



Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät II
Institut für Informatik
Lehrstuhl für Wissensmanagement

**„Automatische Zitationsextraktion aus wissenschaftlichen
Artikeln“**

Studienarbeit

**Eingereicht von:
Gunar Maiwald
Matr.-Nr. 141370**

Betreuung durch:

Prof. Dr. Tobias Scheffer, Institut für Informatik an der Humboldt-Universität Berlin

Ulf Brefeld, Institut für Informatik an der Humboldt-Universität Berlin

Stefan Lohrum, Kooperativer Bibliothekenverbund Berlin Brandenburg

Berlin, den 5. September 2006

Inhaltsverzeichnis

1. EINLEITUNG	3
2. VERWANDTE ARBEITEN	4
3. GRUNDLAGEN UND METHODEN	6
3.1 Problemstellung	6
3.2 Support Vector Machines	8
3.3 Hidden Markov Modelle	10
3.4 Conditional Random Fields	14
4. EXPERIMENTE UND ERGEBNISSE	16
4.1 Datenaufbereitung	16
4.2 Durchführung	19
4.3 Ergebnisse	22
4.3.1 Allgemeine Performance	22
4.3.2 Performance auf ungesehenen Zitationsstilen	23
4.3.3 Performance bei reduzierter Labelmenge	24
4.3.4 Performancevergleich der Verfahren	25
5. ZUSAMMENFASSUNG	26
6. LITERATURANGABEN	27

1. Einleitung

Dem Kooperativen Bibliothekenverbund Berlin Brandenburg (KOBV) stehen mit dem Volltext Dokumenten Server (VDS) wissenschaftliche Artikel der Verlage Kluwer, Springer und Elsevier zur Verfügung. Es handelt sich dabei um Artikel verschiedener Zeitschriften, die zwischen den Jahren 1997 und 2003 publiziert wurden. Die Anzahl aller Artikel, die in Volltextform zur Verfügung stehen, beträgt insgesamt über 1 Million. Jeder wissenschaftliche Artikel kann von anderen Artikeln anhand seiner Metadaten (Autor, Titel, Zeitschrift, Jahr etc.) referenziert werden. Ein Verweis einer Publikation auf eine andere wird als Zitation bezeichnet. Ausgehend von den Zitationen einer Artikelmenge lässt sich ein Zitationsgraph aufbauen, in dem die Publikationen die Knoten bilden zwischen denen die Zitationen die Kanten aufspannen. Anhand eines solchen Graphen lassen sich unter anderem *Autorencommunities* mittels Netzwerkanalyse ausfindig machen sowie *Artikelranking* auf der Basis von Zitationshäufigkeiten erstellen.

Diese Arbeit setzt sich mit Zitationsextraktion auseinander, deren Ziel es ist, die zur Referenzierung eines Artikels angegebenen Metadaten korrekt zu erkennen und zu extrahieren. Das Hauptproblem der Zitationsextraktion ist, dass die Art und Weise, wie Artikel referenziert werden, generell jeder Zeitschrift freigestellt ist. Somit kommt es vor, dass ein und derselbe Artikel unterschiedlich referenziert wird (vgl. *Abbildung 1.1*).

Davenport, T., DeLong, D., and Beers, M. "Successful knowledge management projects," Sloan Management Review (39:2)

Davenport, T., DeLong, D., & Beers, M. (1998) Successful knowledge management projects. Sloan Management Review, 39(2), 43-57.

1. Davenport, T., DeLong, D. and Beers, M. 1998. Successful knowledge management projects. Sloan Management Review, 39 (2). 43-57.

[1] T. Davenport, D. DeLong, and M. Beers, "Successful knowledge management projects," Sloan Management Review, vol. 39, no. 2, pp. 43-57, 1998.

Abbildung 1.1: Verschiedene Referenzen auf denselben Artikel

Dies bedeutet prinzipiell, dass die Anzahl möglicher Zitierweisen sehr groß ist. Anstatt ein Verfahren zu verwenden, das sich auf eine festgeschriebene Anzahl von Zitationsstilen beschränkt, ist es sinnvoll, eine Methode einzusetzen, welche die Metadaten unabhängig von der verwendeten Zitierweise extrahiert.

