Personalisiertes Hierarchisches Strukturieren

Korinna Bade

Umweltbundesamt, FG IV 2.1; Wörlitzer Platz 1; 06844 Dessau-Roßlau
korinna.bade@googlemail.com

Abstract: Die Organisation einer wachsenden Menge an Daten sowie die Extraktion von relevanten Informationen daraus ist eine aktuelle Fragestellung von steigender Bedeutung. Die Dissertation leistet dazu einen Beitrag in Richtung einer (teil-) automatisierten Unterstützung dieser Prozesse durch personalisiertes, hierarchisches Strukturieren. Dazu wurden verschiedene Data Mining Methoden entwickelt, die eine Datensammlung automatisch in eine Hierarchie strukturieren können. Dabei werden nutzerspezifische Anforderungen für die Generierung der geeignetsten Struktur berücksichtigt. Diese nur implizit vorliegenden Strukturierungspräferenzen können sowohl die gesamte Zielhierarchie als auch nur Teile davon beschreiben. Anhand von verschiedenen Benchmarkdatensätzen konnte die hohe Qualität der entwickelten Algorithmen gezeigt werden.

1 Motivation und Problemstellung

Das Internet ermöglicht uns heutzutage Zugriff auf eine fast unbegrenzte Menge an Informationen. Effektive Suchsysteme sind daher eine Notwendigkeit, um die gerade benötigten Informationen zu erreichen. Die heutigen, Stichwort basierten Suchmaschinen können bereits viele verschiedene Suchanfragen angemessen unterstützen. Allerdings gilt dies insbesondere für ”leichte” Suchaufgaben, wo der Nutzer sowohl genau weiß, was er finden möchte, als auch welche Stichwörter ihn zum Ziel bringen. Probleme treten zum Beispiel bei mehrdeutigen Suchbegriffen auf, insbesondere wenn die populärste Bedeutung nicht die vom Nutzer gewünschte ist. Beim Finden von neuen, zum Ergebnis führenden Suchbegriffen steht der Nutzer jedoch alleine da, obwohl diese Aufgabe häufig nicht trivial ist. Ungleich schwieriger wird das Suchen, wenn das Ziel eine Recherche zu einem bestimmten Thema ist. Der Nutzer weiß hier meist noch gar nicht genau, was er eigentlich finden möchte und muss sich zunächst ein Bild über das Thema machen. Daher ist es sehr schwierig, adäquate Suchanfragen zu formulieren. Beispiele für solche Rechercheaufgaben sind die Suche eines Wissenschaftlers nach interessanten verwandten Arbeiten oder die Recherche eines Journalisten für einen Artikel.

Daher war es Ziel der Dissertation [Bad09], Methoden zu entwickeln, mit denen solche Rechercheaufgaben besser unterstützt werden können. Insbesondere soll dies durch eine hierarchische Strukturierung von Dokumentsammlungen (die z.B. das Ergebnis einer noch vagen, Stichwort basierten Suche sein können) erreicht werden. Hierarchien ermöglichen einen guten Überblick über größere Datenmengen und werden daher bereits in vielen Be-

reichen erfolgreich eingesetzt. Ihr Nachteil ist der große Aufwand, der für den Aufbau und die Pflege nötig ist. Daher sind das Ziel dieser Arbeit (teil-)automatische Methoden, die den Aufwand für den Nutzer verringern können. Dabei sollen sich diese Methoden an die jeweiligen Vorlieben eines bestimmten Nutzers anpassen können, damit dieser optimal mit den aufbereiteten (strukturierten) Daten umgehen kann.

Damit Methoden Daten bzw. Dokumente automatisch, personalisiert, hierarchisch strukturieren können, müssen die Strukturierungspräferenzen des Nutzers bekannt sein und der Methode zugänglich gemacht werden. In der Dissertation sollte dabei auf implizit vorhandenes Wissen zugegriffen werden, um den zusätzlichen manuellen Aufwand des Nutzers möglichst gering zu halten. Insbesondere wird auf Daten zurückgegriffen, die der Nutzer in der Vergangenheit bereits hierarchisch strukturiert hat. Der Aufbau solcher Hierarchien ist meist Teil der täglichen Arbeit, zum Beispiel beim Organisieren von Daten im Dateisystem des Rechners oder beim Verwalten von Bookmarks. Zusätzlich wird in Abschnitt 3 eine allgemeinere Form der Darstellung solcher Präferenzen vorgestellt und verwendet.

