# Konstruktion von Featureräumen und Metaverfahren zur Klassifikation von Webdokumenten

Stefan Siersdorfer, Sergej Sizov
{*siersdorfer, sizov*}*@cs.uni-sb.de*

Datenbanken und Informationssysteme
Universität des Saarlandes
66123 Saarbrücken, Deutschland
*http://www-dbs.cs.uni-sb.de*

**Abstract:** Dieses Papier befasst sich mit der automatischen Klassifikation von Webdokumenten in eine vorgegebene Taxonomie. Wir betrachten dabei vektorbasierte Verfahren des maschinellen Lernens am Beispiel von SVM (Support Vector Machines). In diesem Papier beschreiben wir Möglichkeiten zur Generierung von Featurevektoren unter Berücksichtigung der Besonderheiten von Webdokumenten für solche Verfahren. Weiterhin untersuchen wir die Berechnung von Metaresultaten aus den partiellen Klassifikationsergebnissen.

## 1 Einführung und Grundlagen

### 1.1 Problemstellung

Die Klassifikation von Webinhalten gehört zu den wichtigen Aufgaben des Web Mining. Konventionelle Klassifikationsstrategien basieren auf Verfahren des maschinellen Lernens und verwenden Term-basierte Featurevektoren bei Aufbau und Anwendung des Klassifikationsmodells. Dabei bleiben weitere Aspekte (Umgebung von Webdokumenten, strukturelle Besonderheiten etc.) typischerweise unberücksichtigt.

Diese Arbeit betrachtet unterschiedliche Verfahren zur Generierung von Featurevektoren und deren Zusammenspiel im Rahmen von Meta-Klassifikationsstrategien.

### 1.2 Dokumentverarbeitung

Um das Klassifikationsverfahren anwenden zu können, müssen wir Dokumente zunächst in Vektoren transformieren. Wir verarbeiten Dokumente in folgenden 3 Schritten mit im Information Retrieval üblichen Methoden:

1. Parsen des Dokuments
2. Elimination von Stoppwörtern
3. Reduktion der Terme auf ihre Stammformen. Wir verwenden den Stemming-Algorithmus nach Porter [Pora, Porb].
4. Berechnung der Feature-Vektoren. Entsprechende Verfahren werden in Kapitel 2 näher betrachtet.

### 1.3 Hierarchische Klassifikation

Wir betrachten den Taxonomiebaum der benutzerspezifischen Themen (Abbildung 1). Jedem Knoten ist eine Menge von intellektuell bestimmten Trainingsdokumenten zugeordnet. Für alle Knoten außer "ROOT" berechnen wir nun einen SVM-Klassifikator. Für eine Klasse $K$ betrachten wir dabei die Dokumente aus $K$ als Positivbeispiele, die Dokumente aus den Nachbarklassen von $K$ mit dem selben Vater wie $K$ ("Gegnerklassen" von $K$) als Negativbeispiele. Ein neues Dokument können wir nun klassifizieren, indem wir den Baum ausgehend von der Wurzel traversieren und die Klassifikationen mittels der einzelnen Knotenmodelle durchführen. Wird ein Dokument dabei ausgehend von einer Oberkategorie in mehrere Unterkategorien klassifiziert, so wählen wir den Knoten mit der höchsten Klassifikationskonfidenz (im Falle von SVM: der größte Abstand von der Hyperebene). Wird das Dokument in keine der Unterkategorien positiv klassifiziert, so ordnen wir das Dokument einer Sonderklasse "OTHERS" zu. Wir verwenden linea-

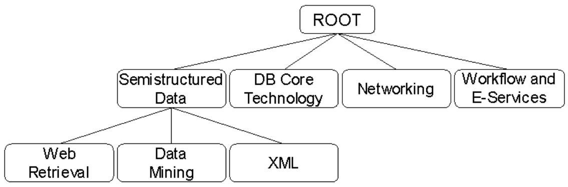


Abbildung 1: Beispieltaxonomie

re Support Vector Machines (SVM) [Bur98, Vap98] als themenspezifischen Klassifikator. Diese Methode hat sich als effizient und effektiv für die Textklassifikation erwiesen (siehe [DC00, CD00, Joa98]). Das Training besteht dabei in der Berechung einer trennenden Hyperebene im $m$-dimensionalen Featureraum, die eine Menge von positiven Trainingsbeispielen von einer Menge von negativen Beispielen trennt (Abbildung 2). Die Hyperebene kann in der Form $\vec{w}\vec{x} + b = 0$ beschrieben werden. Die Parameter $\vec{w}$ und $b$ der optimalen Hyperebene werden bei SVM nun so bestimmt, dass der Euklidische Abstand $\delta$ der nächstgelegenen Vektoren von der Hyperebene maximiert wird:

$$C_i \frac{1}{||\vec{w}||} (\vec{w}\vec{x}_i + b) \geq \delta \qquad (1)$$

für alle $i$, wobei $\vec{x}_i \in \mathbf{R}^m$ das $i$-te Trainingsbeispiel ist und $C_i \in \{1, -1\}$ beschreibt, ob $x_i$ ein positives ($C_i = 1$) oder ein negatives ($C_i = -1$) Beispiel ist.

