BINGO!
Ein thematisch fokussierender Crawler zur Generierung personalisierter Ontologien

Martin Theobald, Stefan Siersdorfer, Sergej Sizov

Universität des Saarlandes
Fachbereich Informatik
Postfach 151150, D-66041 Saarbrücken
{mtheobald; stesi; sizov}@cs.uni-sb.de

Zusammenfassung Fokussierendes Crawling ist ein viel versprechender Ansatz zur Verbesserung der *Ausbeute* einer Expertensuche über einem spezifischen Themenbereich des Webs. Dieses Verfahren beinhaltet die automatische Klassifikation von Dokumenten in eine benutzerspezifische Hierarchie von Themen, die wir auch als *Ontologie* bezeichnen. Die Qualität der Trainingsdaten des Klassifikators ist der kritischste Punkt für die Effektivität eines fokussierenden Crawlers. Der BINGO!-Ansatz versucht die Grenzen einer Trainingsbasis mit nur wenigen intellektuell kategorisierten Dokumenten zu überwinden und in einer automatisierten *Wachstumsphase* selbständig eine breite Trainingsbasis durch die Identifikation themenspezifischer "Archetypen" zu generieren. Die anschließende *Erntephase* vervollständigt dann die Ontologie nach iterativem Neutrainieren des Klassifikators mit einer verbesserten Ausbeute und Präzision.

1 Einführung

Typischerweise liefern Suchmaschinen sehr viele Dokumente mit einer gewissen Relevanz für ein gesuchtes Thema, allerdings muss sich der Benutzer oft noch manuell durch die Nachbarschaft dieser Ergebnisse klicken, um die gewünschte Seite zu finden. Oft können die besten Ergebnisse vielmehr über Portale wie `www.yahoo.com` erreicht werden, die den Zugriff auf ihre Verzeichnisse in Form einer intellektuell aufgebauten *Ontologie* bieten. Leider ist der Aufbau und die Wartung einer solchen Ontologie durch menschliche Experten sehr aufwendig und teuer, da enorme Datenmengen manuell gefiltert und klassifiziert werden müssen. An dieser Stelle tritt der fokussierende Crawler in Kraft: Er startet von einer benutzer- oder communityspezifischen Hierarchie von Themen mit nur wenigen Dokumenten pro Klasse und durchsucht selbständig das World-Wide-Web nach neuen Dokumenten zu diesen fokussierten Themen. Dieser Prozess kann entweder eine personalisierte, hierarchische Ontologie erzeugen, deren Knoten mit relevanten, qualitativ hochwertigen Dokumenten des gewünschten Themas besetzt werden, oder er kann dazu initialisiert werden, Expertenanfragen gezielt zu bearbeiten und deren Ergebnismengen automatisch zu vervollständigen.

2 Bookmark-Analyse

Der Benutzer initialisiert das System durch Einlesen seiner Bookmarks auf einer Ausgangsmenge von inhaltlich hochwertigen Beispieldokumenten[6], die die gesamte Trainingsbasis für die Berechnung des Klassifikators in der Wachstumsphase bilden. Da der SVM-Klassifikator zu jedem Knoten dieses Baums auf die Angabe von Positiv- und Negativbeispielen angewiesen ist, behandeln wir die Dokumente aller Nachbarkategorien einer Baumstufe als Negativbeispiele der zu modellierenden Klasse, was auch durch die Definition expliziter Sammelklassen mit Negativbeispielen ('OTHERS'-Knoten, Abb. 1) hervorgehoben werden kann.

Der *Dokument-Analysator*, ein erweiterter HTML-Parser, generiert Feature-Vektoren[5] aus allen Eingabedokumenten unter Berücksichtigung des von der Feature-Selektion bestimmten Feature-Raumes jedes Hierarchieknotens. Stoppwörter werden entfernt und Porter-Stemming reduziert die morphologischen Varianten der übrigen Terme[4, 5]. Gewichtsvektoren mit TF/IDF-Gewichtung[5] repräsentieren die Dokumente und dienen so als direkte Eingabe für den Klassifikator.