Eine Möglichkeit hierfür bildet die Textannotation, deren Ziel es ist, Wörter in Texten automatisch mit korrekten semantischen Labels zu versehen. In dieser Arbeit sollen 3 verschiedene Methoden vorgestellt werden. Darüberhinaus wird die Möglichkeit geprüft, inwieweit diese innerhalb der Suchmaschine des VDS eingesetzt werden können.

Der Rest der Arbeit gliedert sich wie folgt. Im Anschluss an die „Einleitung“ werden im 2. Kapitel mehrere Ansätze vorgestellt, die mit dem Thema der Zitationsextraktion verwandt sind. Abschnitt 3 erläutert 3 Verfahren, die zur Textannotation verwendet werden. Diese werden durch vergleichende Experimente im 4. Kapitel evaluiert. Eine Zusammenfassung mit Ausblick erfolgt in Abschnitt 5.

2. Verwandte Arbeiten

Einen Überblick über verschiedene Methoden, welche Zitationen analysieren und verarbeiten liefert [Law99]. Er unterscheidet dabei 4 Klassen. (1) Hierbei handelt es sich um Methoden, die den Editabstand als Maß verwenden, um Unterschiede zwischen Zeichenketten zu quantifizieren. Als Beispiel wird der *LikeIt*-Algorithmus von [Yia97] angeführt. Eine Erweiterung des *Likelt*-Algorithmus bildet die Grundlage für ein Verfahren, das in der Artikeldatenbank *CiteSeer*¹ eingesetzt wird. Darin sind Publikationen aus dem Bereich der Informatik und Informationstechnologie frei recherchierbar. Zu jedem Artikel sind Quellen angegeben, die diesen referenzieren. (2) Worthäufigkeiten und Wortvorkommen werden mittels TF-IDF gewichtet und sind ein wichtiger Bestandteil des Information Retrievals geworden [Man99]. (3) Regelbasierte Verfahren analysieren die Struktur von Zitationen bzw. die einzelner Felder. Ziel ist es, reguläre Ausdrücke für eine oder mehrere Zitierweisen zu generieren, mit deren Hilfe Metadaten extrahiert werden können. Einen solcher Ansatz wird in [Din99] beschrieben. Zwar werden darin die Daten zu über 90% richtig extrahiert, jedoch beschränken sich die Versuche auf 12 Zeitschriften. Deswegen ist dieses Verfahren nur bedingt einsetzbar,

¹ <http://citeseer.ist.psu.edu/>

Metadaten aus neuen und bisher ungesesehenen Zitationsstilen zu extrahieren. (4) Probabilistische Verfahren können mittels bibliographischer Informationen trainiert werden. Zu diesen zählen unter anderem Hidden Markov Modelle (HMM) und Conditional Random Fields (CRF). [Sey99] und [Cal00] beschreiben HMMs, zur Informationsextraktion aus wissenschaftlichen Artikeln, welche Metadaten jeweils zu über 90% korrekt extrahieren. CRFs, die ebenfalls in der Informationsextraktion eingesetzt werden können, werden in [Wel04] beschrieben. Experimente haben gezeigt, dass auch dieses Verfahren, die Daten mit über 90% korrekt extrahiert. Sowohl HMMs als auch CRFs werden auch als sequentielle, probabilistische Klassifikationsverfahren bezeichnet, die zur Textannotation verwendet werden können. *Support Vector Machines* (SVM), ein weiteres Klassifikationsverfahren, wird in [Alt03] vorgestellt. Im Gegensatz zu den anderen beiden Ansätzen handelt es sich dabei um kein sequentielles, probabilistisches sondern um ein vektorenbasiertes Verfahren. Part-of-speech-tagging wird zu 88% und Named-Entity-Recognition zu über 90% korrekt erledigt. [Laf01] und [Pen04] vergleichen HMM und CRF bzw. HMM, CRF und SVM und kommen jeweils zu der Schlussfolgerung, dass unter identischen Umständen CRFs korrekter als die beiden anderen Verfahren arbeiten. Den drei genannten Klassifikationsverfahren ist gemein, dass sie sich im Gegensatz zu anderen Ansätzen nicht auf eine feste Anzahl von Zitationsstilen beschränken. Deswegen sind sie besonders für das eingangs beschriebene Problem der Textannotation geeignet und werden in dieser Arbeit vorgestellt und experimentell evaluiert.

