February 2005

Semiautomatische Annotation von Textdokumenten mit semantischen Metadaten

Torsten Priebe
Universität Regensburg

Jan Kolter
Universität Regensburg

Christine Kiss
Technische Universität München

In: Ferstl, Otto K, u.a. (Hg) 2005. *Wirtschaftsinformatik 2005: eEconomy, eGovernment, eSociety*; 7. Internationale Tagung Wirtschaftsinformatik 2005. Heidelberg: Physica-Verlag

ISBN: 3-7908-1574-8

Semiautomatische Annotation von Textdokumenten mit semantischen Metadaten

Torsten Priebe, Jan Kolter
Universität Regensburg

Christine Kiss
Technische Universität München

Zusammenfassung: Metadaten sind eine verbreitete Lösung, den Herausforderungen des Wissens- und Dokumentenmanagements zu begegnen. Mit Tim Berners-Lees Vision des Semantic Web sind Metadaten wieder in das Zentrum der Betrachtung gerückt. Dies drückt sich in Standardisierungsbestrebungen (z.B. RDF, OWL, Dublin Core) und einer Vielzahl von neueren Forschungsarbeiten aus. Das Hauptproblem, die Erstellung der Metadaten, ist jedoch nach wie vor ungelöst. Bislang müssen Dokumente manuell annotiert werden, was oft zu mangelnder Benutzerakzeptanz führt. Daher untersucht dieser Beitrag, inwiefern Techniken des Text Mining und der Information Extraction den Prozess der Anreicherung von Textdokumenten mit semantischen Metadaten unterstützen können. Weiterhin wird skizziert, wie als nutzbar identifizierte Verfahren im Rahmen eines semiautomatischen Ansatzes in ein Wissensportalsystem integriert werden können.

Schlüsselworte: Metadaten, Taxonomie, Ontologie, Text Mining, Information Extraction

1 Einleitung

In der heutigen Informationsgesellschaft ist ein effizienter Zugang zu relevanten Informationen zum Schlüsselproblem in allen Lebensbereichen geworden. Dies gilt insbesondere im betrieblichen Umfeld, wo durch Wissensvorsprung massive Wettbewerbsvorteile erzielt werden können. Dabei existiert eine wahre Flut an Daten und Informationen. Jedoch ist eben durch diese Informationsflut das Auffinden von in einer bestimmten Entscheidungssituation tatsächlich relevanten Informationen zum Kunststück geworden. Dies liegt insbesondere daran, dass der Datenbestand eines Unternehmens nur zu einem geringen Teil aus strukturierten (und damit leicht greifbaren) Datenbankdaten besteht. Die meisten Informationen liegen in unstrukturierten Textdokumenten vor (z.B. als Emails, in Formaten wie Microsoft Word oder PDF).

Abhilfe kann hier die Verwendung von Metadaten schaffen. Unter Metadaten versteht man von Mensch und Maschine lesbare Zusatzinformationen, um die man die Dokumente anreichert (man spricht auch von Annotation). Diese können da einen wertvollen Beitrag zur Informationsfindung leisten, wo klassische volltextbasierte Suchverfahren aufgrund der fehlenden Semantik an ihre Grenzen stoßen. Metadaten sind bereits seit vielen Jahren im Dokumenten- und Wissensmanagement im Einsatz. Jedoch fehlte es bislang an standardisierten Techniken und Verfahren, die Lösungen waren meist proprietär und wenig interoperabel. Als Hauptproblem gilt jedoch die mangelnde Benutzerakzeptanz. Wenn ein Benutzer ein Textdokument in ein Wissens- oder Dokumentenmanagementsystem einbringt, muss er die relevanten Metadaten zusätzlich manuell angeben. Die Bereitschaft zu dieser Mehrarbeit ist jedoch nur zu erreichen, wenn ein Nutzen aufgezeigt werden kann. Eine Verbesserung in der Informationsfindung wird sich jedoch erst einstellen, wenn eine gewisse kritische Masse an mit Metadaten angereicherten Dokumenten vorhanden ist. Dies stellt gewissermaßen einen Teufelskreis dar.

Erst mit Aufkommen der Idee des Semantic Web [BeHL01] sind Metadaten wieder in das Zentrum der Betrachtung gerückt. Dies drückt sich zum einen in Standardisierungsbestrebungen (z.B. RDF [W3C04a], Dublin Core [DCMI03]) und zum anderen in einer Vielzahl von neueren Forschungsarbeiten aus. Mittlerweile existieren interoperable Werkzeuge zur Speicherung und Anfrage (z.B. Sesame[1] oder Jena[2]), das Hauptproblem, nämlich die Erstellung der Metadaten, ist jedoch nach wie vor ungelöst.

Ziel dieses Beitrages ist es daher, geeignete Techniken zu finden, die den Prozess der Anreicherung von Textdokumenten mit semantischen Metadaten unterstützen können. Da Metadaten nur bei ausreichend hoher Qualität einen wirklichen Nutzen bringen und, wie der Beitrag zeigen wird, vollautomatische Verfahren diese Qualität nicht gewährleisten können, streben wir eine semiautomatische Vorgehensweise an, d.h. der Benutzer bekommt automatisch generierte Metadaten vorgeschlagen, muss diese aber noch bestätigen, bzw. kann sie ggf. korrigieren.

Der Rest dieses Beitrages gliedert sich wie folgt: Abschnitt 2 motiviert die Verwendung von Metadaten im Wissensmanagement und geht kurz auf Standards sowie die Rolle von Taxonomien und Ontologien ein. Abschnitt 3 gibt einen Überblick über Techniken des Text Mining und der Information Extraction. Die beschriebenen Techniken werden dann in Abschnitt 4 dazu herangezogen, Verfahren zur automatischen Generierung diverser Metadatenelemente vorzuschlagen. Darauf aufbauend beschreibt Abschnitt 5 die geplante Umsetzung im Rahmen des Wissensportalsystems INWISS. Nachdem in Abschnitt 6 eine Abgrenzung zum Stand der Technik und zu verwandten Forschungsarbeiten erfolgt ist, schließt Abschnitt 7 den Beitrag mit einer Zusammenfassung und einem Ausblick.

1 http://www.openrdf.org
2 http://jena.sourceforge.net

2 Wissensmanagement und die Rolle von Metadaten

Metadaten sind eine verbreitete Lösung, den Herausforderungen des Wissens- und Dokumentenmanagements zu begegnen. Die Grundidee ist es, zu den Ressourcen zusätzliche, beschreibende Informationen abzulegen, die sowohl für Menschen als auch für Maschinen lesbar und verständlich sind. Diese Metadaten stellen also eine Kennzeichnung oder Klassifikation der Ressource dar. Im Ergebnis soll die Informationsflut eingedämmt bzw. handhabbarer gemacht werden. Insbesondere das Navigieren und Suchen nach Ressourcen soll effizienter werden.