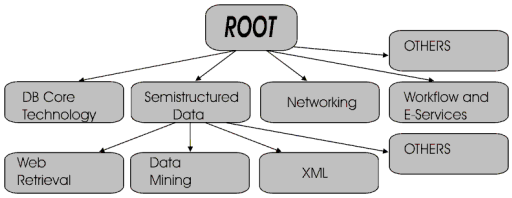

Abbildung1. Beispiel einer Ontologie mit Tiefe 2

3 Zweistufige Feature-Selektion

Neben der TF/IDF-Gewichtung von Termen in den Feature-Vektoren dient das klassenspezifische IDF-Maß[5] als Kriterium für die Feature-Selektion erster Stufe zur Effizienz-Steigerung des Gesamtverfahrens. Als Feature-Selektion der zweiten Stufe hat sich die Berechnung der *Kullback-Leibler-Distanz* (bzw. *Mutual Information MI*) als statistisches Maß für die Diskriminationsgüte eines Features im Hinblick auf die zur Auswahl stehenden Klassen einer Baumstufe der Ontologie bewährt. Als Entropiemaß lässt sich MI zur Generierung der Feature-Räume sowohl von Blattklassen als auch von inneren Knoten des Hierarchiebaums verwenden.

Neben der Reduktion von Rauschen unter den Dokumenttermen kann durch die Selektion der d besten Diskriminatoren jeder Klasse die entsprechende Dimensionalität d der Feature-Vektoren drastisch eingeschränkt (in unseren Experimenten auf $d = 300$) und so die Effizienz des SVM-Verfahrens insbesondere in der Lernphase verbessert werden.

4 Hierarchische SVM-Klassifikation

In seiner einfachsten, linearen Form definiert ein SVM-System als Menge von Stützvektoren eine *Hyperebene* mit Normalvektor w, die zwei komplementäre Klassen

von Trainingsdaten C und $\neg C$ mit maximalem Euklidischem Abstand δ trennt[1, 2]. Den beiden Klassen liegen Trainingsvektoren x_i zugrunde, die sich auf eine Menge von Positivbeispielen für C und eine Menge von Negativbeispielen für $\neg C$ aufteilen. Nach der Berechnung von w bildet eine Entscheidungsfunktion $f(z) = (w \cdot z) + b = \sum_{i=1}^{d} w_i z_i + b \ > 0$ in linearer Zeit $O(d)$ den binären Klassifikator eines Hierarchieknotens.

Zur Klassifikation von Dokumenten in eine mehrstufige Ontologie wird der Hierarchiebaum rekursiv traversiert, wobei in jedem inneren Knoten eine binäre SVM-Klassifikation über die Wahl des Nachfolgers entscheidet. Der Betrag $f(z)$ der Entscheidungsfunktion für positiv klassifizierte Dokumente dient als Maß für die Klassifikationskonfidenz in jedem Knoten und wird im folgenden als *SVM-Konfidenz* bezeichnet. Ist die SVM-Konfidenz für mehrere Knoten einer Baumstufe positiv, so wird als einfachste Lösung[1] derjenige Knoten mit der größten SVM-Konfidenz in den Ergebnispfad der hierarchischen Klassifikation aufgenommen und als Elternknoten des nächsten Rekursionsschrittes gewählt.

Die Menge SVM_A der Dokumente jeder Klasse mit der höchsten SVM-Konfidenz bildet die *erste Gruppe potentieller Archetypen*, die für eine Erweiterung der Trainingsbasis in Frage kommen.

5 Fokus-Management

5.1 Wachstumsphase

Vom Crawler erfasste und heruntergeladene Dokumente werden unmittelbar nach ihrem Download mit Hilfe des hierarchischen SVM-Klassifikators klassifiziert. Unser besonderes Interesse dabei gilt solchen Zieldokumenten, die *derselben Klasse wie das Quelldokument* eines Links zugeordnet werden. In der Wachstumsphase möchten wir also ausgehend von den Bookmarks ganze Pfade von Dokumenten verfolgen, die alle derselben (Ober-)Kategorie angehören und evtl. einen bestimmten Schwellwert in ihrer Klassifikationskonfidenz überschreiten können. Diese Strategie nennen wir *starke Fokussierung*.