Ein ontologiebasiertes Verfahren, welches auf einer Begriffshierarchie basiert, wird in [Day05] beschrieben. Das zentrale Element bildet eine Wissensbasis, die für Texte aus dem Bereich der Bioinformatik und 6 verschiedene Zitationsstilen ausgerichtet wurde. Bei diesen Zitierweisen wurden im Durchschnitt über 97% Korrektheit erreicht, bei 30 weiteren Zitierweisen im Durchschnitt über 87% Korrektheit ermittelt. Allerdings verlangt der Aufbau einer Ontologie einen extrem hohen Zeitaufwand, welcher den Rahmen dieser Arbeit sprengen würde. Darüberhinaus beschränkt sich dieser Ansatz prinzipiell ebenfalls auf eine feste Anzahl von Zitationsstilen.

3. Grundlagen und Methoden

Der folgende Teil der Arbeit erklärt die untersuchten Verfahren und deren Grundlagen. In 3.1 wird die eine allgemeine, für alle Verfahren gültige Problemstellung formuliert und es werden wichtige Grundbegriffe eingeführt. 3.2. beschreibt die Arbeitsweise von Support Vector Machines, in 3.3. werden Hidden Markov Modelle erläutert und 3.4. geht näher auf Conditional Random Fields ein.

3.1 Problemstellung

Die Textannotation kann als Optimierungsproblem interpretiert werden: Gegeben sei eine Menge von n Trainingsbeispielen $T = \{(X^i, Y^i)\}$ mit $i = 1 \dots n$. $X^i = x_1^i, \dots, x_{L_i}^i$, mit $x_j^i \in \Omega$ ist dabei eine Sequenz von L_i Token und $Y^i = y_1^i, \dots, y_{L_i}^i$, mit $y_j^i \in \Sigma$, eine korrespondierende Sequenzen von Klassenlabeln. Dabei ist Σ eine endliche Menge von Klassenlabeln. Gesucht ist eine Funktion h und deren Parameter λ , die zu einer neuen Sequenz von Token X die korrespondierende Sequenz von Klassenlabeln vorhersagt [Die02]:

$$Y^* = \arg \max_Y h_\lambda(X, Y) \quad (1)$$

Die Klassifikation durch h erfolgt dabei nicht direkt auf der Beobachtungssequenz X , sondern auf einer gleichlangen Sequenz von binären Featurevektoren. Ein binäres Feature kann man sich als ein charakteristisches Merkmal vorstellen, welches entweder vorhanden ist oder nicht.

Dabei werden 2 Arten von Features unterschieden: *Observationsfeatures*

$$f^\sigma(Y, X|t) = \mathbb{I}[y_t = \sigma] \cdot \vec{\Psi}(x_{t \pm s}) \quad (2)$$

mit $s = 0, 1, 2, 3, \dots$ beschreiben die Beobachtung eines Labels und eines Tokens, wobei $\mathbb{I}[cond] = 1$, wenn $cond$ erfüllt und 0 sonst. Sämtliche Observationsfeatures eines Tokens werden dabei durch den Featurevektor $\vec{\Psi}$ der Dimension l beschrieben:

$$\vec{\Psi}(x_t) = (\Psi_1(x_t), \dots, \Psi_l(x_t))^T. \quad (3)$$

$\Psi_j(x_t)$ bezeichnet eine charakteristische Eigenschaft eines Tokens, beispielsweise $\Psi_{234}(x_t) = \mathbb{I}[x_t = \text{"Mueller"}]$.

Transitionsfeatures

$$g^{\sigma,\tau}(Y|t) = \llbracket y_{t-1} = \sigma \wedge y_t = \tau \rrbracket, \sigma, \tau \in \Sigma \quad (4)$$

beschreiben die Beobachtung aufeinander folgender Labels.