Mittlerweile existieren einige Standards zur Speicherung und Verwaltung von Metadaten. Das Resource Description Framework (RDF) [W3C04a] definiert eine standardisierte, XML-basierte Form, Metadaten zu repräsentieren. RDF erweitert also das nur auf syntaktischer Ebene definierte XML um formale Semantik. Dazu verwendet es ein einfaches Tripel-basiertes Modell. Ressourcen werden über eine URI eindeutig identifiziert und besitzen eine Menge von Properties mit spezifischen Werten. Die Werte können dabei Literale oder andere Ressourcen sein (wiederum identifiziert durch ihre URI). RDF Schema [W3C04b] erweitert RDF um Modellierungskonstrukte, ähnlich wie XML Schema bei XML.

2.1 Dublin Core

Während RDF also selbst nur das Grundmodell definiert, d.h. die Tatsache, dass eine Ressource Properties mit Werten haben kann, ist es mit RDF Schema möglich, die zu verwendenden Metadatenelemente genauer festzulegen. Allerdings muss dies je nach Anwendungsfeld individuell geschehen. Um Interoperabilität, insbesondere im Internet, zu ermöglichen, ist es jedoch wünschenswert, gewisse universelle Metadatenelemente als eine Art Grundwortschatz festzulegen. Hier setzt die Dublin-Core-Initiative[3] an. Die Aufgabe und Motivation von Dublin Core ist es, mit einer festgesetzten Menge von Elementen das Auszeichnen von Textdokumenten und anderen Ressourcen zu standardisieren.

Die Ursprünge von Dublin Core liegen im WWW und im Bereich der Digital Libraries. Daneben existieren mittlerweile viele domänenspezifische Metadatenstandards (z.B. IMS Global Learning Consortium[4]). Für die Zwecke dieses Beitrages wurde der anwendungsunabhängige Dublin Core als Betrachtungsgrundlage ausgewählt, die Konzepte lassen sich jedoch auch auf andere Modelle übertragen. Der in [DCMI03] definierte Standard umfasst (im Grundmodell) fünfzehn Elemente, deren Semantik von einer internationalen und interdisziplinären Gruppe von Experten aufgestellt wurde. Im Wissens- und Dokumentenmanagement sind dabei insbesondere *Title*, *Creator*, *Subject*, *Description*, *Type*, *Format*, *Language*

[3] http://www.dublincore.org
[4] http://www.imsglobal.org

und *Coverage* von Interesse. Diese werden auch im weiteren Verlauf dieses Beitrages verwendet.

Prinzipiell sollte als Wertmenge, soweit möglich, kein Freitext, sondern ein kontrolliertes Vokabular verwendet werden. Das Element *Subject* wird dabei in der Regel mit Hilfe von vordefinierten Schlüsselworten oder Klassifikationen angegeben. Am gebräuchlichsten ist die Verwendung einer Taxonomie (siehe nächster Unterabschnitt). *Coverage* bezeichnet den Umfang bzw. die Reichweite des Inhaltes der Ressource, nach Dublin Core z.B. räumliche Orte und zeitliche Perioden. Im Unternehmenskontext lassen sich jedoch auch andere mögliche „Objekte" finden, beispielsweise Produkte oder Abteilungen. Als kontrolliertes Vokabular bietet sich hier die Verwendung einer Ontologie an.

Die gebräuchlichste Form der Darstellung von Dublin-Core-Metadaten ist RDF. Die Dublin-Core-Elemente werden dabei über den Namespace „http://purl.org/dc/elements/1.1/" identifiziert. So ist das Element *Subject* beispielsweise als „http://purl.org/dc/elements/1.1/subject" (kurz „dc:subject") definiert.

2.2 Taxonomien und Ontologien

Ein weit verbreiteter Ansatz zur Organisation von Dokumenten und anderen Ressourcen ist die Verwendung einer Taxonomie. Eine Taxonomie besteht aus einer hierarchisch geordneten Menge von Kategorien. Ein Dokument ist dabei (beispielsweise über das Dublin-Core-Element *Subject*) einer oder mehreren Kategorien zugeordnet. Man kann sich eine Kategorie also als eine Art virtuellen Ordner vorstellen, in dem alle Dokumente, die das entsprechende Thema behandeln, aufgeführt sind. Diese Vorgehensweise kennt man auch aus Web-Verzeichnissen, wie Yahoo[5] oder DMOZ[6].

Ein Auszug aus einer beispielhaften Taxonomie ist in Abbildung 1 dargestellt. Als Szenario gehen wir von einem Unternehmen aus, welches diverse Produkte aus dem Consumer-Bereich über Call Center vertreibt. Als erste Kategorieebene wird in „Sales", „Distribution" und „Marketing" verzweigt. Während der Bereich „Sales" geographisch untergliedert wird, orientiert sich die Verfeinerung innerhalb von „Marketing" an den Produkten des Unternehmens. Für die Beschreibung der Taxonomie bietet sich ebenfalls RDF an. Dieser Ansatz wird auch von DMOZ verfolgt[7]. Die einzelnen Kategorien werden dabei als per URI identifizierte Ressourcen aufgefasst, der hierarchische Aufbau der URIs spiegelt die Hierarchie der Taxonomie wieder. Die Kategorie „Audio" in Abbildung 1 könnte so beispielsweise als „http://www.inwiss.org/topics/Marketing/Electronics/Audio" kodiert werden.

---

5 http://www.yahoo.com

6 http://www.dmoz.org

7 DMOZ RDF Dumps sind unter http://rdf.dmoz.org verfügbar

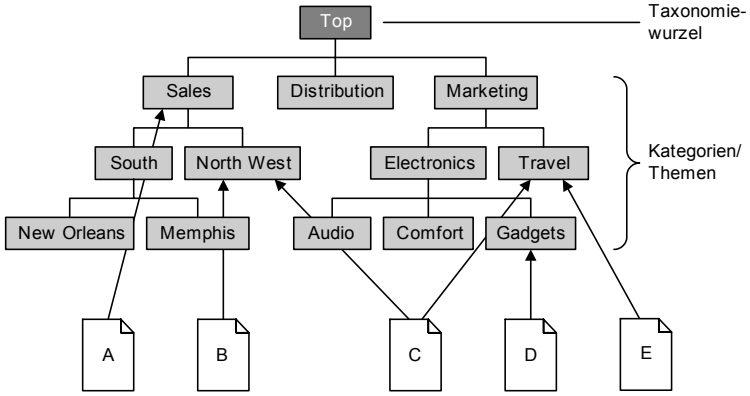

Abbildung 1: Einordnung von Dokumenten in eine thematische Taxonomie

Taxonomien werden in Wissensmanagementsystemen meist zur Navigation als eine Art Menüstruktur eingesetzt. Daher ist ihre Einfachheit und Verständlichkeit von besonderer Bedeutung. Um den inhaltlichen Umfang einer Ressource (z.B. per Dublin-Core-Element *Coverage*) vollständig zu erfassen, sind komplexere Strukturen notwendig. Wie bereits erwähnt, bietet sich hier die Verwendung einer Ontologie an.

Ontologien dienen dazu, einen bestimmten Ausschnitt der Realität formal zu beschreiben. Im Wesentlichen auf DAML+OIL[8] aufbauend hat sich als Formalismus hier die Web Ontology Language (OWL) [W3C04c] etabliert. OWL baut auf RDF Schema (s.o.) auf und erweitert es um zusätzliche Konstrukte. Es werden Klassen und zugehörige Instanzen definiert. Die Klassen folgen einer Vererbungshierarchie, beispielsweise „Person“ mit „Customer“ und „Employee“ als Subklassen. Die Instanzen in der Ontologie liefen in ihrer Gesamtheit die Objekte, die in einer betrachteten Domäne (hier im Unternehmenskontext) eine Rolle spielen.