### 1.4 Featureselektion

Featureselektion reflektiert die Annahme, dass einige Terme irrelevant für die Klassifikation sind und daher bei der Berechnung von Featurevektoren ignoriert werden können. Der Featureselektionsalgorithmus sollte für eine gegebene Klasse die charakteristischsten Features auswählen. Ein gutes Feature sollte eine Klasse gut von seinen "Gegnerklassen" unterscheiden. Daher sollte Featureselektion themenspezifisch sein: sie wird individuell für jede Klasse des Ontologiebaums durchgeführt.

Wir verwenden das Mutual Information (MI)- Kriterium für themenspezifische Features. Diese Technik, die eine Spezialfall von Kreuzentropie oder Kullback-Leibler Divergenz [MS99] ist, ist als eine der effektivsten Methoden bekannt [YP97]. Die MI-Gewichtung eines Terms $X_i$ und einer Klasse $V_j$ ist definiert durch:

$$MI(X_i, V_j) = P[X_i \wedge V_j] \log \frac{P[X_i \wedge V_j]}{P[X_i] P[V_j]} \qquad (2)$$

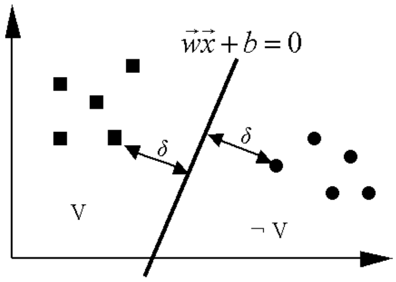


Abbildung 2: Separierende Hyperebene eines linearen SVM-Klassifikators

Mutual Information kann als Maß dafür interpretiert werden, wie stark sich die gemeinsame Verteilung der Features $X_i$ und der Klassen $V_j$ von einer hypothetischen Verteilung unterscheiden, in denen Features und Klassen unabhängig voneinander sind.

## 2 Konstruktion von Featureräumen

### 2.1 Einzeltermfeatureräume

Zu den einfachsten Möglichkeiten zur Konstruktion von Featurevektoren gehören Verfahren, die auf den relativen Häufigkeiten von Einzeltermen in einem Dokument basieren. Hier beschreiben wir zwei einfache Varianten zur Konstruktion von Einzeltermvektoren mit Hilfe von MI.

**Strategie der klassenweise besten Terme** Zur Berechung des Vektors zu einem Dokument $D$ bezüglich Klasse $K$ wählen wir durch MI-Selektion eine Menge von $m$ charakteristischen Termen $\{t_1, \ldots, t_m\}$. Wir erzeugen den Featurevektor aus den relativen Häufigkeiten $tf(D, t_i)$ der $t_i$ in $D$:

$$\big(tf(D, t_1), \ldots, tf(D, t_m)\big) \tag{3}$$

Da die Terme aus $T(K)$ $K$ nach Konstruktion besonders gut von ihren "Gegnerklassen" beim Training unterscheiden, werden die Vektoren zu Dokumenten aus $K$ eher *dicht* besetzt sein, die Vektoren aus den Nachbarklassen hingegen eher *dünn* besetzt sein. "Dicht besetzt" soll hier heißen, dass viele Komponenten einen relativen großen Wert $> 0$ haben. "Dünn besetzt" heißt, dass viele Komponenten den Wert 0 haben (keine Terme im Dokument vorhanden) bzw. eher kleinere Werte für die relative Häufigkeit besitzen. Dadurch erreichen wir eine gute Trennbarkeit.

**Strategie der Kontrastterme** Die Idee der *Konstraststrategie* für Einzelterme ist, charakteristische Terme für Nachbarklassen bei der Modellbildung für eine Klasse $K$ mit einzubeziehen. Diese Terme der Nachbarklassen bezeichnen wir als *Kontrastterme*. Die Featurevektoren konstruieren wie analog zu den Vektoren aus den klassenweise besten Einzeltermen. Ziel ist es, Vektoren zu konstruieren, die mehrere unterschiedlich dicht besetzte Bereiche haben. Dabei gibt es Bereiche von Komponenten, die bei Positivvektoren typischerweise dicht besetzt, bei Negativvektoren hingegen dünn besetzt sind und umgekehrt. Dadurch wollen wir wieder eine bessere Trennbarkeit erreichen.