Der Featurevektor $\vec{\Phi}$ besteht aus $l+m$ vielen charakteristischen Merkmalen, wobei $m = |\Sigma|^2$:

$$\vec{\Phi}(Y, X|t) = (\dots, f^\sigma(Y, X|t), \dots, g^{\sigma,\tau}(Y|t), \dots)^T, \sigma, \tau \in \Sigma. \quad (5)$$

Das Optimierungsproblem hat die allgemeine Form: Minimiere/maximiere die Zielfunktion $Z(X^i, Y^i, \lambda)$ für $i = 1 \dots n$ bzgl. λ unter Einhaltung aller Nebenbedingungen. Die konkrete Zielfunktion, die Anzahl der Nebenbedingungen und das Verfahren zur Lösung des Optimierungsproblems hängt von der jeweiligen Funktion h ab.

Mit den Modellparametern λ einer Funktion h kann für eine Sequenz X die optimale Labelsequenz Y^* ermittelt werden. Bei nichtsequentiellen Ansätzen wie der SVM wird hierbei für *jede Position* l einzeln das optimale Label y_l^* ermittelt und die Labelsequenz Y^* entsteht durch Konkatination der einzelnen Label:

$$y_l^* = \arg \max_{y_l} h(x, y, \lambda). \quad (6)$$

Bei sequentiellen Ansätzen wird hingegen die *optimale Folge* von Labeln ermittelt:

$$Y^* = \arg \max_Y h(X, Y, \lambda). \quad (7)$$

		Realität	
		+	-
Vorhersage	+	<i>True Positive</i> (TP_σ)	<i>False Positive</i> (FP_σ)
	-	<i>True Negative</i> (TN_σ)	<i>False Negative</i> (FN_σ)

Abbildung 3.1: TP_σ , FP_σ , TN_σ und FN_σ bilden die Grundbausteine der Evaluierung einer Klasse $\sigma \in \Sigma$.

In dieser Arbeit werden *Micro-F1* und *Macro-F1* für die Evaluierung verwendet:

$$Micro - F1 = \frac{2 \cdot \frac{\sum_{\sigma} TP_{\sigma}}{\sum_{\sigma} (TP_{\sigma} + FP_{\sigma})} \cdot \frac{\sum_{\sigma} TP_{\sigma}}{\sum_{\sigma} (TP_{\sigma} + FN_{\sigma})}}{\frac{\sum_{\sigma} TP_{\sigma}}{\sum_{\sigma} (TP_{\sigma} + FP_{\sigma})} + \frac{\sum_{\sigma} TP_{\sigma}}{\sum_{\sigma} (TP_{\sigma} + FN_{\sigma})}}, \quad (8)$$

$$Macro - F1 = \frac{2|\Sigma|^{-2} \cdot \sum_{\sigma} \frac{TP_{\sigma}}{TP_{\sigma} + FP_{\sigma}} \cdot \sum_{\sigma} \frac{TP_{\sigma}}{TP_{\sigma} + FN_{\sigma}}}{|\Sigma|^{-1} \sum_{\sigma} \frac{TP_{\sigma}}{TP_{\sigma} + FP_{\sigma}} + |\Sigma|^{-1} \sum_{\sigma} \frac{TP_{\sigma}}{TP_{\sigma} + FN_{\sigma}}}. \quad (9)$$

Bei Micro-F1 sind alle Token gleich gewichtet und es wird die durchschnittliche Token-genauigkeit ermittelt. Diese ist von großen Klassen dominiert. Macro-F1 gewichtet alle Klassen gleich und ermittelt die durchschnittliche Klassengenauigkeit, welche von kleinen Klassen dominiert ist.

3.2 Support Vector Machines

Support Vector Machines (SVM) wurden erstmals in [Vap74] vorgestellt. Es handelt sich dabei um einen nichtsequentiellen, vektorbasierten Ansatz. Der (binäre) Klassifikator ordnet ein Token genau einer von zwei gegebenen Klassen (z.B. Positiv / Negativ) zu.