Die Begriffe Taxonomie und Ontologie lassen sich nur schwer klar von einander abgrenzen. Teilweise wird eine Taxonomie als „einfache“ Ontologie gesehen, oder als Teilmenge einer Ontologie [McGu02]. In diesem Beitrag verwenden wir die Terminologie wie dargestellt. Eine Taxonomie stellt eine hierarchische Struktur zur Benutzernavigation dar. Eine Ontologie hingegen beschreibt einen Realitätsausschnitt formal; hier steht die Vollständigkeit und maschinelle Verarbeitbarkeit im Vordergrund.

Im Vergleich zu einer Taxonomie ist eine Ontologie also komplexer strukturiert und beinhaltet üblicherweise eine größere Zahl an Instanzen. Während eine Ressource tendenziell einer oder wenigen Taxonomiekategorien zugeordnet sein wird, kann ihr Umfang durchaus eine größere Menge an Ontologieinstanzen umfassen.

8 http://www.daml.org

Dies hat Auswirkungen auf die anwendbaren Techniken zur Metadatengenerierung. Daher betrachten wir die Einordnung von Dokumenten in eine Taxonomie und ihre Verlinkung mit Elementen einer Ontologie in Abschnitt 4 getrennt.

Der folgende Code zeigt die vollständigen RDF-Metadaten einer Beispielressource. Es kommen die in Unterabschnitt 2.1 aufgeführten Dublin-Core-Elemente zum Einsatz. Neben literalen Werten finden dabei Kategorien aus einer Taxonomie (als *Subject*) und Instanzen aus einer Ontologie (als *Coverage*) Verwendung:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY rdf
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <!ENTITY dc "http://purl.org/dc/elements/1.1/">
  <!ENTITY onto "http://www.inwiss.org/ontology#">
  <!ENTITY top "http://www.inwiss.org/topics/">
]>
<rdf:RDF xmlns:rdf="&rdf;" xmlns:dc="&dc;"
  xmlns:onto="&onto;">
<rdf:Description rdf:about="SomeResource">
  <dc:title>The Freeplay(TM) Solar Radio</dc:title>
  <dc:creator rdf:resource="ldap://cn=Tina
    Techwriter,ou=Marketing,o=MyCompany"/>
  <dc:date>1999-01-14</dc:date>
  <dc:type rdf:resource="&onto;ProductBrochure"/>
  <dc:format>application/pdf</dc:format>
  <dc:language>en-US</dc:language>
  <dc:description>The Freeplay Solar Radio...
    </dc:description>
  <dc:subject rdf:resource=
    "&top;Marketing/Electronics/Audio"/>
  <dc:coverage rdf:resource=
    "&onto;FreeplaySolarRadio"/>
</rdf:Description>
</rdf:RDF>
```

3 Text Mining und Information Extraction

Text Mining beschäftigt sich mit der Erkennung von Mustern und der Extraktion von Wissen aus unstrukturierten Textdokumenten. Da sich die ältere Disziplin Data Mining mit der Extraktion von Mustern aus strukturierten Daten beschäftigt, kann man Text Mining als ein spezielles Anwendungsgebiet von Data Mining bezeichnen [Tan99]. Text Mining beinhaltet darüber hinaus Elemente aus einer Rei-

he weiterer Wissensgebiete wie z.B. Information Retrieval [BaRi99], Information Extraction und Machine Learning. Beim Text Mining wird versucht, aus einer Menge von Dokumenten Zusammenhänge festzustellen und Regeln abzuleiten.

Um klassische Data-Mining-Verfahren anwenden zu können, müssen die Dokumente in eine strukturierte Form konvertiert werden. Tan [Tan99] bezeichnet dies als Text-Refining-Phase, welche eine „Document-based Intermediate Form" als Ergebnis hat. Die einzelnen Schritte der Textvorverarbeitung sind dabei:

- Bei der Tokenization werden logische Einheiten des Textes erkannt, d.h. er wird üblicherweise in Wörter separiert.
- Im Zuge der Stoppworteleminierung werden häufig vorkommende Wörter ohne eigene Bedeutung (z.B. „und", „ist") eliminiert.
- Reduktion von Flexionsformen auf die Stammform eines Wortes.
- Identifikation von Mehrwortgruppen und das Auflösen von Mehrdeutigkeiten (Homonyme und Synonyme).

Nach abgeschlossenen Vorverarbeitungsschritten wird das Dokument als Vektor $d_j = (w_{1,j}, w_{2,j}, ...w_{t,j})$ repräsentiert. Diese Vorgehensweise ist auch im Information Retrieval üblich [BaRi99]. Es wird ein t-dimensionaler Raum definiert, wobei *t* der Anzahl der Terme des verwendeten Vokabulars entspricht. Man spricht daher auch vom Vektorraummodell.

Die einzelnen Dimensionen entsprechen dem Auftreten bzw. der Gewichtung der einzelnen Terme des Vokabulars. Alle Vektoren der vorliegenden Dokumentensammlung bilden eine Term Frequency Matrix. Die Weighted Term Frequency gewichtet die einzelnen Terme nach deren Vorkommenshäufigkeit in einem Dokument und nach der Trennschärfe des Begriffs bezogen auf die Dokumentensammlung. Die Gewichtung berechnet sich wie folgt:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Dabei steht $tf_{i,j}$ für die Vorkommenshäufigkeit des Termes *i* im Dokument *j*, *N* für die Anzahl der Dokumente in der Dokumentensammlung und df_i für die Anzahl der Dokumente, die den Term *i* enthalten. Das Ergebnis $w_{i,j}$ ist der Wert des Terms *i* im Vektor des Dokuments *j*.

Befindet sich das Dokument in der strukturierten Form des Vektorraummodells, können Data-Mining-Methoden angewendet werden.

3.1 Automatische Textkategorisierung

Unter Textkategorisierung versteht man die Zuordnung eines booleschen Wertes zu jedem Paar $< d_j, c_i > \in D \times C$ wobei *D* eine Menge von Dokumenten und *C* ei-

ne Menge von Klassen darstellt [Seba02]. Man unterscheidet zwischen single-label und multi-label Kategorisierung, je nachdem, ob einem Dokument genau eine Kategorie oder mehreren Kategorien zugeordnet werden soll. Zur automatischen Kategorisierung bieten sich aus dem Data Mining bekannte Klassifikationsverfahren an.

Die Klassifikation ist ein überwachtes Lernverfahren, welches einer gegebenen Instanz eine aus einer vorher festgelegten Menge diskreter Klassen vorhersagt [WiFr00]. Für die Zuordnung einer neuen Instanz zu einer Klasse ist ein Klassifikator erforderlich, der mithilfe von vorklassifizierten Trainingsdaten erlernt wird. Bei der Auswahl geeigneter Klassifikationsverfahren ist zu beachten, dass Dokumentenvektoren die Eigenschaft haben, dass die Anzahl der Attribute sehr groß ist, hingegen die Anzahl der Instanzen eher gering. Weiteres Merkmal ist, dass alle Attribute metrische Werte haben, die Klassen jedoch nominale.