5.2 Erntephase

Das starke Fokussierungsschema auf Linkfolgen derselben Klasse erhöht die Präzision des Crawlers insbesondere bei einer noch niedrigen Klassifikationskonfidenz ausgehend von wenigen initialen Bookmarks. Zur Steigerung der *Ausbeute* des Crawlers in der Erntephase kann diese starke Fokussierung abgeschwächt werden, indem der Fokus-Manager Dokumente aller Blattklassen des Hierarchiebaums unabhängig von der Klasse des Vorgängers akzeptiert und nur Dokumente aus 'ROOT/OTHERS/' ablehnt. Diese Strategie nennen wir *schwache Fokussierung*.

In der Praxis kann die starke Fokussierung nach der Initialisierung des Systems auf den Bookmarks im ersten Schritt dazu verwendet werden, neue Archetypen trotz einer noch geringen Klassifikationskonfidenz zu finden. Nach dem Neu-Trainieren des Klassifikators auf Bookmarks *und* Archetypen kann dann auf das schwache Fokussierungsschema umgeschaltet werden, um mit einer hohen Ausbeute Benutzeranfragen zu bearbeiten.

6 Link-Analyse nach HITS

Die Analyse der Linkstruktur zwischen den Dokumenten einer Klasse bietet eine zusätzliche Quelle dafür, wie gut diese das Thema erfassen. Wir wenden dazu Kleinberg's bekanntes HITS-Verfahren[7] auf den Hyperlink-Graphen $G = (V, E)$ der Dokumente einer Ontologieklasse an. Diese Methode dient der Identifikation einer Menge $HITS_A \subseteq V$ guter Authorities, die diejenigen Dokumente mit den besten Hyperlink-Referenzen unseres Web-Ausschnitts enthält. $HITS_A$ dient so als *zweite potentielle Quelle für neue Archetypen*, die unabhängig von der inhaltlichen Bewertung durch den SVM-Klassifikator ist.

7 Neu-Training basierend auf Archetypen

Wenn eine Klasse unserer Ontologie einen gewissen Füllstatus von N_{max} Dokumenten erreicht hat, beispielsweise wenn ein Thema mehrere hundert neue Dokumente enthält, kann ein Neu-Trainieren des Klassifikators ausgelöst werden. Um sicherzustellen, dass wir tatsächlich einen ausreichenden Crawling-Fortschritt erreicht haben, stellen wir die Zusatzbedingung, dass alle Klassen unserer Ontologie mindestens N_{min} Dokumente enthalten, was wir durch eine dynamische Verlagerung der Fokussierung auf schwach besiedelte Klassen erreichen können.

Ein vollständig automatisiertes Neu-Trainieren birgt die Gefahr des Themen-Drifts. Um dieses Phänomen zu vermeiden, wählen wir nur die besten Dokumente aus $HITS_A \cap SVM_A$, was gerade der Schnittmenge aus besten *Authorities* mit den durch den Klassifikator am eindeutigsten bewerteten Dokumenten entspricht. Als Zusatzbedingung stellen wir außerdem die Forderung, dass diese neuen Archetypen eine SVM-Konfidenz haben müssen, die mindestens der durchschnittlichen SVM-Konfidenz der Bookmarks entspricht.

8 Experimentelle Evaluation

Die folgenden Ergebnisse beschreiben die intellektuelle Auswertung von insgesamt 7 Iterationen über unserer Beispiel-Ontologie, während denen insgesamt 4200 Dokumente erfasst wurden (Neu-Training nach jeweils 600 neu erfassten Dokumenten).