Für eine Vielzahl von Klassifikationsprobleme ist eine lineare Trennung der Token nicht möglich. Aus diesem Grund werden diese vor der Klassifikation aus dem Eingaberaum mittels Ψ in einen höherdimensionalen Merkmalsraum abgebildet. In diesem höherdimensionalen Raum existiert eine Hyperebene, anhand derer das Token x einer der beiden Klassen zugeordnet werden kann.

Diese Hyperebene ist bestimmt durch den Normalenvektor w und den Bias b . Der Abstand von $\Psi(x)$ zur Trennebene beträgt γ . Sei $\hat{\gamma}$ der kleinste Abstand aller x der Trainingsmenge $T = \{(X^i, Y^i)\}$ mit $i = 1..n$. Dann gilt für alle x eines linear trennbaren Klassifikationsproblems:

$$y(\langle w, \Psi(x) \rangle + b) \geq \hat{\gamma}, \text{ mit } y \in \{\pm 1\}. \quad (10)$$

Das in 3.1 angesprochene Optimierungsproblem besteht darin, den kleinsten Abstand beider Klassen zur Hypertrennebene zu maximieren. Dieses wird auch als *Margin-Maximierung* bezeichnet. Die Zielfunktion lautet dann: Maximiere $\gamma = \frac{\hat{\gamma}}{\|w\|}$.

Daraus resultiert das duale Optimierungsproblem für $\hat{\gamma} = 1$: Minimiere bzgl. w, b :

$$Z(w, b) = \|w\|^2, \text{ so dass } y(\langle w, \Psi(x) \rangle + b) \geq 1. \quad (11)$$

Allerdings sind die Trainingsdaten in der Regel nicht streng linear separierbar. In diesem Fall wird das Optimierungsproblem durch einen *Schlupfvektor* ξ der für die Integrität „unsauberer“ Trainingsdaten sorgt. Das duale Optimierungsproblem lautet nun wie folgt: Minimiere bzgl. w, b :

$$Z(w, b) = \|w\|^2 + C \sum_i \sum_{j=1..L_i} \xi_j^i, \quad (12)$$

$$\text{so dass } y_j^i (\langle w, \Psi(x_j^i) \rangle + b) \geq \hat{\gamma} - \xi_j^i \text{ und } \xi_j^i \geq 0, \forall i = 1..n, \forall j = 1..L_i,$$

wobei es sich bei C um eine positive Gewichtungskonstante für den Schlupfvektor handelt. Für das optimale w gilt folgende Relation:

$$w = \sum_i a_i y_i \bar{\Psi}(x_i). \quad (13)$$

Mit Hilfe von Lagrange-Multiplikatoren können die Nebenbedingungen in die Zielfunktion eingesetzt werden: Das daraus resultierende duale Optimierungsproblem hat die Form: Maximiere bzgl. a :

$$Z(X^i, Y^i, a) = \sum_i a_i - \frac{1}{2} \sum_i \sum_j y_i y_j a_i a_j \langle \Psi(x_i), \Psi(x_j) \rangle, \quad (14)$$

$$\text{so dass } \forall i: 0 \leq a_i \leq C \text{ und } \sum_i a_i y_i = 0.$$

Die Klassifikation erfolgt hierbei Merkmalsraum, was bei einer hohen Dimension zu einer sehr komplexen Berechnung führt. Einen Ausweg aus diesem Problem liefert ein so

genannter Kernel, eine Funktion im Eingaberaum, die sich wie ein Skalarprodukt im Merkmalsraum verhält. Statt $\langle \Psi(x_i), \Psi(x_j) \rangle$ genügt es $\langle x_i, x_j \rangle$ zu berechnen. Dies bedeutet, dass die Dimension des Merkmalsraumes bei der Berechnung der optimalen Hyperebene keine Rolle spielt.