## 2.2 Paarstrategien

**Grundideen** In vielen Dokumenten tauchen zusammengehörige Begriffe auf (z.B. "network" und "protocol"). Außerdem existieren viele Begriffe, die in bestimmten Kontextbereichen häufig gemeinsam auftreten. Wir können also versuchen, Textterme zu Tupeln ($n$-*Grammen*) zusammenzufassen.

Der Ansatz, alle $n$-Tupel im Termraum zu betrachten scheitert, da deren Anzahl exponentiell ist. Wir wollen uns im Folgenden auf $n = 2$, also auf *Paare* beschränken. Dies rechtfertigen wir zum einen durch die Beobachtung, dass wir signifikante Häufigkeiten in Dokumenten seltener bei höher dimensionierten Tupeln finden können. Außerdem impliziert eine Korrelation von $n$ Termen auch eine Korrelation der $\binom{n}{2}$ daraus bildbaren Paare.

Schränken wir die Paarmenge nicht ein, ist deren Anzahl quadratisch in der Größe des Termraums. Die Bildung von Vektoren und die Anwendung des SVM-Verfahrens wären somit mit einem erheblichen Berechnungsaufwand verbunden. Ein weiteres Problem ist die Existenz von verschiedenen *Kontextbereichen* in Dokumenten. Ohne Einschränkungen erhalten wir somit Paare mit Termen aus jeweils unterschiedlichen Sinnabschnitten.

**Paarräume** Formal stellen wir Dokumente $D$ als Menge von Tupeln $(t, pos)$ dar, wobei $t$ der Term selbst und $pos$ die *Position* des Terms im Dokument ist. Die für uns relevanten Informationen über ein Termpaar können wir in der Form

$$(t_a, t_b, pos_a, pos_b) \tag{4}$$

darstellen. $t_a$ und $t_b$ sind dabei die Terme, $pos_a$ und $pos_b$ sind die Positionen von $t_a$ bzw. $t_b$ im betrachteten Dokument.

Es soll $t_a \neq t_b$ gelten, d.h. wir assozieren keine Terme mit sich selbst. Wir behandeln Paare symmetrisch , d.h. wir machen keinen Unterschied zwischen $(t_a, t_b)$ und $(t_b, t_a)$. Um eine kanonische Form zu erhalten, legen wir fest, dass gelten soll: $t_a <_{lex} t_b$.

Für eine Klasse $K$ können wir den Paarraum einschränken, indem wir nur Paare über einer Menge $\tilde{T}(K) = \{t_1, \ldots, t_q\}$ von Termen bilden, die wir durch eine der beschriebenen Einzeltermstrategien bestimmen.

Bei der *Sliding-Window* -Strategie betrachten wir nur Paare aus Termen, die sich innerhalb eines gedachten Fensters der Größe $dis$ befinden, dass wir über den Text eines Dokumentes bewegen:

$$\forall (t_a, t_b, pos_a, pos_b) \in D : |pos_a - pos_b| < dis \tag{5}$$

Insgesamt definieren wir die Menge $P(K)$ der für die Modellierung von $K$ relevanten Paare bei Verwendung der Trainingsdokumentmenge $Tr(K)$:

$$\begin{aligned} P(K) \quad := \quad \Big\{ (t_a, t_b) \quad \Big| \quad & t_a \neq t_b \quad \wedge \quad t_a <_{lex} t_b \\ \wedge \quad \Big( & \exists D \in Tr(K), pos_a, pos_b : \\ & (t_a, t_b, pos_a, pos_b) \in D \wedge |pos_a - pos_b| < dis \Big) \Big\} \end{aligned} \tag{6}$$

bzw. bei Einschränkung der Einzelterme auf $\tilde{T}(K)$ die Menge $\tilde{P}(K)$:

$$\tilde{P}(K) = \{(t_a, t_b) \in P(K) | t_a \in \tilde{T}(K) \wedge t_b \in \tilde{T}(K)\} \tag{7}$$

Wir könnten für ein Paar $(t_a, t_b)$ nun einfach die Anzahl der Tupel $(t_a, t_b, pos_a, pos_b)$ in einem Dokument $D$ zählen. Dann kann allerdings z.B. folgende Situation auftreten: Gehen wir von einem Term-Positionspaar $(t_a, pos_a)$ in $D$ aus, so kann ein Term-Positionspaar $(t_b, pos_b)$ und $(t_b, pos'_b)$ existieren, so dass $pos'_b > pos_b$. Das Paar $(t_a, t_b)$ würde also doppelt gezählt. Die Vermutung liegt jedoch nahe, dass $t_a$ eher mit dem näher gelegenen der beiden $t_b$ assoziiert werden kann. Daher zählen wir in solchen Fällen im Dokument weiter auseinander liegende Paare nicht mit. Diese bezeichnen wir als *Pseudopaare* in $D$, $Pseudo(D)$ als deren Menge.