3.1.1 Entscheidungsbäume

Entscheidungsbäume werden durch rekursive Aufteilung der Datenbestände konstruiert. Hierzu gibt es verschiedene Algorithmen, einer der bekanntesten ist ID3, welcher auch bei C4.5 [Quin86] Anwendung findet. Bei diesem Verfahren wird der Informationsgewinn, der durch einzelne Attribute erreicht wird, als Entscheidungskriterium für die Verzweigung im Entscheidungsbaum herangezogen. Die Blätter des Baumes bilden die Klassen. Es findet eine rein nominelle Auswertung statt, weshalb metrische Attribute zuvor in nominale umgerechnet werden müssen. Da der Dokumentenvektor ausschließlich metrische Attribute enthält, ist dies ein großer Nachteil. Auch die große Anzahl an Attributen spricht gegen den Entscheidungsbaum.

3.1.2 Naive Bayes

Das Naive-Bayes-Verfahren eignet sich prinzipiell besser für die Klassifikation von Dokumenten, da es auch numerische Werte berücksichtigen kann, wobei angenommen wird, dass diese normalverteilt sind. Hier wird die wahrscheinlichste Klasse nach dem Gesetz der bedingten Wahrscheinlichkeiten ermittelt. Allerdings ist zu beachten, dass bei der Klassifikation von neuen Dokumenten eine Dichtefunktion für jede Klasse für jedes Attribut berechnet werden muss.

3.1.3 Instanzbasierte Verfahren

Bei instanzbasierten Verfahren, wie z.B. K-Nearest-Neighbor, erfolgt keine Modellgenerierung; stattdessen werden die Trainingsdaten gespeichert und für die Klassifizierung einer neuen Instanz herangezogen. Der K-Nearest-Neighbor-Algorithmus geht davon aus, dass jede Instanz einem Punkt (bzw. Vektor) in einem n-dimensionalen Raum entspricht. Der nächste Nachbar einer Instanz wird durch die Distanz der einzelnen Attribute zu anderen Instanzen bestimmt. *K* defi-

niert die Anzahl der Nachbarn, nach denen gesucht wird. Für die zu klassifizierende Instanz wird dann die Mehrheitsklasse der *K* Nachbarn gewählt.

Als Distanz- (bzw. Ähnlichkeits-) Maß bietet sich im Text Mining die aus dem Information Retrieval bekannte Cosine Similarity an [BaRi99]. Wenn $d_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$ dem Dokumentenvektor entspricht, dann ist die Ähnlichkeit zwischen Dokument d_j und Dokument d_k folgendermaßen definiert:

$$sim(d_j,d_k) = \frac{d_j \cdot d_k}{|d_j||d_k|} = \frac{\sum_{i=1}^{t} w_{i,j} \cdot w_{i,k}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,k}^2}}$$

Dieses Verfahren eignet sich besonders für das Text Mining, da hier ausschließlich metrische Attribute vorliegen und die Berechnung schnell und einfach durchzuführen ist. Das fehlende Modell stellt bei der Dokumentenklassifikation keinen Nachteil dar.

3.1.4 Support Vector Machines

Ein weiteres Verfahren zur Klassifikation von Textdokumenten bieten die Support Vector Machines. Diese basieren auf dem Structural-Risk-Minimation-Prinzip [Vapn95] der Computational Learning Theory. In ihrer Grundform werden Support Vector Machines für binäre Klassifikation eingesetzt. Es existiert jedoch eine Reihe von Abwandlungen, die die Definition mehrerer Klassen erlauben. Es werden lineare Grenzfunktionen erlernt; das Ziel ist, in einem n-dimensionalen Raum eine Hyperebene zu finden, welche den Raum anhand der Klassenzuordnung in zwei Hälften teilt.

3.1.5 Bewertung

Über die Eignung bzw. Performance von Klassifikationsverfahren kann man keine allgemeinen Aussagen treffen, diese ist grundsätzlich abhängig vom Datenbestand [MiST94]. Betrachtet man die Charakteristika des Datensets, kann die Anzahl der Klassifikationsverfahren eingeschränkt werden, um aber eine optimale Performance zu erzielen, sollten immer mehrere Verfahren evaluiert werden.

Bei der Evaluierung von Klassifikatoren wird deren Effektivität betrachtet, sprich die Fähigkeit, die richtige Klassifizierungsentscheidung zu treffen. Üblicherweise werden Performance-Kennzahlen wie Genauigkeit (Anteil der korrekt klassifizierten Instanzen) und die Fehlerrate (Anteil der falsch klassifizierten Instanzen) für die Evaluierung herangezogen. Bei der Textkategorisierung werden hingegen oft auch die aus dem Information Retrieval übernommenen Kennzahlen *Precision* und *Recall* als Entscheidungskriterium berechnet. Die *Precision* gibt den Anteil

der relevanten Dokumente unter den gefundenen Dokumenten an, der *Recall* gibt den Anteil der relevanten Dokumente an, die gefunden wurden [BaRi99].

Es gab bereits zahlreiche Untersuchungen über die Eignung von Klassifikationsverfahren für Text Mining. Das meist verwendete Datenset für die Evaluierung ist der Reuters 21578 Textkorpus, bestehend aus Berichten aus verschiedenen Bereichen der Wirtschaft (siehe auch in Abschnitt 7). Sebastiani [Seba02] hat einen Vergleich von fünf Benchmarks aufgestellt. Im Ergebnis schneiden Naive Bayes und Entscheidungsbäume schlecht ab. Die Support Vector Machines und der K-Nearest-Neighbor-Algorithmus sind unter den Gewinnern. Ebenfalls sehr gute Ergebnisse erzielt der Boosting-Algorithmus, welcher eine Kombination von verschiedenen Verfahren darstellt.

3.2 Clustering von Dokumenten

Ziel des Clustering ist die Aufteilung von Eingabedaten in dynamisch erzeugte Gruppen (Cluster), so dass die Datensätze innerhalb einer Gruppe sehr ähnlich und die Attribute unterschiedlicher Gruppen möglichst verschieden sind [WiFr00]. Das Clustering ist ein unüberwachtes Lernverfahren, bei dem keine Klassen vorgeben sind. Clustering-Algorithmen können im Text Mining ihre Anwendung finden, indem ähnliche Dokumente in Gruppen aufgeteilt werden.

Die große Anzahl an Attributen führt hier zu keiner Einschränkung, da Clustering-Algorithmen (beispielsweise K-Means) wie instanzbasierte Klassifikationsverfahren auf Ähnlichkeits- bzw. Distanzfunktionen beruhen. Es bietet sich also auch hier die Verwendung von aus dem Information Retrieval bekannten Maßen wie der Cosine Similarity an.

3.3 Termbasierte Abhängigkeitsanalyse