Tabelle 1 zeigt die Präzision des Crawlers für die beiden Beispielklassen *Root/ Semistructured Data/Data Mining/* (aus 10 initialen Bookmarks) und *Root/Semistructured Data/XML/* (aus 9 initialen Bookmarks) als Makro-Durchschnitt über die Gesamt-Ontologie inklusive 'OTHERS'-Klassen (insgesamt 81 Bookmarks). Dahinter steht jeweils die Anzahl der neu gefundenen Archetypen, wobei die Anzahl in Klammern die fälschlicherweise als Archetypen identifizierten, aber nicht für das Thema relevanten Dokumente wiedergibt. Unsere Auswertung zeigt, dass diese Anzahl als Folge unserer strikten Auswahlkriterien in allen Iterationen gering bleibt. Auf der anderen Seite zeigt sich aber auch, dass noch nicht alle vom Crawler gefundenen Archetypen vom System sicher erkannt und in die Trainingsbasis aufgenommen werden.

Um die erfolgreiche Identifikation neuer Archetypen zu illustrieren, zeigt Tabelle2 die 10 besten Trainingsdokumente der Klasse *Data Mining* nach der letzten Iteration inklusive der neu entdeckten Archetypen (unterstrichen) sortiert nach ihrer SVM-Konfidenz.

Iteration	Data Mining	XML	Ges. Ontologie
1	0,98 10(1)	0,94 5(0)	0,98 24(4)
2	0,98 10(2)	0,93 11(0)	0,98 27(5)
3	0,99 9(1)	0,97 17(1)	0,96 32(4)
4	0,87 8(0)	0,99 7(0)	0,97 29(3)
5	0,90 22(2)	0,95 26(2)	0,96 62(8)
6	0,98 43(4)	0,98 12(2)	0,95 77(10)
7	0,94 38(0)	0,97 13(1)	0,96 75(8)

Tabelle1. Die BINGO! Klassifikations-Präzison (starke Fokussierung)

URL	SVM-K
http://www.it.iitb.ernet.in/~sunita/it642/	1.35
http://www.research.microsoft.com/research/datamine/	1.31
http://www.acm.org/sgs/sigkdd/explorations/	1.28
http://robotics.stanford.edu/users/ronnyk/	1.24
http://www.kdnuggetts.com/index.html	1.18
http://www.wizsoft.com/	1.16
http://www.almaden.ibm.com/cs/people/ragrawal/	1.14
http://www.cs.sfu.ca/~han/DM_Book.html	1.14
http://db.cs..sfu.ca/sections/publication/kdd/kdd.html	1.14
http://www.cs.cornell.edu/johannes/publications.html	0.78

Tabelle2. Beste 10 Archetypen der Klasse "*Data Mining*"

9 Fazit

Durch die Implementierung des BINGO!-Systems haben wir unser primäres Ziel, benutzer- oder community-spezifische Ontologien ausgehend von einer kleinen Menge von initialen Bookmarks aufzubauen, erreicht. Allerdings zeigen unsere Experimente, dass das System noch nicht alle im Crawling-Bereich erfassbaren Archetypen verlässlich identifiziert und Leistungssteigerungen hinsichtlich der Itertationspräzision und -ausbeute durch eine zuverlässigere Destillation der Dokumentenbasis, etwa durch eine Verfeinerung des HITS-Verfahrens[7], zu erwarten wären.

Literatur

1. V. Vapnik (1998): *Statistical Learning Theory*, John Wiley NY, 1998
2. C.J.C. Burges: *A Tutorial on Support Vector Machines for Pattern Recognition*, Data Mining and Knowledge Discovery Vol.2 No.2, 1998
3. H. Chen, S. Dumais: *Bringing Order to the Web: Automatically Categorizing Search Results*, ACM CHI Conference on Human Factors in Computing Systems, 2000
4. C.D. Manning, H. Schuetze: *Foundations of Statistical Natural Language Processing*, MIT Press, 1999
5. R. Baeza-Yates, B. Ribeiro-Neto: *Modern Information Retrieval*, Addison-Wesley, 1999
6. S. Chakrabarti, M. van den Berg, B. Dom: *Focused Crawling: A new Approach to Topic-Specific Web Resource Discovery*, Proc. World Wide Web Conference (WWW8), 1999
7. J.M. Kleinberg: *Authoritative Sources in a Hyperlinked Environment*, Journal of the ACM Vol.46, No.5, 1999