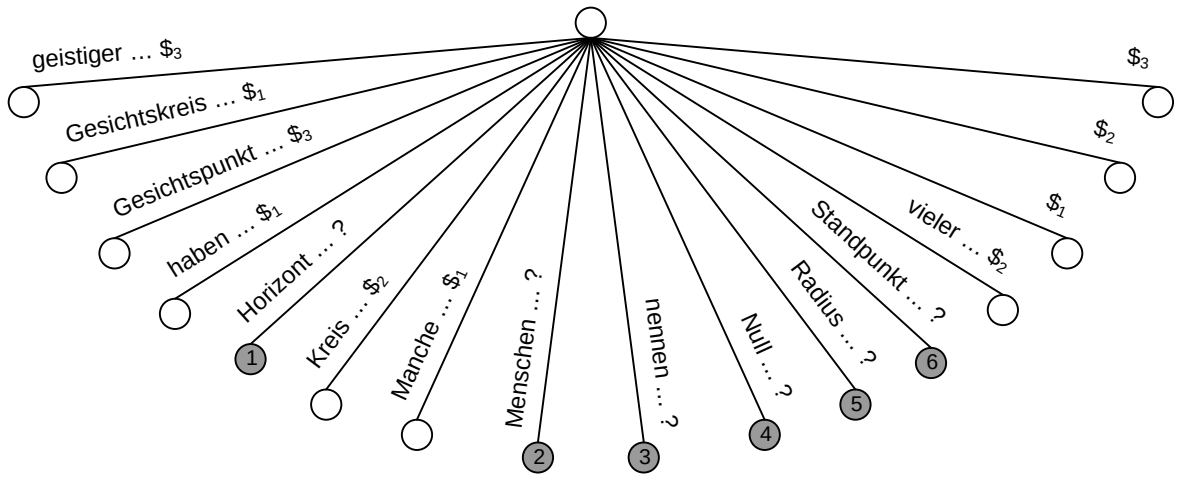

Abbildung 11: Suffix-Tree, der auf Grundlage der tfidf-Vektoren aus Abbildung 10 extrahiert werden kann.

Die Aufstellung des Suffix-Baums selbst ist allerdings für die Kategorisierung nicht notwendig. Da alle Basiskategorien, die aus tfidf-Vektoren analog zu Suffix-Bäumen

herausgefiltert werden können, bereits bekannt sind, kann direkt ein Ähnlichkeitsgraph aufgestellt werden, der in Abbildung 12 zu sehen ist.

$b_1 = \{d_2,d_3\}$
$b_2 = \{d_1,d_2\}$
$b_3 = \{d_1,d_2\}$
$b_4 = \{d_1,d_2,d_3\}$
$b_5 = \{d_1,d_2,d_3\}$
$b_6 = \{d_1,d_2\}$

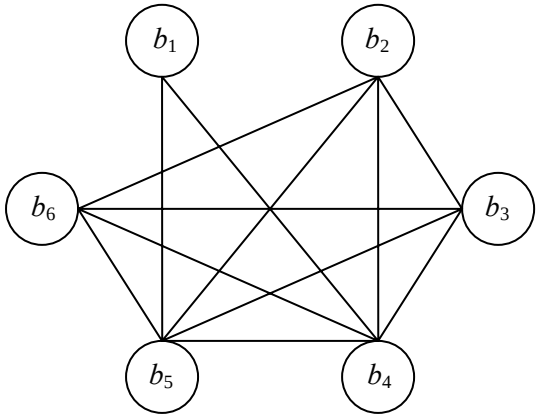

Abbildung 12: Ähnlichkeitsgraph für die aus tfidf-Vektoren extrahierten Basiskategorien.

An dieser Stelle erfolgt analog zum in Kapitel 2 demonstrierten SLC die Auswertung des Ähnlichkeitsgraphen. Es ist zu sehen, dass auch hier nur eine einzige Kategorie erzeugt wird, da der Graph sehr dicht ist und daher nur eine Zusammenhangskomponente existiert.

Da nicht alle Basiskategorien aus den tfidf-Vektoren gewonnen werden können, die STC verwendet, ist das hier vorliegende Ergebnis nicht äquivalent zu dem von STC. Die Erlangung eines äquivalenten Ergebnisses ist auch nicht möglich, sofern für eine Dokumentenmenge Basiskategorien existieren, die STC aus tieferen Ebenen als der ersten des Suffix-Baums gewinnt.