Der binäre Klassifikator für das Token x lautet dann:

$$f(x) = \text{sgn}(\langle w, \Psi(x) \rangle + b). \quad (15)$$

Da es sich bei der Textannotation um ein mehrwertiges Klassifikationsproblem handelt, muss für jedes Klassenlabel σ eine eigene SVM verwendet werden. Diese werden nach dem „Einer-gegen-Alle-Prinzip“ trainiert. Das optimale Label y^* für ein Token x ergibt sich dann aus dem maximalen Funktionswert aller SVM:

$$y^* = \arg \max_{\sigma} f_{\sigma}(x) = \arg \max_{\sigma} (\langle w_{\sigma}, \Psi(x) \rangle + b_{\sigma}), \forall \sigma \in \Sigma. \quad (16)$$

Die optimale Labelsequenz Y^* für eine Beobachtungssequenz X ergibt sich dann aus der Konkatenation der optimalen Label:

$$Y^* = y_1^*, y_2^*, \dots, y_L^*. \quad (17)$$

3.3 Hidden Markov Modelle

Das Problem der Textannotation kann auch mit Hilfe sequentieller, probabilistischer Verfahren gelöst werden. Zu ihnen zählen die Hidden Markov Modelle (HMM). Im Gegensatz zur SVM, welche für jedes Token x das optimale Label y generiert, liefert ein HMM die optimale Labelfolge Y^* zur *gesamten* Beobachtungssequenz X .

Ein HMM λ ist ein Tripel (A, B, π) . A ist die Matrix der *Transitionswahrscheinlichkeiten*, B die Matrix der *Emissionswahrscheinlichkeiten* und π der Vektor der *Startwahrscheinlichkeiten*. man kann sich ein HMM kann als ein Modell zweier hierarchischer, diskreter Prozesse vorstellen. Dabei handelt es sich um einen sichtbaren und einen unsichtbaren Prozess (vgl. *Abbildung 3.2*).

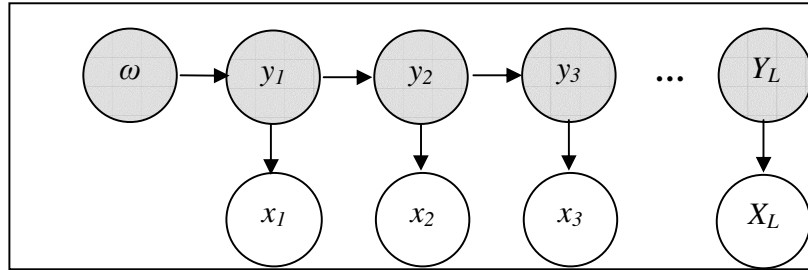


Abbildung 3.2: Der sichtbare Prozess (unten), welcher die Beobachtungssequenz X generiert hängt direkt vom unsichtbaren Prozess (oben) ab. Für jeden Zeitpunkt t gilt, dass y_t allein von y_{t-1} und x_t nur von y_t abhängig ist. $\omega \rightarrow y_1$ kennzeichnet den Start.

Zu jedem Zeitpunkt t befindet sich das HMM in einem von außen nicht sichtbaren Zustand. Der Wechsel von einem Zustand y_{t-1} in einen Zustand y_t wird als Transition bezeichnet. Ihm liegt die so genannte *Transitionswahrscheinlichkeit* zugrunde:

$$a_{\sigma\tau} = P(y_t = \tau | y_{t-1} = \sigma), \quad \sigma, \tau \in \Sigma. \quad (18)$$

Die Abfolge der Zustände wird auch als *Markov-Kette* bezeichnet, da jeder Zustand nur von seinem unmittelbaren Vorgänger abhängig ist. Die Matrix A enthält die Transitionswahrscheinlichkeiten $a_{\sigma\tau}, \forall \sigma, \tau \in \Sigma$. Die Variable π_σ gibt die *Startwahrscheinlichkeit* an:

$$\pi_\sigma = P(y_1 = \sigma), \quad \sigma \in \Sigma. \quad (19)$$

Zu jedem Zeitpunkt t wird ein Token x_t generiert. Dies geschieht in Abhängigkeit vom Zustand y_t . Der Generierung eines Token liegt die so genannte *Emissionswahrscheinlichkeit* zugrunde:

$$b_\sigma(o) = P(x_t = o | y_t = \sigma), \quad \sigma \in \Sigma, o \in \Omega. \quad (20)$$

Die Matrix B enthält die Emissionswahrscheinlichkeiten $b_\sigma(o), \forall \sigma \in \Sigma, \forall o \in \Omega$.