Mit der Abhängigkeitsanalyse sollen Abhängigkeiten zwischen Attributen identifiziert werden. Es können einzelne Attribute oder auch eine Kombination von Attributen prognostiziert werden [WiFr00]. Klassisches Beispiel für eine Abhängigkeitsanalyse im Data Mining ist die Warenkorbanalyse (d.h. es werden Produkte gefunden, die häufig gemeinsam gekauft werden). Im Text Mining bedeutet dies, dass Wörter bzw. Terme gefunden werden, die häufig gemeinsam in Dokumenten erscheinen. Mithilfe der Abhängigkeitsanalyse lassen sich daher beispielsweise im Rahmen der Vorverarbeitung Synonyme und Hypernyme (Oberbegriffe) erkennen.

Interessanter als Zusammenhänge zwischen beliebigen Wörtern sind jedoch oft Zusammenhänge zwischen bestimmten Entitäten, wie Personen- oder Firmennamen. Diese müssen dann zunächst mit Hilfe von Information-Extraction-

Techniken (s.u.) extrahiert werden. Dadurch reduziert sich dann auch die Anzahl der Attribute.

3.4 Information Extraction

Information Extraction ist eine relativ junge Disziplin und wird zum Teil als Teildisziplin des Text Mining gesehen. Jedoch steht hier die Extraktion und Strukturierung von bestimmten Textsegmenten im Vordergrund, während Text Mining die Aufgabe hat, Muster zu erkennen bzw. neues Wissen zu generieren. Während ein Text im klassischen Text Mining auf eine Menge von Worten (bzw. seinen Dokumentenvektor) reduziert werden kann, müssen bei Information Extraction im Rahmen der Vorverarbeitung sprachwissenschaftliche Verfahren herangezogen werden, beispielsweise das Part of Speech Tagging (Erkennen von Worttypen, wie Substantiv, Verb, etc.) und die syntaktische Analyse (Erkennen von Satzstrukturen: Subjekt, Prädikat, Objekt, etc.).

Man unterscheidet Entity, Attribute, Fact und Event Extraction. Eine Entität ist ein Objekt wie z.B. eine Person oder eine Organisation. Da solche Objekte meist einen Namen tragen, den es zu extrahieren gilt, wird auch von Named Entity Extraction gesprochen. Heutige Extraktionssysteme leisten die Extraktion von (benannten) Entitäten bereits mit einer akzeptablen Zuverlässigkeit, die Verfahren sind jedoch sehr aufwendig. Scheffler et al. [ScDW01] beschreiben beispielsweise einen auf Hidden Markov Models basierenden Ansatz. Bei den anderen komplexeren Elementen (insbesondere Fakten und Ereignissen) ist jedoch noch einiges an Forschungsarbeit zu leisten.

3.5 Automatische Textzusammenfassung

Ein weiteres mit Text Mining und Information Extraction verwandtes Gebiet ist die automatische Zusammenfassung von Texten. Dabei soll eine Art Abstract erzeugt werden, der alle wichtigen Informationen des Originaldokuments enthält, jedoch wesentlich kürzer ist. Man unterscheidet zwischen statistischen und sprachwissenschaftlichen Verfahren.

Statistische Verfahren orientieren sich an einer strukturierten Repräsentation der Dokumente, beispielsweise dem Vektorraummodell. Ein Ansatz ist hier die schrittweise Suche nach ausdrucksstarken Wörtern anhand der Weighted Term Frequency. Zunächst wird dazu das Dokument in mehrere Abschnitte (z.B. Absätze oder Sätze) aufgeteilt. Anschließend wird für alle Wörter eines Textes die Weighted Term Frequency ermittelt und abschnittsweise aufaddiert. Für die Textzusammenfassung werden diejenigen Abschnitte mit der höchsten Gesamtwertung extrahiert.

In sprachwissenschaftlichen Verfahren (z.B. [BaEl97]) wird nicht nur das Vorkommen einzelner Terme im Dokument untersucht, hier werden auch Wortbedeutungen und Sinnzusammenhänge betrachtet. Bei statistischen Verfahren besteht die Gefahr, dass semantisch zusammengehörige Sätze auseinander gerissen werden und die Zusammenfassung somit unverständlich wird. Dafür ist die Verarbeitungsgeschwindigkeit schneller und die Umsetzung einfacher. Sprachwissenschaftliche Verfahren sind sehr aufwendig, da sie wie Information Extraction komplexere Vorverarbeitungsschritte voraussetzen.

4 Ansätze zur automatischen Metadatengenerierung

Im Folgenden soll nun auf eine automatische Generierung der einzelnen Dublin-Core-Metadatenelemente aus Abschnitt 2.1 eingegangen werden. Zunächst erfolgt ein Blick auf Elemente, deren Generierung keine Text-Mining- oder Information-Extraction-Techniken erfordert:

Das Element *Title* bestimmt die Überschrift bzw. den Titel eines Dokumentes. Dieser könnte der ersten Zeile oder dem Dateinamen entnommen werden, oder es können Layoutmerkmale, z.B. HTML Tags, herangezogen werden. Auch die Elemente *Creator* und *Date* können automatisch generiert werden, indem der Benutzername bzw. das Erstellungsdatum vom Betriebssystem abgefragt werden. Möglicherweise sind diese Informationen auch bereits in strukturierter Form im Dokument (beispielsweise bei Microsoft Office) gespeichert. Das Dublin-Core-Element *Format*, welches den Medientyp (MIME Type) des Dokumentes beschreibt, kann üblicherweise über die Dateierweiterung oder eine Kennung im Dateikopf erkannt werden.

Die Bestimmung der Sprache bzw. des Elementes *Language* ist für die Textvorverarbeitung ohnehin notwendig. Dies kann durch einen Vergleich der Wörter des Dokumentes mit den Vokabularen der verschiedenen Sprachen erfolgen. Grefenstette und Nioche [GrNi00] beschreiben ein Verfahren, bei welchem das Dokument nach häufig auftretenden Stoppwörtern der einzelnen Sprachen untersucht wird.

Das Dublin-Core-Element *Description* soll den Inhalt einer Ressource, in Form eines Inhaltsverzeichnisses oder eines Abstracts, beschreiben. Da das zu annotierende Dokument bereits in der Repräsentationsform eines Dokumentenvektors vorliegt, bieten sich hierfür statistische Verfahren der automatischen Textzusammenfassung an (siehe Abschnitt 3.5).

Die Werte von *Subject* und *Coverage* greifen auf ein kontrolliertes Vokabular zurück, welches neben weiteren Kodierungsschemata eine Taxonomie oder eine Ontologie sein kann. In den nächsten Abschnitten wird hierzu daher auf die automati-

sche Einordnung in eine Taxonomie und die Verlinkung mit Elementen einer Ontologie eingegangen.