Wir erweitern den Häufigkeitsbegriff auf Paare. Für $(t_a, t_b) = p \in P(K)$ definieren wir die absolute Häufigkeit in Trainingsdokument $D$ als:

$$\begin{aligned} h_{abs}(p, D) \quad = \quad & \Big| \{(w_s, w_t, k, l) \in D \quad | \\ & \quad w_s = t_a \wedge w_t = t_b \wedge (w_s, w_t, k, l) \notin Pseudo(D) \wedge |k - l| < dis\} \Big| \end{aligned} \tag{8}$$

Die Komponenten des Featurevektors für $D$ bestehen nun aus den daraus abgeleiteten relativen Häufigkeiten der Paare aus $\tilde{P}(K)$.

Wir können die Menge der Paare noch weiter durch Featureselektion für *Paare* eingrenzen. Analog zu Einzeltermen verwenden wir dazu die Strategie der klassenweise besten *Paare* und die Strategie der Kontrast*paare*.

## 2.3 Anchorterme

*Anchortexte* sind die Texte zu Links in HTML-Seiten. Häufig liefern sie kurze und prägnante Beschreibungen der Seiten, auf die die entsprechenden Links zeigen. Wir wollen daher auch *Anchorterme* zur Modellbildung und Klassifikation von Seiten verwenden. Dabei wollen wir nur die Anchortexte von Seiten betrachten, die auf die interessierende Seite $D$ zeigen. Diese Links beziehen sich unabhängig vom Inhalt der Seite *von* der verwiesen wird auf die Seite $D$.

Die resultierende (Multi)-Menge der Anchorterme können wir nun im Prinzip wie ein virtuelles Einzeltermdokument behandeln. Dabei ist eine Elimination von Stoppwörtern notwendig um Artefakte wie "*click here*" zu vermeiden. Das Hauptproblem ist allerdings dass die Mengen der Anchorterme häufig sehr klein sind. MI-Featureselektion, die statistisch begründet ist, greift daher oft nur schlecht. Daher wird eine reine Anchortermstrategie wahrscheinlich wenig aussichtsreich sein. Trotzdem erlaubt dieser Ansatz sinnvolle Kombinationen mit anderen Strategien (siehe Abschnitt 2.5). Dies unterscheidet unsere Vorgehensweise von verwandten Klassifikationsmethoden für Webdaten [CJT01].

## 2.4 Dokumentvereinigung

Die Idee bei der *Dokumentvereinigung* besteht darin, die Vorgänger und Nachfolger eines Dokuments $D$ zur Konstruktion von Featurevektoren mit einzubeziehen. Nachbardokumente haben häufig mit dem Inhalt von $D$ und damit der Kategorie von $D$ zu tun. Die Qualität von Nachbardokumenten lässt daher oft Rückschlüsse auf die Qualität von $D$ zu.

Der einfachste Weg wäre nun die Gesamtheit der zu vereinigenden Dokumente wie ein einziges zu behandeln und die entsprechenden Einzeltermstrategien aus Abschnitt 2.1 anzuwenden. Leider führt diese Strategie in primitiver Form zu unbefriedigenden Klassifikationsergebnissen [CJT01]. Wir wollen allerdings vermeiden, dass Dokumente, die groß sind relativ zu anderen Dokumenten der Vereinigung, eine unverhältnismäßige Rolle spielen. Dies können wir durch einfache Termgewichtungen nicht erreichen. Statt die Terme der Dokumente zu gewichten, können wir dies aber auch mit den Dokumenten selbst tun. Wir bilden also einen gewichteten Mittelwert über die $tf$-Werte der Einzeldokumente.

Zum Bilden eines SVM-Modells ist die Dokumentvereinigung nur bedingt geeignet, da nur in sehr seltenen Fällen die Klasse eines Dokuments $D$ mit den Klassen *aller* Nachbarn übereinstimmen wird. Somit ist meist auch unter Einbeziehung von Featureselektion eine Verschlechterung der Modellqualität zu erwarten. Modellierung mittels Dokumentvereinigung sollte höchstens als Ergänzung im Rahmen von Metaverfahren (siehe Kapitel 3) angewendet werden.

### 2.5 Kombinationsräume

**Einzelterme und Paare** Bei der Bildung reiner Paarvektoren können Informationen über separat auftretende Einzelterme schnell verloren gehen. Bei den Einzeltermstrategien bleibt dagegen die Anordnung der Terme im Dokument unberücksichtigt. Aus diesen Gründen ist es nahe liegend, Einzeltermvektoren und Paarvektoren zu kombinieren.