Daraus ergibt sich die folgende allgemeine Lernaufgabe, die von den entwickelten Algorithmen zu lösen ist: Aufgabe ist es, eine Dokumentensammlung hierarchisch zu strukturieren unter der Berücksichtigung von Strukturierungspräferenzen eines Nutzers. Dabei werden zwei verschiedene Arten von Nutzerpräferenzen unterschieden. Zunächst werden Strukturierungspräfenzen in Form einer festen Zielhierarchie untersucht (Abschnitt 2). In diesem Fall müssen die Methoden die Dokumente möglichst gut in die gegebene Hierarchie einsortieren. Es handelt sich somit um eine Klassifikationsaufgabe aus dem Maschinellen Lernen. Danach wird diese Anforderung aufgeweicht und nur noch angenommen, dass Präferenzen für einen Teil der gesamten, in den Daten existierenden (aber unbekannten) Hierarchie gegeben sind (Abschnitt 3). Wird dies auch durch eine gegebenen Hierarchie dargestellt, so ist es Aufgabe der Algorithmen, diese Hierarchie basierend auf den Daten angemessen zu erweitern und die Dokumente in diese erweiterte Hierarchie einzusortieren. Das ist dem teilüberwachten Clustering zuzuordnen und wird in Abb. 1 dargestellt. Auf der rechten Seite wird die zu strukturierende Datensammlung gezeigt, die in die Nutzerhierarchie auf der linken Seite einsortiert wird, wobei die roten Knoten die automatische Hierarchieerweiterung durch die Algorithmen darstellen. Diese Knoten entfallen bei der ersten Art von Präferenzen, bei denen die Hierarchie nicht verändert werden darf.

Abbildung 1: Hierarchisches Strukturieren

Bei beiden Arten der Strukturierungspräferenzen müssen die Algorithmen damit umgehen können, dass die zu strukturierende Dokumentensammlung Dokumente enthält, die bisher nicht adäquat in der gegebenen Nutzerhierarchie berücksichtigt werden, da sie zum Beispiel zu Themen gehören, für die sich der Nutzer bisher noch nicht interessiert hat. Diese Anforderung ist von hoher praktischer Relevanz, da man nie davon ausgehen kann, dass eine manuell erstellte Hierarchie alle auftretenden Themen abdecken kann. Gleichzeitig schließen die meisten existierenden Verfahren diesen Fall aus bzw. untersuchen ihn nicht, so dass hierauf besonders Wert gelegt wurde.

Im Folgenden wird zunächst in Abschnitt 2 auf Algorithmen der hierarchischen Klassifikation für feste Zielhierarchien eingegangen. Danach werden in Abschnitt 3 Methoden des teilüberwachten, hierarchischen Clusterings für das allgemeinere Szenario vorgestellt. Und schließlich diskutiert Abschnitt 4 die praktische Anwendbarkeit der in der Dissertation entwickelten Methoden.

2 Hierarchische Klassifikation

Wie bereits im vorherigen Abschnitt erwähnt, soll zunächst auf den Fall einer fest vorgegebenen Zielhierarchie eingegangen werden. Benötigt wird dazu ein Algorithmus, der Dokumente möglichst gut in diese Hierarchie einsortieren kann. Doch was bedeutet hier ”möglichst gut”? Während Standardevaluierungsmaße teilweise recht abstrakte Kennzahlen für eine Bewertung heranführen, steht hier die vom Nutzer wahrgenommene Nützlichkeit der Klassifikation im Vordergrund. Das bedeutet, dass er beim Navigieren durch die Hierarchie möglichst schnell die richtigen Dokumente findet. Unter der Annahme, dass es für jedes Dokument genau einen Ordner gibt, zu dem es im Idealfall zugeordnet sein sollte (aus Sicht des jeweiligen Nutzers), ist natürlich eine automatische Klassifikation in diesen Ordner das bestmögliche Verhalten des Algorithmus. Das bedeutet aber nicht zwangsläufig, dass dies die einzige Stelle ist, an der der Nutzer nach relevanten Dokumenten suchen würde, insbesondere wenn die Dokumente dieses Ordners nicht ausreichend sind für die Erfüllung seines Informationsbedürfnisses. So wird der Nutzer z.B. sehr wahrscheinlich auch in einer allgemeineren Ebene nach relevanten Dokumenten suchen, da ja auch allgemeinere Dokumente Informationen zu einem speziellen Teilgebiet enthalten können. Eine automatische Klassifikation in einen solchen Ordner kann somit immer noch als teilweise nützlich angesehen werden, da der Nutzer die entsprechende Dokumente immer noch finden kann, wenn auch mit höherem Zeitaufwand. Aus algorithmischer Sicht kann eine solch generellere Einordnung sinnvoll sein, wenn es nicht genügend Anhaltspunkte für eine konkrete Einordnung gibt und eine Fehlklassifikation so vermieden werden kann. Eine Klassifikation in völlig andere Bereiche der Hierarchie birgt hingegen keine Nützlichkeit mehr für den Nutzer, da er dort auch nie nach dem Dokument suchen würde, und muss somit durch den Algorithmus vermieden werden.