In *Abbildung 3.4* ist die schematische Darstellung eines HMM mit 3 Zuständen und den Token #,\$,% zu sehen.

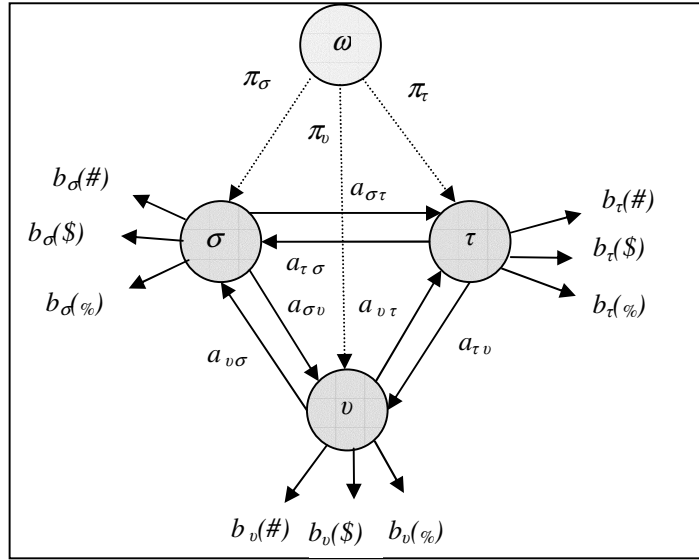


Abbildung 3.4: Schema eines HMMs mit den Zuständen σ , τ , ν und dem Ausgabealphabet $\#, \$, \%$. Pfeile und Beschriftungen stehen für Transitionen, Emissionen, Startzustände und deren Wahrscheinlichkeiten.

Das in 3.1 angesprochene Optimierungsproblem für ein HMM besteht darin, die Matrizen A, B und den Vektor π so zu bestimmen, dass die Summe der Wahrscheinlichkeiten für die Trainingsmenge maximal wird: Maximiere bzgl. λ :

$$Z(X^i, Y^i, \lambda) = \sum_i P(X^i, Y^i, \lambda), \text{ für } i = 1 \dots n \text{ so dass gilt:} \quad (21)$$

$$\forall \sigma, \tau \in \Sigma: \sum_{\tau} a_{\sigma\tau} = 1, \forall \sigma \in \Sigma, \forall o \in \Omega: \sum_o b_{\sigma}(o) = 1 \text{ und } \forall \sigma \in \Sigma: \sum_{\sigma} \pi_{\sigma} = 1. \quad (22)$$

Eine einfache Möglichkeit, die Parameter λ für ein HMM zu schätzen, bietet die *Maximum-Likelihood-Methode*:

$$a_{\sigma\tau} = \frac{A_{\sigma\tau}}{\sum_{\tau'} A_{\sigma\tau'}}, b_{\sigma}(o) = \frac{B_{\sigma}(o)}{\sum_{o'} B_{\sigma}(o')} \text{ und } \pi_{\sigma} = \frac{A_{\omega\sigma}}{\sum_{\sigma'} A_{\omega\sigma'}} \quad (23)$$

$$\sigma, \sigma', \tau, \tau' \in \Sigma, o, o' \in \Omega,$$

wobei $A_{\sigma\tau}$ die Anzahl der beobachteten Übergangshäufigkeiten von σ nach τ in der Trainingsmenge bezeichnet und $B_{\sigma}(o)$ die Anzahl der beobachteten Emissionshäufigkeiten von o im Zustand σ beschreibt. Um Nullwahrscheinlichkeiten

zu verhindern, können *Pseudocounts* c zu den beobachteten Häufigkeiten hinzuaddiert werden:

$$a_{\sigma\tau} = \frac{A_{\sigma\tau} + c}{\sum_{\tau'} A_{\sigma\tau'} + |\Sigma|}, b_{\sigma}(o) = \frac{B_{\sigma}(o) + c}{\sum_{o'} B_{\sigma}(o') + |\Sigma|} \text{ und } \pi_{\sigma} = \frac{A_{\alpha\sigma} + c}{\sum_{\sigma'} A_{\alpha\sigma'} + |\Sigma|} \quad (24)$$

$$\sigma, \sigma', \tau, \tau' \in \Sigma, o, o' \in \Omega .$$

Bei einer Laplace-Glättung gilt: $c = 1$.