4.1 Einordnung von Dokumenten in eine Taxonomie

Wie bereits in Abschnitt 2.2 beschrieben, besteht eine Taxonomie aus einer hierarchisch geordneten Menge von Kategorien. Eine vorhandene Taxonomie kann als Kodierungsschema für das Dublin-Core-Element *Subject* dienen, welches das Thema eines Dokumentes bestimmt. Eine Taxonomie muss dabei nicht vollständig manuell erstellt werden. Der Prozess der Aufstellung einer Taxonomie kann durch Clustering-Methoden des Text Minings unterstützt werden. Hier bieten sich insbesondere hierarchische Clustering-Verfahren an.

Bei der automatischen Einordnung eines Textdokumentes in eine Taxonomie handelt es sich um eine Textkategorisierung. Jedes Element der Taxonomie stellt hierbei eine Klasse dar. Die Taxonomie eines größeren Unternehmens wird dabei aus hunderten bis tausenden Kategorien bestehen. Ein Dokument wird jedoch, wie bereits ausgeführt, tendenziell nur einer oder wenigen Kategorien zugeordnet sein. Es erscheint also sinnvoll, eine single-label Kategorisierung durchzuführen. Die konkrete Problemstellung hat folgende Eigenschaften:

- hohe Anzahl an Klassen
- hohe Anzahl an Attributen
- die Klassen sind nominal
- die Attribute sind metrisch

Nach den Ergebnissen aus 3.1 bieten sich instanzbasierte Verfahren und mehrere Klassen unterstützende Erweiterungen der Support Vector Machines an. Ein Problem bleibt jedoch die hohe Anzahl der Klassen.

Um die Anzahl der Klassen zu reduzieren und so die Performance der Klassifikationsverfahren zu verbessern, könnte man sich die hierarchische Struktur der Taxonomie zunutze machen und bei der Klassifikation hierarchisch vorgehen. Anstatt dass jedes Element der Taxonomie eine Klasse bildet, sollen nur noch die Elemente einer Hierarchieebene als Klassen betrachtet werden. Bei der Klassifikation wird iterativ vorgegangen; sobald die geeignete Klasse (Kategorie) in einer Hierarchieebene gefunden wurde, werden die Kategorien innerhalb dieser Klasse betrachtet und die Klassifikation wird erneut durchgeführt. Die Vertiefung entlang der Hierarchiestruktur erfolgt so lange, bis ein weiterer Detaillierungsgrad keine Verbesserung der Performance darstellt oder die Konfidenz der Klassifikation unter einen bestimmten Wert sinkt. Abbildung 2 zeigt ein Beispiel für eine solche hierarchische Klassifikation. Die grau schattierten Elemente stellen die Klassen dar.

Diese Vorgehensweise sollte die Genauigkeit der Klassifikation wesentlich erhöhen, da bei jeder Iteration nur die relevanten Subklassen betrachtet werden. Allerdings besteht die Gefahr, dass durch einen Fehler bei der Klassifizierung am Anfang dieser Fehler mitgezogen wird und die richtige Klasse nicht mehr identifiziert werden kann. Die empirische Evaluation einer hierarchischen Klassifikation zur Einordnung in eine Taxonomie steht noch aus.

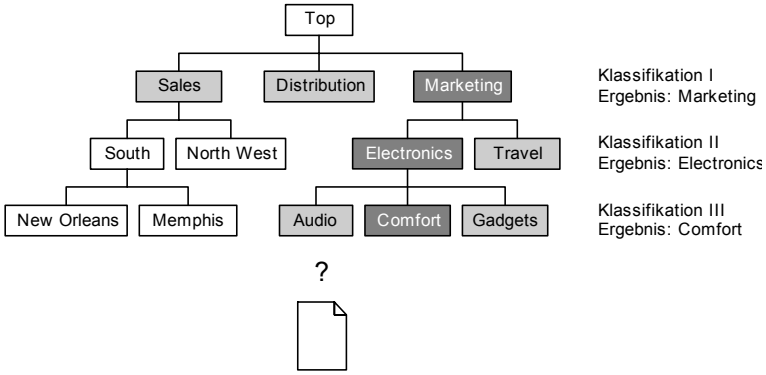

Abbildung 2: Hierarchische Klassifikation zur Einordung in eine Taxonomie

4.2 Verlinkung mit Elementen einer Ontologie

In diesem Abschnitt sollen Wege zur automatischen Erstellung des Dublin-Core-Elements *Coverage* untersucht werden. Das Element beschreibt laut Definition den Umfang des Inhalts einer Ressource.

Der Umfang wird durch eine Menge von Objekten bzw. Entitäten bestimmt, die im betrachteten Dokument vorkommen. Existiert kein Verzeichnis von möglichen Objekten (beispielsweise in Form einer Ontologie) bietet sich daher die Extraktion von Entitäten aus dem Gebiet der Information Extraction an (siehe Abschnitt 3.4). Hierzu kann ein Name Recognizer erstellt werden, der in einem zu annotierenden Dokument die entsprechenden Begriffe erkennt und extrahiert. Die Verfahren sind jedoch sehr aufwendig und fehleranfällig.

Es ist daher anzuraten, als kontrolliertes Vokabular für *Coverage* eine Unternehmensontologie zu verwenden (siehe Abschnitt 2.2). Die Erstellung einer Ontologie kann auch semiautomatisch erfolgen. Maedche [Maed02] beschreibt Ansätze, die semantische Informationen zur Erstellung einer Ontologie extrahieren. Dazu werden ebenfalls Text-Mining-Techniken verwendet. So kann beispielsweise eine Abhängigkeitsanalyse semantische Zusammenhänge zwischen Termen aufdecken [PaBe03]. Weiterhin können die genannten Verfahren der Information Extraction verwendet werden, um eine Ontologie zu erstellen oder zu erweitern. Im Unter-

nehmenskontext ist es jedoch oft praktikabler, eine Ontologie aus der Struktur und dem Inhalt existierender Datenbanken oder einem Data Warehouse zu erstellen, in dem die für das Unternehmen relevanten Objekte (Produkte, Abteilungen, etc.) in der Regel erfasst sind.

Auf diese Weise entsteht wie bei der Generierung von *Subject* ein kontrolliertes Vokabular, das jedoch deutlich mehr Begriffe enthält. In einer Unternehmensontologie können unter Umständen mehrere tausende Instanzen enthalten sein. Ein weiteres Merkmal von *Coverage* ist, dass einem Dokument üblicherweise deutlich mehr als ein Element aus der Ontologie zugewiesen werden kann. Es gilt zu untersuchen, ob unter diesen Bedingungen ebenfalls Verfahren der automatischen Textkategorisierung geeignet sind. Die Problemstellung hat folgende Eigenschaften:

- sehr hohe Anzahl an Klassen
- hohe Anzahl an Attributen
- die Attribute sind metrisch
- Ergebnis können beliebig viele Klassen sein

Es handelt sich also um eine multi-label Kategorisierung mit sehr vielen Klassen. Will man hierfür Klassifikationsverfahren verwenden, muss für jede einzelne Klasse ein eigener binärer Klassifikator gebildet werden. Dabei entscheidet jeder Klassifikator, ob ein Dokument ein bestimmtes Element besitzt oder nicht; es werden also jeweils nur die Klassen *true* und *false* unterschieden. Dem Inhalt von *Coverage* werden anschließend die positiv klassifizierten Elemente zugewiesen.