Es soll hier noch angemerkt werden, dass es für ein Dokument immer einen optimalen Ordner in der Hierarchie gibt, auch wenn das eigentliche Thema noch nicht direkt in der Hierarchie repräsentiert ist. Der beste Ordner ist dann allgemeiner als das Thema des Dokuments, aber dabei so speziell wie die Hierarchie erlaubt. Im schlimmsten Fall ist das der

Wurzelknoten der Hierarchie, wenn das Thema noch gar nicht von der Hierarchie abgedeckt wird.

Die Dissertation untersucht im Wesentlichen drei verschiedene Aspekte für diese Problemstellung der hierarchischen Klassifikation. Zunächst wird auf das Problem der unvollständigen Hierarchie eingegangen, d.h. wie mit Dokumenten umgegangen werden muss, deren Thema in der Hierarchie noch nicht adäquat abgebildet ist. Hier wurde zunächst das gewünschte Verhalten einer automatischen Methode aufgezeigt (ähnlich wie oben angerissen). Danach wurden die speziellen Probleme analysiert, die thematisch neue Dokumente bei einer späteren Klassifikation mit sich bringen, und Lösungsvorschläge diskutiert. Insbesondere betrifft das die Beschreibung der Dokumente durch eine Menge von Eigenschaften, die sich bei Dokumenten in der Regel aus einer Menge von Worten/Termen verbunden mit einer entsprechenden Gewichtung ergeben. Die wesentlichen Terme eines thematisch neuen Dokuments sind höchstwahrscheinlich noch gar nicht Teil des Klassifikationsmodells, so dass bei seiner Anwendung eine Klassifikationsentscheidung sehr wahrscheinlich auf Grund von sehr wenigen, eher unwesentlichen Termen erfolgt, was zu starken Verzerrungen führen kann. Hierfür wurde aufgezeigt, wie neue Terme sinnvoll in das Klassifikationsmodell eingebettet werden können.

Im zweiten Teil werden in der Dissertation Methoden entwickelt, die die zuvor untersuchten Sachverhalte aufgreifen und unter Berücksichtigung der Hierarchie klassifizieren. Im Gegensatz dazu betrachten existierende Standardklassifikatoren in der Regel nur flache Klassenstrukturen. Durch die hierarchischen Beziehungen zwischen den Klassen kann jedoch zusätzliche Information gewonnen und bei der Klassifikation genutzt werden. Die Dissertation führt dazu das Konzept der Vorhersagenützlichkeit ein, mit dem festgelegt werden kann, wie hoch die Nützlichkeit einer Klassifikation für den Nutzer ist, in Abhängigkeit von der tatsächlichen und der vorhergesagten Klasse. Im Allgemeinen kann das von Nutzer zu Nutzer sehr individuell sein. Basierend auf den vorherigen Betrachtungen wurde aber eine für die oben beschriebene Strategie gültige Funktion definiert, die durch Parameter an den individuellen Nutzer angepasst werden kann. Mit diesem Konzept wurden dann zwei verschiedene, hierarchische Klassifikationsmethoden entwickelt, CUClass und HUClass. Ziel beider Methoden war es, etablierte Standardmethoden wie Support-Vektor-Maschinen [Joa02] universell, also unabhängig von der eigentlichen Methode, mit der Vorhersagenützlichkeit zu kombinieren, anstatt die Klassifikatoren selbst zu verändern, wie z.B. in [CH04]. Die CUClass-Methode versucht dabei eine für den Nutzer nicht nützliche Klassifikation zu vermeiden, indem sie mittels Vorhersagegenauigkeit des Klassifikators die Klassifikationswahrscheinlichkeiten auf verschiedenen Hierarchieebenen miteinander abwägt. Der Klassifikator durchläuft die Hierarchie dabei von oben nach unten und entscheidet auf jeder Hierarchieebene, ob noch eine spezifischere Klassifikation vorgenommen werden kann oder ob die Wahrscheinlichkeit einer falschen, nicht mehr nützlichen Klassifikation zu groß ist. Während bei CUClass die Vorhersagenützlichkeit nur implizit durch die Methodik genutzt wird, integriert HUClass diese direkt. Mittels Entscheidungstheorie [vNM53] wird die Vorhersagenützlichkeit direkt mit den Vorhersagewahrscheinlichkeiten des Klassifikators verknüpft und die Klasse mit der höchsten, erwarteten Nützlichkeit vorhergesagt.