4.2.3 Bewertung der Approximation

Bei der Kategorisierung vollständiger Dokumente ist die Approximation von STC durch die alternative Auswertung des tfidf-Vektorraummodells nicht als gut zu bewerten. Der Grund dafür ist, dass die gewonnenen Basiskategorien der ersten Ebene des Suffix-Baums, wie in Kapitel 3.4 geschildert wurde, sehr viele Dokumente umfassen. Diese Basiskategorien würden von STC durch die dort getroffene Auswahl der vielversprechendsten Basiskategorien vernachlässigt. Die Ergebnisse von STC würden sich daher substanziell von denen der Approximation unterscheiden und wären qualitativ besser.

Dennoch gibt es auch Fälle, in denen die Approximation näher an den Ergebnissen von STC liegt, nämlich bei der Kategorisierung von Schnipseln. Hier sind die Basiskategorien der ersten Ebene des Suffix-Baums aussagekräftiger und es gibt zudem wenigere, die aus tieferen Ebenen gewonnen werden. Das hat zur Folge, dass die durch die Approximation gewonnenen Basiskategorien mit größerer Wahrscheinlichkeit auch von STC verwendet werden und auf diese Weise die Ergebnisse beider ähnlicher sind.

5 Zusammenfassung und Ausblick

Die Ergebnisse dieser Ausarbeitung lassen sich anhand der Fragestellung, ob sich das Dokumentmodell Suffix-Baum auch losgelöst von Suffix-Baum-Kategorisierung verwenden lässt wie folgt zusammenfassen.

Für das Dokumentmodell Suffix-Baum wird gezeigt, dass die Definition eines Ähnlichkeitsmaßes, das es erlaubt, die inhaltliche Übereinstimmung zwischen zwei im Modell indizierten Dokumenten zu quantifizieren, ausreicht, um Kompatibilität zu anderen Fusionierungsalgorithmen herzustellen. Welcher Art das Ähnlichkeitsmaß ist, und wie plausibel die Werte sind, die es errechnet, bleibt weiteren Studien überlassen.

Betreffend Suffix-Baum-Kategorisierung wird gezeigt, dass das Verfahren eine Menge von Dokumenten unter Umständen unvollständig kategorisiert. Das bedeutet, dass es bei einer Konstellation, in der Dokumente nur wenig oder gar keine Ähnlichkeit zu anderen aufweisen, dazu kommen kann, dass sie in den erzeugten Kategorien nicht enthalten sind. Suffix-Baum-Kategorisierung ist daher unvollständig zu nennen. Eine annähernd vollständige Kategorisierung, bis auf einen Grenzfall, gelingt dann, wenn auf die Auswahl der vielversprechendsten Basiskategorien im Fusionierungsalgorithmus verzichtet wird.

Des Weiteren wird gezeigt, dass die Auswahl der vielversprechendsten Basiskategorien grundlegend auf den Gedanken beruht, die auch die Grundlage für die inverse Dokumenthäufigkeit bilden. Es sollen damit Wörter unberücksichtigt bleiben, die in sehr vielen Dokumenten enthalten sind. Je nachdem, welche Art von Dokumenten kategorisiert wird, hat der Verzicht auf die Auswahl der vielversprechendsten Basiskategorien zur Wahrung vollständiger Kategorisierung, unterschiedliche Auswirkungen. Bei vollständigen Dokumenten wird die Qualität der Kategorisierung schlechter, wohingegen es bei Schnipseln keinen Unterschied macht.

Neben der Auswertung direkter Ähnlichkeiten zwischen Dokumenten durch die Aufstellung des Suffix-Baums, wird gezeigt, dass STC auch transitive Ähnlichkeiten verwendet, die durch die Aufstellung des Ähnlichkeitsgraphen mit der sim_{STC}-Funktion ermittelt werden. Dies macht für die Kategorisierung von Dokumenten jedoch weder Sinn, noch ist erkennbar, dass solcherlei Ähnlichkeiten in größerem Umfang auftreten. Zu sehen ist ferner, dass die eigentliche Kategorisierung in STC entgegen dem Anfangs vorgestellten Schema nicht im zweiten Schritt, der Fusionierung, stattfindet, sondern vielmehr durch die Aufstellung der Basiskategorien, die auf dem Dokumentmodell Suffix-Baum beruhen.