Sei also eine zu untersuchende Klasse $K$, eine Menge $\tilde{T} = \{t_1, \ldots, t_n\}$ von relevanten Einzeltermen und eine Menge $P(K) = \{p_1, \ldots, p_m\}$ von relevanten Paaren gegeben. Dann können wir für ein Dokument $D$ Featurevektoren der folgenden Form bilden:

$$(tf(t_1, D), \ldots, tf(t_n, D), \ldots, tf(p_1, D), \ldots, tf(p_m, D)) \quad (9)$$

**Anchors und Terme** Wir haben schon erwähnt, dass Anchortermvektoren für sich gesehen wohl wenig Sinn machen. Als Gründe hatten wir dünne Besetzung und unterschiedliche Verfügbarkeit genannt. Einzeltermvektoren und Anchortermvektoren analog zum oberen Abschnitt zusammenzulegen, scheint aus demselben Grund wenig viel versprechend. Wir beziehen die Anchorterme stattdessen linear mit einer Gewichtung in die Berechnung der relativen Häufigkeiten der Dokumentterme (Einzelterme) mit ein.

## 3 Metaverfahren

Die Idee von *Metaverfahren* ist es nun, mehrere Verfahren auf eine gegebene Problemstellung anzuwenden. Diese Idee ist verwandt mit der Technik der zusammengesetzten Kernel [JCST01]. Es wurde allerdings gezeigt, dass diese Technik bei dünn besetzten Featureräumen mit vielen Stützvektoren (spezifisch für Webdaten) und stark unterschiedlicher Präzision einzelner Klassifikationsstrategien keine Verbesserung der resultierenden Klassifikationsgüte erzielt. Statt dessen versuchen wir, auf die *Ergebnisse* einzelner Verfahren unser Metaverfahren anzuwenden und ermitteln so das *Metaresultat*.

Sei eine Menge von $n$ unterschiedlichen Verfahren $V = \{v_1, \ldots, v_n\}$ zur Konstruktion von Termfeatureräumen und damit zur Klassifikation von Dokumenten gegeben. $Res(v_i, D, K)$ bezeichne das Resultat der Klassifikation von Dokument $D$ bei Betrachtung der Klasse $K$ und Verwendung des Verfahrens $v_i$. Dabei soll $Res(v_i, D, K) = +1$ gelten, falls $D$ durch $v_i$ in $K$ klassifiziert wurde, $Res(v_i, D, K) = -1$ sonst.

Wir können den Verfahren aus $V$ nun eine Gewichtung $w(v_i) \in \mathbf{R}_+$ zuordnen, die ein Maß für die "Bedeutung" sein soll, die wir dem Verfahren zuordnen. Die Qualität eines Verfahrens können wir z.B. durch die Auswertung von Experimenten abschätzen.

Damit können wir eine *Metaresultatsfunktion* $Meta(V, D, K)$ zur Klassifikation eines Dokumentes $D$ bei Betrachtung einer Klasse $K$, Anwendung der Verfahrensmenge $V$ und

Schranken $t_1, t_2 \in \mathbf{R}$ mit $thres_1 \geq thres_2$ definieren:

$$Meta(V, D, K) = \begin{cases} +1 & \text{falls } \sum_{i=1}^{n} Res(v_i, D, K) \cdot w(v_i) > t_1 \\ -1 & \text{falls } \sum_{i=1}^{n} Res(v_i, D, K) \cdot w(v_i) < t_2 \\ 0 & \text{sonst} \end{cases} \quad (10)$$

Dabei soll $Meta(V, D, K) = +1$ bedeuten, dass $D$ nach dem Metaverfahren in $K$ liegt, $Meta(V, D, K) = -1$ soll bedeuten, dass $D$ nicht in $K$ liegt. Nimmt die Metaresultatsfunktion den Wert 0 an, so treffen wir keine Aussage über das Klassifikationsergebnis.

Wir wollen drei wichtige Spezialfälle betrachten.

- "Einstimmigkeit": Wir verlangen, dass die Resultate aller Verfahren aus $V$ für die Klassifikation von Dokument $D$ bezüglich Klasse $K$ übereinstimmen sollen. Das Metaresultat soll gleich dem Resultat der Verfahren sein. Liefern die Verfahren keine einheitlichen Resultate, so treffen wir keine Aussage über das Klassifikationsergebnis. (Parameterisierung: $w(v_i) = 1 \quad \forall v_i \in V, \quad t_1 = n - 0.5 = -t_2$).
- "Mehrheitsentscheid": Wir übernehmen das Resultat, das die Mehrheit der Verfahren aus $V$ liefert, als Metaresultat. ($w(v_i) = 1 \quad \forall v_i \in V, \quad t_1 = t_2 = 0$.)
- Gewichtung durch $\xi\alpha$-Estimatoren: Diese von Joachims für SVM eingeführte Estimatoren $PR^{\xi\alpha}$ für die Präzisionsqualität stellen eine effiziente Abschätzung für leave-one-out Verfahren zur Verfügung [Joa00]. ($w(v_i) = PR^{\xi\alpha}(v_i) \ \forall v_i \in V, \ t_1 = t_2 = 0$)

Statt bei der Metaresultatsfunktion über $\pm 1$ zu summieren könnte man auch über die jeweiligen Konfidenzen (Abstand zu der Hyperebene) summieren. Die Abstände für ein Verfahren, könnte man z.B. normieren, indem man diese durch den mittleren Abstand von der Hyperebne bei Anwendung des Verfahrens auf die Trainingsdaten dividiert.