Zur Ermittlung der Qualität eines Klassifikators existieren verschiedene Standardmaße.

Diese konzentrieren sich aber auch auf eine flache Klassifizierung. Da es jedoch Ziel der entwickelten Algorithmen war, die Nützlichkeit für einen Nutzer im hierarchischen Szenario zu verbessern, wurden im dritten Teil diese Maße so erweitert, dass sie die Vorhersagenützlichkeit direkt bewerten können. Mit diesen Maßen wurden die entwickelten Methoden mit verschiedenen Textdatensätzen und verschiedenen Szenarien, die einen unterschiedlichen Anteil an unbekannten Klassen und Menge der Trainingsdaten abdecken, untersucht. Dabei hat sich gezeigt, dass die Algorithmen im Vergleich zur jeweiligen Standardmethode um so mehr Qualitätsgewinn erbringen, je schwieriger das Klassifikationsproblem wird. Ist die Hierarchie vollständig und die Anzahl der Trainingsbeispiele hoch, bringen beide Algorithmen kaum Verbesserung zur ohnehin schon guten Standardklassifikation. Gibt es aber nur sehr wenig Trainingsdaten oder deckt die Hierarchie neue Dokumente nicht ausreichend ab, zeigen sich die Stärken von CUClass und HUClass. Und genau diese Situationen entsprechen der Realität, was die praktischer Relevanz beider Methoden zeigt. Im direkten Vergleich gibt es keine generell bessere Methode. Stattdessen muss das speziell für den jeweiligen Anwendungsfall abgewägt werden.

3 Teilüberwachtes, Hierarchisches Clustering

Im Folgenden soll nun gezeigt werden, mit welchen Methoden eine unvollständige Hierarchie für neue Dokumente automatisch erweitert werden kann. Es soll also eine hierarchische Strukturierung der Dokumente vorgenommen werden, wobei eine existierende Hierarchie als Richtlinie für die Ergebnisstruktur dient. Für flache Clusterstrukturen, bei denen Daten zwar auch gruppiert werden, es aber keine hierarchischen Beziehungen zwischen diesen Gruppen gibt, wurden in der Vergangenheit verschiedene Varianten des bedingten Clusterings vorgeschlagen, erstmals in [WCRS01]. Hierbei wird Wissen über die Clusterstruktur mittels Must-Link- und Cannot-Link-Bedingungen modelliert, die jeweils angeben, ob zwei Objekte zum gleichen oder zu unterschiedlichen Clustern gehören, nicht aber um welche Klassen es sich dabei handelt. Das hat den Vorteil, dass kein explizites Wissen über tatsächliche Klassen erforderlich ist, was die paarweisen Bedingungen allgemeiner anwendbar macht. In Clusterhierarchien stehen Objekte allerdings über mehrere Hierarchieebenen in Beziehung, so dass diese absolute Definition der gemeinsamen Clusterzugehörigkeit ungeeignet ist. Daher schlägt die Dissertation einen neuen, hierarchischen Bedingungstyp vor, die Must-Link-Before-Bedingungen (MLB). Diese fokussieren auf die hierarchische Reihenfolge, in der Objekte miteinander in Beziehung stehen. Anstatt von Paaren werden Objekttripel betrachtet: $MLB_{xyz} = (d_x; d_y; d_z)$. Diese besagen, dass die Objekte d_x und d_y auf einer spezifischeren Hierarchieebene mit einander verknüpft sind als die Objekte d_x und d_z wie in Abb. 2 skizziert. Für diese Bedingungen können formale Eigenschaften gezeigt werden, worauf hier aber nicht näher eingegangen werden soll.