Für eine binäre Klassifikation sind von den in Abschnitt 3.1 genannten Verfahren die Support Vector Machines besonders geeignet. Zusätzlich können die binär logistische Regression und die Diskriminanzanalyse herangezogen werden. Diese sind statistische Verfahren, welche ähnlich wie die Support Vector Machines mittels einer Hyperebene versuchen, die zwei Klassen best möglich zu teilen. Fishers lineare Diskriminanzanalyse optimiert eine quadratische Kostenfunktion während die logistische Regression den Likelihood maximiert. Da die Diskrimanzanalyse mit nominalen Attributen arbeitet, die logistische Regression hingegen mit metrischen, wäre die binär logistische Regression zu wählen. Diese hat auch bei vielen Klassifikationsvergleichen sehr gut abgeschnitten [KiBi04, MiST94].

5 Semiautomatische Annotation in INWISS

Wissensportale leisten einen wichtigen Beitrag zum Wissensmanagement eines Unternehmens. Unter Verwendung von Web-Technologien bieten sie einen einheitlichen, zentralen Zugriff auf verschiedenste Arten von Informationen. Sie stel-

len zum einen Content- und Dokumenten-Management-Funktionalität, sowie Suchmechanismen zur Verfügung, integrieren zusätzlich aber auch externe Anwendungen, wie ERP- und CRM-Systeme, oder ein OLAP-System zum Zugriff auf Data-Warehouse-Daten. Dies geschieht durch Zusammenfassung mehrerer Portalkomponenten (sog. Portlets) auf einer einzelnen Portal-Webseite.

Heutige Portalsysteme bieten jedoch keine oder nur eingeschränkte Interaktion zwischen den Portlets. In [PrPe03] wurde daher ein kontextbasierter Integrationsansatz für Wissensportale entwickelt und im Rahmen des Wissensportalsystems INWISS[9] prototypisch implementiert. Dieser basiert auf einem Kommunikationsbus in Form eines Publish/Subscribe-Mechanismus, über den Portlets den aktuellen Benutzerkontext austauschen können. So werden beispielsweise durch die Navigation in einem OLAP-Bericht ausgelöste proaktive (implizite) Suchen in einer Dokumentenbasis möglich. Neben der Portlet-Integration beinhaltet der Prototyp einen unscharfen metadatenbasierten Suchmechanismus [PrSP04] und ein Sicherheitsmodul mit einem metadatenbasierten Zugriffskontrollmodell [PMDP04].

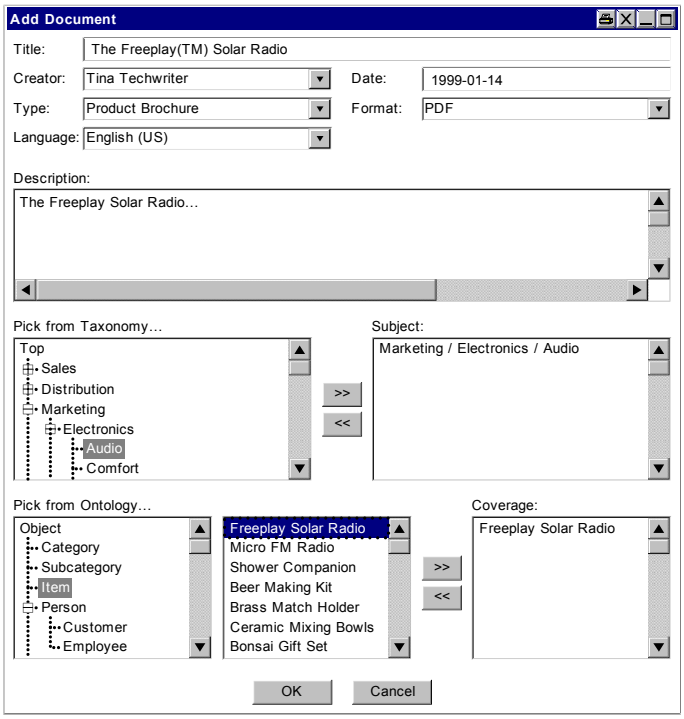

Abbildung 3: Portlet zur Metadatenpflege beim Anlegen eines neuen Dokuments

9 http://www.inwiss.org

Abbildung 3 zeigt ein Screen Design des Content-Management-Portlet, welches sich zurzeit in Entwicklung befindet. Es dient der Verwaltung von Intranet-Inhalten, die innerhalb des Portals dargestellt werden können, sowie von Dokumenten in Formaten wie PDF und Microsoft Word, für die externe Anwendungen benötigt werden. Sowohl interne Inhalte als auch externe Dokumente werden mit an den Dublin-Core-Standard angelehnten Metadaten versehen. Sie dienen zur Navigation (Taxonomie) und Suche. Insbesondere implizite (durch die genannte Kontextintegration angestoßene) Suchanfragen sind von der Existenz qualitativ hochwertiger Metadaten abhängig. Als Metdaten-Repository kommt das Sesame RDF Framework[10] zum Einsatz.

Beim Anlegen eines neuen Dokumentes müsste der Benutzer normalerweise sämtliche relevanten Metadatenfelder manuell ausfüllen. Mithilfe der in Abschnitt 4 dargestellten Verfahren kann nun eine semiautomatische Annotation ermöglicht werden, indem die Felder soweit möglich vorausgefüllt werden. Der Benutzer muss diese dann nur noch überprüfen und ggf. korrigieren.

6 Stand der Technik und verwandte Arbeiten

Mittlerweile gibt es einige kommerzielle Werkzeuge für Text Mining und Information Extraction. Prominente Beispiele sind der IDOL Server von Autonomy[11] und AeroText[12] von Lockheed Martin. Unterstützt werden die meisten der in Abschnitt 3 dargestellten Techniken: automatische Textklassifikation, Clustering, automatische Textzusammenfassung, Named Entity Extraction, etc.

Auf der Gegenseite gibt es eine Vielzahl an Systemen, die als Wissens-, Dokumenten- und Content-Management-Systeme, sowie Wissens- oder Unternehmensportale am Markt auftreten. Livelink von OpenText[13], Hyperwave[14] und das SAP Enterprise Portal[15] sind nur einige wenige Beispiele. Diese Systeme verwenden ebenfalls Metadaten zur Organisation der verwalteten Informationen und Dokumente. Hier bietet sich also eine automatische oder semiautomatische Annotation an. Jedoch gehört bislang einzig die automatische Klassifikation zur Einordnung der Texte in eine vordefinierte Taxonomie zum Stand der Technik. Sie wird prinzipiell von allen genannten Produkten unterstützt. Allerdings handelt es sich dabei meist um relativ einfache (auf dem Naive-Bayes-Verfahren basierende) Klassifikatoren. Aufgrund der nicht garantierbaren Klassifikationsqualität ist die Praxis-

10 http://www.openrdf.org
11 http://www.autonomy.com
12 http://mds.external.lmco.com/products/gims/aero/
13 http://www.opentext.com
14 http://www.hyperwave.com
15 http://www.sap.com/solutions/netweaver/enterpriseportal/

tauglichkeit der verfügbaren Mechanismen daher zum Teil fraglich, insbesondere da nur wenige Systeme, beispielsweise Livelink, einen semiautomatischen Ansatz verfolgen und eine Kontrolle durch den Benutzer vorsehen.

Im wissenschaftlichen Umfeld finden sich einige Arbeiten zur automatischen Erstellung einer Ontologie (z.B. [Maed02], ansatzweise auch [PaBe03]). Die Anwendung von Text-Mining-Techniken zur semantischen Annotation von Dokumenten wird jedoch weniger betrachtet. In [PaBe03] finden sich einige, allerdings wenig konkrete Vorschläge. Kao et al. [KQPW03] beschreiben einen semiautomatischen Ansatz zur Einordnung von Dokumenten in eine Taxonomie. S-CREAM [HaSC02] bietet einen Verfahren zur Verlinkung mit Elementen einer Ontologie, basierend auf dem Information-Extraction-System (vgl. Abschnitt 3.4) Amilcare[16].

7 Zusammenfassung und Ausblick

In diesem Beitrag haben wir Text-Mining- und Information-Extraction-Techniken auf ihre Tauglichkeit zur Erzeugung von semantischen Metadaten für Textdokumente untersucht. Angelehnt an den Dublin-Core-Metadatenstandard [DCMI03] haben wir auf automatischer Textkategorisierung basierende Verfahren für diverse Metadatenelemente entwickelt, die eine vollständige semiautomatische Annotation in Wissensmanagementsystemen unterstützen sollen, und skizziert, wie diese im Rahmen eines semiautomatischen Ansatzes in ein Wissensportalsystem integriert werden können.