Je nach Wahl von $V$ und der Parameter erhalten wir unterschiedlich restriktive Verfahren. Diese Restriktivität führt zwar dazu, dass ein bestimmter Anteil an zu klassifizierenden Dokumenten verworfen wird; allerdings zeigen Experimente (siehe Kapitel 4), dass sich die Qualität der Klassifikation der restlichen Dokumente erhöht.

# 4 Experimente

## 4.1 Versuchsaufbau

In unseren Experimenten verwendeten wir für die Aquisition der Webdaten den fokussierten Crawler BINGO! [SSTW02] in Verbindung mit der Ontologie 1.3. Als Bookmarks wurden Homepages bekannter Wissenschaftler aus entsprechenden Forschungsbereichen manuell ausgewählt. Die Blattknoten der Ontologie wurden initialisiert mit 9-15 Bookmarks pro Blatt; die gesamte Trainingsbasis enthielt 81 Dokumente.

Der anschließende fokussierte Crawl fand, ausgehend von dieser Startmenge, 2738 positiv klassifizierte Seiten auf Link-Entfernungen zwischen 1 und 7. Der Crawl wurde nach 6 Stunden angehalten; insgesamt wurden dabei ca. 24750 URLs auf 7149 unterschiedlichen Hosts besucht.

Für die Bewertung der Crawling-Iterationen wurde - durch manuelle Auswertung der Ergebnismenge - die Präzision des Klassifikators (Anteil der richtigen positiven Klassifikationsentscheidungen pro Knoten) berechnet. Weitere interessante Kenngrößen sind die Ausbeute (Anzahl der Crawl-Resultate pro Knoten), die Zahl der Archetypen und deren "Qualität" (Anteil der korrekt ermittelten themenspezifischen Dokumente unter den Archetypen).

Die Auswertung erfolgte für folgende Klassifikationsstrategien:

1. Einfacher Term-basierter SVM Klassifikator ohne Selektion der Features [AllTerms]
2. Term-basierter SVM Klassifikator mit Selektion der Features durch MI (300 klassenweise beste Features [SelTerms(300)];
3. Term-basierter SVM Klassifikator mit Selektion der Features durch MI mit ”Kontrasttermen” (300 beste klassenspezifische Features plus je 100 beste Features aus jeder Gegnerklasse)[ContrTerms(300,100)];
4. Paar-basierter SVM Klassifikator (Sliding-Window der Größe $SW = 16$) mit Pre-Selektion der Terme durch MI (300 beste Terme für Paarbildung) [Pairs(pre:300,SW:16)];
5. Paar-basierter SVM Klassifikator ($SW = 16$) mit Pre-Selektion der Terme durch MI und Verwendung der ”Kontrastfeatures” (300 Klassenspezifische Termen plus je 100 pro Gegnerklasse) [Pairs(contr:(300,100),SW:16)];
6. Kombinierter SVM Klassifikator auf Termen und Paaren ($SW = 16$) mit Selektion der Features durch MI (Vorauswahl 300 beste Terme für Paarbildung) [Pairs&Terms(pre:300,SW:16)];
7. Kombinierter SVM Klassifikator auf Termen und Paaren ($SW = 16$) mit Selektion der Kontrastfeatures und Vorauswahl durch MI (300 beste klassenspezifische Terme plus je 100 Kontrastterme pro Gegnerklasse für Paarbildung) [Pairs&Terms(contr:(300,100),SW:16)];

Außerdem bewerteten wir für diese Gruppe der Klassifikatoren die Ergebnisse der Meta-Strategien ”Mehrheitsentscheid” und ”Gewichteter Mehrheitsentscheid” nach dem $\xi\alpha$-Gewichtungsschema. Aufgrund des hohen Berechnungsaufwandes, verzichteten wir in unseren ersten Experimenten auf die Auswertung der Strategien auf der Basis der Dokumentvereinigung.

In einer weiteren Experimentserie haben wir die restriktive Meta-Strategie ”Einstimmigkeit” betrachtet.