Die verschiedenen Algorithmen für bedingtes Clustering lassen sich im Wesentlichen in zwei verschiedene Arten unterteilen, die Instanz basierten und die Metrik basierten Methoden. Die Instanz basierten Methoden nutzen die Bedingungen direkt während des Clusteringprozesses, z.B. bei der Initialisierung oder der Zuweisung von Objekten zu Clustern [KL02]. Hierbei gibt es Methoden, die alle Bedingungen zwingend erfüllen [WCRS01],

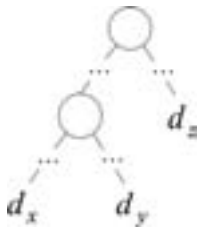

Abbildung 2: MLB-Bedingung in der Hierarchie

und Methoden, bei denen einige Bedingungen verletzt werden dürfen, falls dies sinnvoll erscheint [BBM04a]. Die Metrik basierten Methoden nutzen die Bedingungen, um eine Metrik zu lernen, die die durch die Bedingungen ausgedrückte Ähnlichkeit widerspiegelt. Diese Metrik wird dann beim Clustering eingesetzt. Meist wird die Metrik vor dem eigentlichen Clustering gelernt [BHHSW03]. Einige wenige Ansätze lernen die Metrik aber auch während des Clusterings [BBM04b], wodurch weiteres Wissens über die Bedingungen hinaus von den restlichen Daten beim Lernen berücksichtigt werden kann. Im Rahmen der Dissertation wurden drei verschiedene, hierarchische, bedingte Clustermethoden entwickelt, ein Instanz basiertes Verfahren (iHAC) sowie zwei Metrik basierte Verfahren, wobei eines (mHAC) vor und eines (biHAC) während des Clusterings die Metrik lernt. Des Weiteren wurde das Instanz basierte Verfahren mit den Metrik basierten Verfahren kombiniert. Alle Methoden basieren auf dem hierarchisch agglomerativen Clustering (HAC) und sollen im Folgenden kurz vorgestellt werden.

Durch ihre hierarchische Natur können MLB-Bedingungen problemlos mit einem Instanz basierten Verfahren in HAC integriert werden. Hierbei wird HAC so angepasst, dass nur noch Cluster verschmolzen werden, die dabei keine MLB-Bedingungen verletzen. Das bedeutet, dass für alle Bedingungen $(d_x; d_y; d_z)$ gelten muss, dass das Cluster, das d_z enthält, nur mit einem Cluster verschmolzen werden darf, das entweder sowohl d_x als auch d_y enthält oder keines von beiden. Das Bestimmen der gültigen Verschmelzungsoperationen ist von Laufzeit kritischer Bedeutung für den Algorithmus. Da die Zahl der Bedingungen exponentiell mit der Größe der verfügbaren Nutzerhierarchie und -daten anwächst, ist eine effiziente Prüfung der Bedingungen nötig. Hierfür wurde in Anlehnung an existierenden Untersuchung zu effizienten HAC-Implementierungen eine entsprechende Datenstruktur mit effizientem Algorithmus entwickelt, der eine schnelle Bestimmung der bestmöglichen Clusterverschmelzung erlaubt.

Ziel des Metriklernens ist die unterschiedliche Gewichtung der verschiedenen Eigenschaften, bei Dokumenten der verschiedenen Terme, so dass die vom Nutzer empfundene (implizite) Wichtigkeit der einzelnen Eigenschaften bei der Wahrnehmung von Ähnlichkeiten abgebildet werden kann. Dazu werden die Bedingungen als Ähnlichkeitsrelation aufgefasst. Die Bedingung $(d_x; d_y; d_z)$ drückt aus, dass d_x und d_y zueinander ähnlicher sein sollten als d_x und d_z, denn dann werden sie beim HAC auch zuerst zusammengefügt. Ist für eine Bedingung diese Ähnlichkeitsbeziehung durch das aktuelle Ähnlichkeitsmaß verletzt, müssen die Gewichtungen angepasst werden. Dies geschieht mit Hilfe eines Gradientenabstiegsverfahrens. Beim mHAC-Verfahren wird dabei vor dem eigentlichen Clus-

tering über alle Bedingungen iteriert und das Ähnlichkeitsmaß schrittweise angepasst, bis möglichst wenige Bedingungen noch verletzt werden. Im Gegensatz dazu wird das Metriklernen bei biHAC in den Verschmelzungsschritt des Clustering intergriert. Nur dann, wenn durch das eigentliche Clustering tatsächlich eine Bedingung verletzt wird, wird die Metrik angepasst. Dabei werden statt der spezifischen Einzelbedingungen die ganzen Cluster zu Grunde gelegt. So wird auch der Einfluss der restlichen Daten auf die Ähnlichkeitsbestimmung ermöglicht. Im Gegensatz zu mHAC kann biHAC mit deutlich weniger Metrikanpassungen eine häufig bessere Metrik lernen. Insbesondere hat sich gezeigt, dass biHAC in Kombination mit iHAC am besten mit den realen Anwendungsanforderungen umgehen kann, in der nur Teilhierarchien bekannt sind und entsprechend erweitert werden müssen. Durch die Berücksichtigung aller Daten ist biHAC besonders robust und bekannte Klassen können weniger über neue dominieren. So konnte durch den Einsatz von MLB-Bedingungen eine deutliche Qualitätsverbesserung im Vergleich zum Standard-HAC-Verfahren erreicht werden. Zur Evaluierung wurde neben Standardmaßen auch ein neues Maß entwickelt, dass einen Instanz basierten Hierarchievergleich ermöglicht und zwei Hierarchien sehr spezifisch vergleichen kann.