Neuartig ist unseres Wissens insbesondere die vorgeschlagene schrittweise hierarchisch vorgehende Klassifikation zur Einordnung der Dokumente in eine Taxonomie (über das Dublin-Core-Element *Subject*). Diese verspricht eine deutliche Verbesserung der Klassifikationsqualität. Neben einer Einordnung in eine Taxonomie ist jedoch insbesondere die Verlinkung mit Elementen einer Ontologie von Bedeutung. Erst durch eine solche reichhaltige semantische Annotation sind intelligente (implizite, proaktive) Suchanfragen möglich. Diese (hier über das Dublin-Core-Element *Coverage* erfasste) Verlinkung bedeutet, wenn sie manuell erfolgen soll, jedoch einen sehr hohen Aufwand für die Benutzer. Der vorgestellte Ansatz über eine mehrfache binäre Klassifikation in Verbindung mit einem Semiautomatismus erscheint daher vielversprechend.

Momentan arbeiten wir an der Auswahl und Implementierung geeigneter Klassifikationsverfahren. Erste Überlegungen hierzu wurden bereits in Abschnitt 4 ausgeführt. Bei der Implementierung bauen wir auf der Lucene Text Retrieval Engine[17] der Apache Group auf. Zur empirischen Evaluation verwenden wir den bereits ge-

16 http://nlp.shef.ac.uk/amilcare/
17 http://jakarta.apache.org/lucene/

nannten Reuters 21578 Textkorpus[18], in dem 21.578 bereits annotierte Dokumente als Trainings- und Testmenge zur Verfügung stehen. Die Dokumente sind Themen zugeordnet (vergleichbar mit einer Taxonomie) und beinhalten Verweise auf in den Texten vorkommende Firmen, Organisationen, Personen und Orte (vergleichbar mit einer Ontologie).

Parallel läuft die Erweiterung der bislang ausschließlich metadatenbasierten Suchmaschine des INWISS-Portals um Volltextsuchfunktionalität, ebenfalls basierend auf der Lucene Engine. So können die implementierten Text-Mining-Verfahren wie in Abschnitt 5 angedacht Einzug in das Portal halten.

Literatur

[BaEl97] Barzilay, R.; Elhadad, M.: Using Lexical Chains for Text Summarization. Ben Gunon University in the Negev, Israel, 1997.

[BaRi99] R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval. Addison Wesley Longman Limited, Essex, 1999.

[BeHL01] Berners-Lee, T.; Hendler, J.; Lassila, O.: The Semantic Web. Scientific American, Mai 2001.

[DCMI03] Dublin Core Metadata Initiative. DCMI Metadata Terms. DCMI Recommendation, November 2003. http://dublincore.org/documents/2003/11/19/dcmi-terms/

[GrNi00] Grefenstette, G.; Nioche, J.: The WWW as a Resource for Lexicography. In M.-H. Corréard: Lexicography and Natural Language Processing, 2000, S. 199-215.

[KiBi04] Kiss, C.; Bichler, M.: Data Mining and Campaign Management in the Telecommunications Industry. In: Coordination and Agent Technology in Value Networks (MKWI 2004), GITO-Verlag, 2004.

[KQPW03] Kao, A.; Quach, L.; Poteet, S.; Woods, S.: User Assisted Text Classification and Knowledge Management. Proc. of the Twelfth International Conference on Information and Knowledge Management (CIKM 2004), New Orleans, LA, USA, November 2003.

[HaSC02] Handschuh, S.; Staab, S.; Ciravegna, F.: S-CREAM – Semi-automatic CREAtion of Metadata. Proc. of the European Conference on Knowledge Acquisition and Management - EKAW-2002. Madrid, Spain, October 1-4, 2002.

[Maed02] Maedche, A.: Ontology Learning For The Semantic Web. Kluwer Academic Publishers, 2002.

[18] Verfügbar unter http://www.daviddlewis.com/resources/testcollections/reuters21578/ und http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

[McGu02] McGuinness, D.L.: Ontologies Come of Age. In Fensel, D.; Hendler, J.; Lieberman, H.; Wahlster, W. (Hrsg.): Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential. MIT Press, 2002.

[MiST94] Michie, D.; Spiegelhalter, D.J.; Taylor, C.C.: Machine Learning, Neural and Statistical Classification, Ellis Horwood, New York, 1994.

[PaBe03] Paralic, J.; Bednar, P.: Text Mining for Document Annotation and Ontology Support. In: Intelligent Systems at the Service of Mankind, Ubooks, November 2003, S. 237-248.

[PMDP04] Priebe, T.; Muschall, B.; Dobmeier, W.; Pernul, G.: A Flexible Security System for Enterprise and e-Government Portals. Proc. of the 15th International Conference on Database and Expert Systems Applications (DEXA 2004), Zaragoza, Spanien, September 2004.

[PrPe03] Priebe, T.; Pernul, G.; Krause, P.: Ein integrativer Ansatz für unternehmensweite Wissensportale. Proc. 6. Internationale Tagung Wirtschaftsinformatik (WI 2003), Dresden, September 2003.

[PrSP04] Priebe, T.; Schläger, C.; Pernul, G.: A Search Engine for RDF Metadata. Proc. of the DEXA 2004 Workshop on Web Semantics (WebS 2004), Zaragoza, Spanien, September 2004.

[Quin86] Quinlan, J.R.: Induction of Decision Trees. In: Machine Learning, Volume 1, Number 1, Kluwer Academic Publishers, 1986.

[Seba02] Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys, Volume 34, Number 1, März 2002.

[ScDW01] Scheffler, T;. Decomain, C.; Wrobel, S.: Active Hidden Markov Models for Information Extraction. International Symposium on Intelligent Data Analysis (IDA), 2001.

[Tan99] Tan, A.: Text Mining: The State of the Art and the Challenges. Kent Ridge Digital Labs, Singapore, 1999.

[Vapn95] Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York, 1995.

[W3C04a] Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation, Februar 2004. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

[W3C04b] RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, Februar 2004. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

[W3C04a] OWL Web Ontology Language Overview. W3C Recommendation, Februar 2004. http://www.w3.org/TR/2004/REC-owl-features-20040210/

[WiFr00] Witten, I.; Frank, E.: Data Mining – Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers, 2000.