### 4.2 Ergebnisse

Die Tabelle 1 zeigt eine Zusammenfassung der Auswertungen für verschiedene Klassifikationstechniken und die Zusammenfassung der Archetyp-Selektion in der ersten Crawling-Iteration für die - mit 10 Bookmarks initialisierte - Klasse *”Root/Semistructured Data/Data Mining”*. Angegeben sind außerdem die Schätzung der Präzision durch den $\xi\alpha$-Estimator und der tatsächliche Wert nach der manuellen Auswertung.

| **Verfahren** | **Klassifiziert** | **Korrekt** | $\xi\alpha$ | **Präzision** | **Archetypen** | **Korrekt** | **Präzision** |
|---|---|---|---|---|---|---|---|
| AllTerms | 229 | 181 | 0.71 | 0.79 | 32 | 27 | 0.84 |
| SelTerms(300) | 197 | 162 | 0.78 | 0.82 | 29 | 25 | 0.86 |
| ContrTerms(300,100) | 184 | 158 | 0.77 | 0.85 | 31 | 24 | 0.77 |
| Pairs(pre:300,SW:16) | 208 | 152 | 0.74 | 0.73 | 27 | 23 | 0.85 |
| Pairs (contr:(300,100),SW:16) | 169 | 149 | 0.81 | 0.88 | 33 | 28 | 0.84 |
| Pairs&Terms (pre:300,SW:16) | 155 | 141 | 0.84 | 0.91 | 42 | 32 | 0.76 |
| Pairs&Terms (contr(300,100), SW16) | 142 | 122 | 0.85 | 0.86 | 35 | 31 | 0.88 |
| Majority | 98 | 92 | - | 0.94 | 15 | 14 | 0.93 |
| Mutual | 95 | 91 | - | 0.96 | 19 | 17 | 0.89 |

Tabelle 1: Präzision der Klassifikationsverfahren und der Archetyp-Selektion

Die Ergebnisse zeigen die Vorteile der Meta-Klassifikationsstrategien. Die parallele Betrachtung verschiedener Dokumenteigenschaften ermöglicht gute Präzision und Qualität der erkannten Archetypen auch bei mäßiger Präzisionserwartung einzelner Klassifikationsmethoden.

Natürlich darf man nicht vergessen, dass höhere Präzision der Metaverfahren mit einem deutlich höheren rechnerischen Aufwand verbunden ist und somit primär in besonders präzisions-sensitiven Anwendungen benutzt werden sollte, z.B. beim Re-Training des fokussierten Crawlers oder für die Revision der besten Suchergebnisse. In diesen Fällen kann die Menge der zu validierenden Dokumente entsprechend begrenzt werden.

Tabelle 2 zeigt die Ergebnisse des Meta-Verfahrens "Einstimmigkeit". Wir haben hier in drei Experimenten unterschiedliche Kombinationen von Einzelverfahren über einer allgemeineren und "flachen" Hierarchie getestet, die aus den Kategorien "*Business*", "*Computers and Internet*", "*Sports*" und "*Health*" bestand. Wir konnten, bei relativ moderater Reduktion der Dokumentmenge, eine deutliche Erhöhung der Präzision gegenüber dem besten Einzelverfahren erzielen.

| Experiment | Klassifiziert | einh. Klassifiziert | Prec bestes Einzelverf. | Prec Einstimmigkeit |
|---|---|---|---|---|
| 1 | 242 | 183 | 0.95 | 0.99 |
| 2 | 1012 | 668 | 0.87 | 0.96 |
| 3 | 757 | 694 | 0.91 | 0.97 |

Tabelle 2: Präzisionsverbesserung bei dem restriktiven Meta-Verfahrens "Einstimmigkeit" mit Dokumentreduktion gegenüber dem jeweils besten Einzelverfahren

Eine interessante Beobachtung hinsichtlich der vorgestellten Metastrategien bestand in der effektiven "Erkennung" von Dokumenten, die keiner der Klassen der Taxonomie zugeordnet werden konnten. Nur 166 von 614 nicht eindeutigen oder falschen Dokumente (ca. 27 %) wurden z.B. in Experiment 2 durch das Metaverfahren einer der Klassen zugeordnet. Somit sind Metaverfahren - insbesondere in Kombination mit anderen Techniken - für die "Grobfilterung"der irrelevanten Dokumente gut geeignet.

## 5 Fazit und Ausblick

**Fazit** In diesem Papier haben wir unterschiedliche Methoden zur Konstruktion von Featureräumen für vektorbasierte maschinelle Lernverfahren (am Beispiel von SVM) zur Klassifikation von Webdokumenten kennen gelernt. Jeder dieser Ansätze untersucht und kombiniert unterschiedliche Aspekte der betrachteten Dokumente wie z.B. Anordnung der Terme im Dokument oder Dokumentumgebung. Unsere Experimente zeigen, dass die meisten dieser Verfahren schon für sich gesehen Sinn machen.