4 Anwendung in der Praxis

Das Hauptziel der Dissertation war die Entwicklung und Analyse von Methoden für die personalisierte, hierarchische Strukturierung. Im Folgenden soll diskutiert werden, wie sich diese in der Praxis für eine bessere Informationssuche und -organisation einsetzen lassen. Im Rahmen der Dissertation wurde dazu die Metasuchmaschine CARSA entwickelt. Dieses modulare Framework zielt auf die prototypische Demonstration verschiedener Komponenten für den Suchprozess und ermöglicht eine einfache Integration neuer Methoden. So konnten die entwickelten Strukturierungsalgorithmen zur Aufbereitung von Ergebnislisten zu einer Suchanfrage eingesetzt und so ihre prinzipielle Verwendbarkeit in der Praxis demonstriert werden (Abb. 3). Das verwendete Webinterface ist dabei eher rudimentär, da es nicht im Fokus der Arbeit stand. Stattdessen sollen ein paar interessante Anwendungsfälle skizziert werden, für die die entwickelten Strukturierungsalgorithmen eine wichtige Grundlage bilden.

Als Beispiel wird im Folgenden ein Wissenschaftler betrachtet, der bei der Verwaltung von wissenschaftlicher Literatur in einer Themenhierarchie unterstützt werden soll. Hierbei ist sowohl eine Unterstützung bei der Ablage von Literatur als auch bei der Suche nach neuen, relevanten Arbeiten von Interesse. Dabei wird das bereits existierende Ablagesystem des Wissenschaftlers verwendet, um seine Strukturierungspräferenzen für die Algorithmen zu modellieren. Bei Änderungen an seiner Ablage (z.B. durch Hinzufügen neuer Dokumente) muss das gelernte Modell entsprechend angepasst werden, wobei dies meist nicht sofort erfolgen muss, sondern kleinere Änderungen gesammelt werden können und das Modell bei verfügbarer Rechenleistung im Hintergrund aktualisiert werden kann. Die entwickelten Methoden können den Wissenschaftler nun bei verschiedenen Aufgaben unterstützen. So ermöglichen sie z.B. einen personalisierten Überblick über eine unbekannte, unstrukturierte Dokumentensammlung wie z.B. einen Konferenzband. Dazu strukturieren