Diese Erkenntnisse führen dazu, dass die Ergebnisse von STC unter der Vorraussetzung vollständiger Kategorisierung von Schnipseln durch eine modifizierte Variante von TBC approximierbar sind. Dies wird sowohl formal, als auch anhand eines Beispiels gezeigt.

Die Frage nach dem jeweils stärkeren Ansatz zur Kategorisierung bleibt dagegen offen, da STC qualitativ unterschiedliche Ergebnisse erzeugt, je nachdem welche Form von Dokumenten kategorisiert werden. Eine quantitative Analyse beider Verfahren könnte hier Aufschluss darüber geben, wie sich die Kategorisierungsergebnisse anderer Formen von Dokumenten zueinander verhalten.

A Literatur

[1] Baeza-Yates, Ricardo / Ribeiro-Neto, Berthier: Modern Information Retrieval. 1999.

[2] Ferber, Reginald: Information Retrieval. Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web. 2003. Abrufbar im Internet. URL: http://information-retrieval.de/irb/ir.html. Stand: 19.12.2004.

[3] Luhn, Hans Peter: A Statistical Approach to Mechanized Encoding and Searching of Literary Information. In: IBM Journal of Research and Development, 1. Jg. 1957, Heft 4,
S. 309-317.

[4] Luhn, Hans Peter: The Automatic Creation of Literature Abstracts. In: IBM Journal of Research and Development, 2. Jg. 1958, Heft 2, S. 159-165.

[5] McCreight, Edward M.: A Space-Economical Suffix Tree Construction Algorithm. In: Journal of the ACM, 23. Jg. 1976, Heft 2, S. 262-272.

[6] Meyer zu Eißen, Sven / Stein, Benno: Analysis of Clustering Algorithms for Web-based Search. In: Practical Aspects of Knowledge Management: 4th International Conference, PAKM 2002 Vienna, Austria, 2002, S. 168-178. Abrufbar im Internet. URL: http://www.springerlink.com/openurl.asp?genre=issue&issn=0302-9743&volume=2569. Stand: 19.12.2004.

[7] Popescul, Alexandrin / Ungar, Lyle H.: Automatic Labeling of Document Clusters. 2000. URL: http://citeseer.ist.psu.edu/popescul00automatic.html. Stand: 04.10.2004.

[8] Potthast, Martin: Seminararbeit über die Datenstruktur Suffix-Tree. Unveröffentlichte Seminararbeit, Paderborn 2004.

[9] Salton, Gerard / McGill, Michael J.: Introduction to Modern Information Retrieval. 1983.

[10] Singhal, Amit / Salton, Gerard: Automatic Text Browsing Using Vector Space Model. In: Proceedings of the Dual-Use Technologies and Applications Conference, 1995,
S. 319-324.

[11] Stein, Benno / Meyer zu Eißen, Sven: AISearch: Category Formation of Web Search Results. 2003. URL: http://www-ai.upb.de/aisearch/ir03-aisearch-frame.pdf. Stand: 06.10.2004.

[12] Stein, Benno / Meyer zu Eißen, Sven: Topic Identification: Framework and Application. In: Proceedings of the 4th International Conference on Knowledge Management
(I-KNOW 04), Graz, Austria, Journal of Universal Computer Science, 2004, S. 256-269.

[13] Stein, Benno / Meyer zu Eißen, Sven: Automatische Kategorisierung für Web-basierte Suche. Einführung, Techniken und Projekte. In: KI – Künstliche Intelligenz: Special Issue on Adaptive Multimedia Retrieval, 2004.

[14] Ukkonen, Eso: On-line construction of suffix trees. In: Algorithmica, 14. Jg. 1995, Heft 3, S. 249-260.

[15] van Rijsbergen, C. J.: Information Retrieval. 2nd Edition. London 1979.

[16] Weiner, Peter: Linear Pattern Matching Algorithms. In: Proceedings of 14th IEEE Annual Symposium on Switching and Automata Theory, 1973, S. 1-11.

[17] Zamir, Oren Eli / Etzioni, Oren: Web Document Clustering: A Feasibility Demonstration. In: Proceedings of the 21st SIGIR Conference, 1998, S. 46-54.

[18] Zamir, Oren Eli: Clustering Web Documents: A Phrase-based Method for Grouping Search Engine Results. Dissertation, Washington 1999.

[19] Mermin, N. David: Ein Zitat sucht einen Autor. In: Die Zeit, Nr. 34 vom 12.08.2004,
S. 35.

B Abbildungen

C Tabellen