Durch Anwendung von Metaverfahren konnten wir noch eine deutliche Erhöhung der Klassifikationspräzision erzielen. Eine Verbesserung der Präzision haben wir insbesondere durch eine restriktive Variante von Metaverfahren erreicht, bei der wir über einen Teil der Dokumente keine Aussage treffen, bei den übrigen Dokumenten aber bessere Ergebnisse erzielen. Interessant ist dies u.a. für ein unüberwachtes Lernsystem, bei dem die ursprüngliche Trainingsdatenmenge automatisch um neu klassifizierte Dokumente ergänzt wird. Bei der Fülle von Inhalten im WWW können wir nicht davon ausgehen, dass unsere Trainingsdaten alle Themenbereiche erfassen (Häufig sind wir auch nur an speziellen Taxonomien interessiert). Diese Themenbereiche lassen sich auch durch Vorschalten von Hilfshierarchien und Hinzunahme von Sonderklassen für sonstige Inhalte (was im Übrigen wieder mit teurem intellektuellen Aufwand verbunden ist) nicht völlig abdecken. Ein automatischer Crawler wird jedoch mit hoher Wahrscheinlichkeit mit solchen Dokumenten umgehen müssen. Dies unterscheidet die hier gegebene Aufgabenstellung von klassischen

Experimenten zu Textklassifikation über Dokumenten, die alle zu wohl definierten Trainingskategorien gehören (z.B. bei den typischen Reuter-Benchmarks). Experimente zeigen, dass Metaverfahren einen ersten Ansatz bieten, mit dieser Problematik umzugehen.

**Ausblick** Wir betrachten das vorgestellte Framework der Klassifikationsstrategien als eine Basis für weiterführende Untersuchungen und Experimentreihen. Zu den wichtigen, bisher unzureichend untersuchten, Aspekten dieser Arbeit gehört die Fehleranfälligkeit des Verfahrens (Auswirkung von falschen Trainingsbeispielen, z.B. inkorrekten Archetypen, auf das Gesamtergebnis des fokussierten Crawls). Auch die Frage des "optimalen" Re-Trainings (Zeitpunkt der Auslösung, Anzahl der verwendeten Archetypen und Features) soll näher untersucht werden. Eine besondere Aufmerksamkeit ist dabei der Erzeugung von Feature-Mengen (Typ, Anzahl der Features verschiedener Arten) zu schenken, um den unnötigen Aufwand in leicht separierbaren Fällen zu vermeiden. Zum Beispiel, die aufwendigeren Strategien (Termpaare, Dokumentvereinigung etc.) können einbezogen werden, falls die geschätzte Präzision des Term-basierten Klassifikators nicht ausreichend ist. Zu unseren weiteren Zielen gehört die Entwicklung eines formalen Modells der vorgestellten Metaverfahren, um deren Funktionsweise besser zu verstehen sowie allgemeine Leistungsgrenzen genauer abschätzen zu können. Ferner soll das vorgestellte Framework in Zukunft durch Clustering-Algorithmen für nicht eindeutig klassifizierte Dokumente erweitert werden, um unvollständige benutzerdefinierte Ontologien durch automatisch generierte Gruppen von thematisch verwandten Dokumenten dynamisch erweitern zu können.

## Literatur


[Bur98] C.J.C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2), 1998.

[CD00] H. Chen and S. Dumais. Bringing Order to the Web: Automatically Categorizing Search Results. *ACM CHI Conference on Human Factors in Computing Systems*, 2000.

[CJT01] S. Chakrabarti, M.M. Joshi, and V.B. Tawde. Enhanced Topic Distillation using Text, Markup Tags, and Hyperlinks. *ACM SIGIR Conference*, 2001.

[DC00] S. Dumais and H. Chen. Hierarchical Classification of Web Content. *ACM SIGIR Conference*, 2000.

[JCST01] T. Joachims, N. Cristianini, and J. Shawe-Taylor. Composite Kernels for Hypertext Categorisation. *Proceedings of the International Conference on Machine Learning (ICML)*, 2001.

[Joa98] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *European Conference on Machine Learning (ECML)*, 1998.

[Joa00] T. Joachims. Estimating the generalization performance of an SVM efficiently. *European Conference on Machine Learning (ECML)*, 2000.

[MS99] C.D. Manning and H. Schuetze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[Pora] M. Porter. An algorithm for suffix stripping. *Automated Library and Information Systems*, 14(3).

[Porb] M. Porter. The Porter Stemming Algorithm. *http://www.tartarus.org/ martin/PorterStemmer/index.html*.

[SSTW02] S. Sizov, S. Siersdorfer, M. Theobald, and G. Weikum. The BINGO! focused crawler: From Bookmarks to Archetypes. *IEEE Computer Society International Conference on Data Engineering (ICDE), San Jose, California*, 2002.

[Vap98] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[YP97] Y. Yang and O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. *International Conference on Machine Learning (ICML)*, 1997.