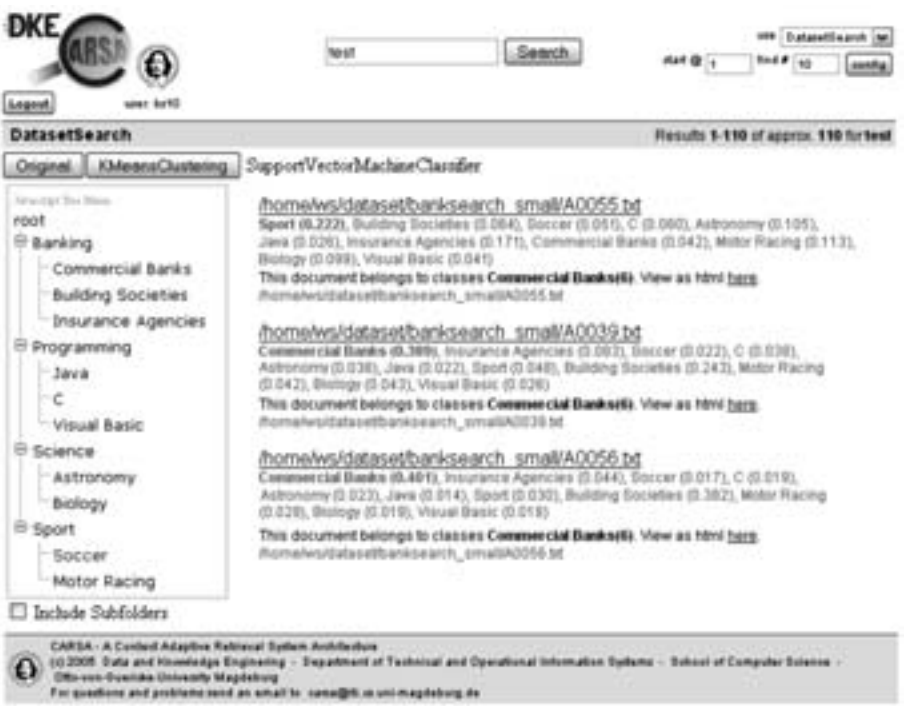

Abbildung 3: Screenshot von CARSA mit strukturierten Suchergebnissen

die Methoden die Sammlung Nutzer spezifisch und geben somit einen Überblick über abgedeckte Nutzerthemen in der Sammlung und die jeweils dazu gehörenden Dokumente. So kann der Nutzer sehr schnell identifizieren, welche Dokumente für ihn interessant sein könnten und er genauer betrachten sollte. Er erspart sich damit das Überfliegen aller Dokumente und kann sich auf die für ihn Wichtigsten konzentrieren. In diesem Zusammenhang kann auch das Archivieren von Dokumenten unterstützt werden. Die Dokumente können entsprechend ihrer automatischen Strukturierung direkt in das Ablagesystem des Nutzers übernommen werden, falls dieser das wünscht. Das betrifft sowohl das Einsortieren einzelner Dokumente als auch das Übernehmen vollständiger, neuer (Teil-)hierarchien.

Ein großes Problem bei der Verwendung komplexer Ablagesysteme ist deren Aktualität. Da sich das Interesse und die Ansichten eines Nutzer (und damit auch seine Strukturierungspräferenzen) über die Zeit hinweg verändern, können Teile der Nutzerhierarchie veralten und der aktuellen Sichtweise des Nutzers widersprechen. Die entwickelten Methoden können eingesetzt werden, um den Nutzer auf solche Unstimmigkeiten hinzuweisen und eine Aktualisierung der Ablage vorzuschlagen. Über einen Zeitstempel können alte Daten identifiziert werden. Anhand der aktuelleren Daten kann ein Modell gelernt werden, auf dessen Basis dann die alten Daten strukturiert werden. Weicht die Neustrukturierung von der alten Anordnung ab, kann eine Anpassung der Datenablage vorgeschlagen werden. Der Nutzer muss dann nur noch entscheiden, ob er (teilweise) diesem Vorschlag folgt oder

doch lieber bei der alten Einordnung bleibt.

Ein weiteres Problem stellt der erstmalige Aufbau eines hierarchischen Ablagesystems dar. Zu Beginn liegen zu wenig Daten vor, so dass eine Hierarchie noch nicht nötig ist. Mit der Zeit erhöht sich die Menge an Daten, aber gleichzeitig scheuen sich die Nutzer vor dem mit dem manuellen Aufbau verbundenen Aufwand, so dass sich ein immer unübersichtlicherer Berg an Daten aufbaut, in dem der Nutzer sich immer weniger zu recht findet. Hierfür ist ein interaktives System denkbar, welches dem Nutzer diese Aufgabe vereinfacht. Startpunkt stellt eine unüberwacht (also ohne Strukturierungspräferenzen) gelernte Hierarchie dar, die der Nutzer sukzessive kritisieren und verändern kann. Aus diesem Feedback kann das System Bedingungen an die Zielstruktur ableiten und damit einen neuen, besser an den Nutzer angepassten Strukturierungsvorschlag generieren. Da der Nutzer "nur noch" kritisieren statt selber kreieren muss, vereinfacht sich seine Aufgabe. Außerdem kann das System von wenigen Beispielen auf andere, ähnliche Dokumente schließen und diese automatisch entsprechend behandeln.

Diese Beispiele zeigen, wie ein Nutzer mittels der entwickelten Strukturierungsverfahren bei der Informationssuche und -organisation unterstützt werden kann. Für die spätere Akzeptanz beim Nutzer ist neben der Qualität der Strukturierung aber auch die Nutzerschnittstelle mit einer adäquaten Interaktion und Darstellung von großer Wichtigkeit, die aber in der Dissertation nicht näher betrachtet wurde.

5 Fazit

In der Dissertation wurden verschiedene Verfahren entwickelt, die eine Dokumentensammlung hierarchisch strukturieren können. Dabei wird die Sichtweise des jeweiligen Nutzers berücksichtigt, um einen optimalen Überblick über die Daten zu gewährleisten. Dafür wurden sowohl Verfahren der hierarchischen Klassifikation als auch des teilüberwachten hierarchischen Clusterings untersucht. Die entwickelten Klassifikationsalgorithmen zeichnen sich gegenüber existierenden Verfahren insbesondere darin aus, dass die Nützlichkeit der Klassifikation für einen interagierenden Nutzer bei der Vorhersage im Vordergrund steht. Des Weiteren wurden insbesondere praktische Probleme untersucht, die für eine spätere Anwendbarkeit von Bedeutung sind. Dazu gehören wenige Trainingsdaten und ein unvollständiges Wissen über die Zielhierarchie. Letzteres kann durch Klassifikationsmethoden nur bedingt behandelt werden, da sie ursprünglich Wissen über die gesamte Zielhierarchie voraussetzen. Daher wurde ein neuartiges, teilüberwachtes, hierarchisches Clusteringverfahren entwickelt, das verfügbares Wissen über die Zielhierarchie integriert, diese aber bei Lücken entsprechend erweitern kann. Für diese Lernaufgabe gab es bisher keine existierenden Verfahren. Die Evaluierung der Verfahren mit Benchmarktextdaten zeigt das große Potential der Verfahren, insbesondere unter realen Bedingungen. Dies konnte auch durch eine Suchmaschinenprototyp gezeigt werden, der die Verfahren in der Informationssuche einsetzt. Für die zukünftige Entwicklung erscheint die Kombination der Verfahren mit verschiedenen Techniken wie Stichwort basierte Suche oder interaktive Hierarchievisualisierung besonders viel versprechend. Außerdem lassen sich die Verfahren auf andere Medientypen wie Bilder, Musik oder Videos übertragen. Voraussetzung dafür ist lediglich

eine adäquate Beschreibung durch verschiedene Eigenschaften. Ebenso ist die Anwendung beim Aufbau von Ontologien denkbar. Die vorgestellten Methoden erlauben einen effizienten, teilautomatischen Aufbau einer Ontologie, in dem sie existierendes Domainwissen mit den vorgeschlagenen Mechanismen in automatische Methoden integrieren.

Literatur

[Bad09] Korinna Bade. *Personalized Hierarchical Structuring*. Sierke Verlag, Göttingen, 2009.

[BBM04a] S. Basu, A. Banerjee und R. Mooney. Active Semi-supervision for Pairwise Constrained Clustering. In *Proceedings of the 4th SIAM International Conference on Data Mining*, Seiten 333–344, 2004.

[BBM04b] Mikhail Bilenko, Sugato Basu und Raymond J. Mooney. Integrating Constraints and Metric Learning in Semi-supervised Clustering. In *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*, Seiten 81–88, 2004.

[BHHSW03] Aharon Bar-Hillel, Tomer Hertz, Noam Shental und Daphna Weinshall. Learning Distance Functions using Equivalence Relations. In *Proceedings of the 20th International Conference on Machine Learning (ICML'03)*, Seiten 11–18, 2003.

[CH04] L. Cai und T. Hofmann. Hierarchical Document Categorization with Support Vector Machines. In *Proc. of the 13th ACM Conf. on Inform. and Knowl. Management*, 2004.

[Joa02] T. Joachims. *Learning to Classify Text Using Support Vector Machines - Methods, Theory and Algorithms.* Kluwer Academic Publishers, 2002.

[KL02] H. Kim und S. Lee. An Effective Document Clustering Method Using User-adaptable Distance Metrics. In *Proc. of the 2002 ACM Symp. on Applied Computing*, 2002.

[vNM53] J. von Neumann und O. Morgenstern. *Theory of Games and Economic Behavior*. John Wiley and Sons, 1953.

[WCRS01] Kiri Wagstaff, Claire Cardie, Seth Rogers und Stefan Schroedl. Constrained K-means Clustering with Background Knowledge. In *Proceedings of 18th International Conference on Machine Learning*, Seiten 577–584, 2001.

Korinna Bade studierte Informatik an der Otto-von-Guericke-Universität Magdeburg (OvGU) und der University of Wisconsin Stevens Point (UWSP). 2002 erhielt sie einen Bachelor-Abschluss von der UWSP und 2003 erhielt sie das Diplom von der OvGU als Jahrgangsbeste. Nach dem Studium arbeitete sie als wissenschaftliche Mitarbeiterin an der OvGU. Ihre Forschungsinteressen liegen in den Bereichen Data Mining und Information Retrieval und dabei insbesondere in der Nutzerunterstützung durch Klassifikation und Clustering. Sie promovierte in 2009 mit der hier vorgestellten Dissertation. Seit 2010 arbeitet sie am Umweltbundesamt als wissenschaftliche Mitarbeiterin an der Weiterentwicklung des Stoffinformationssystem GSBL.