Die Wahrscheinlichkeit für eine Labelsequenz $Y = y_1, y_2, \dots, y_L$ und eine Beobachtungsfolge $X = x_1, x_2, \dots, x_L$, gegeben λ lautet wie folgt:

$$P(X, Y | \lambda) = \pi_{y_1} \cdot b_{y_1}(x_1) \cdot \prod_{t=2}^L a_{y_{t-1}y_t} \cdot b_{y_t}(x_t) \quad (25)$$

Die wahrscheinlichste Labelsequenz Y^* ergibt sich aus der Formel:

$$Y^* = \arg \max_Y P(Y | X, \lambda). \quad (26)$$

Wendet man für die Ermittlung der wahrscheinlichsten Labelsequenz eine naive *Brute-Force-Methode* an, so müssen die Wahrscheinlichkeiten für *alle* potentiellen Labelsequenzen bestimmt werden, wobei $\# \text{Sequenzen} = L^{|\Sigma|}$. Dies ist bei einer langen Beobachtungsfolge und einer großen Zustandsmenge nicht realisierbar. Einen Ausweg aus diesem Problem liefert der *Viterbi-Algorithmus*, der in $O(|\Sigma|^2 \cdot L)$ die optimale Labelsequenz berechnet [Dur98].

Wie in der Einleitung zu Kapitel 3 beschrieben, erfolgt die Klassifikation bei der Textannotation nicht direkt auf der Beobachtungssequenz $X = x_1, \dots, x_L$, sondern auf einer Sequenz von L binären Featurevektoren $\vec{\Psi}(X) = \vec{\Psi}(x_1), \dots, \vec{\Psi}(x_L)$. Da es sich um einen sequentiellen Ansatz handelt, fließen sowohl Transitions- als auch Observations-features in die Betrachtung mit ein.

Dies bedeutet, dass statt des Tokens x_t der Featurevektor $\vec{\Psi}(x_t)$ betrachtet wird. Für diesen wird eine *Multi-Bernoulli-Verteilung* angenommen, da davon ausgegangen wird,

dass die Observationsfeatures unabhängig voneinander auftreten. Die Wahrscheinlichkeiten der Observationsfeatures $f_1(Y, X|t), \dots, f_m(Y, X|t)$ befinden sich in Matrix B . Der Maximum-Likelihood-Schätzer hat nun folgende Form:

$$b_\sigma(\bar{\Psi}(o)) = \frac{B_\sigma(\Psi_1(o))}{\sum_{o'} B_\sigma(\Psi_1(o'))} \cdot \dots \cdot \frac{B_\sigma(\Psi_m(o))}{\sum_{o'} B_\sigma(\Psi_m(o'))}. \quad (27)$$

Auch hier verhindern Pseudocounts, dass Auftreten von Null-Wahrscheinlichkeiten (24). Die Wahrscheinlichkeiten der Transitions-Features $g_{m+1}(Y|t), \dots, g_{m+n}(Y|t)$ befinden sich in Matrix A und im Vektor π . Der Maximum-Likelihood-Schätzer bleibt unverändert..

Für die Berechnung der Wahrscheinlichkeit ergibt sich folgende Formel:

$$P(X, Y|\lambda) = \pi_{y_1} \cdot b_{y_1}(\Psi(x_1)) \cdot \prod_{t=2}^L a_{y_{t-1}y_t} \cdot b_{y_t}(\Psi(x_2)). \quad (28)$$

Für die Berechnung der wahrscheinlichsten Labelsequenz kann auch hier der Viterbi-Algorithmus verwendet werden.

3.4 Conditional Random Fields

Neben Hidden Markov Modellen zählen auch Conditional Random Fields (CRFs) zu den sequentiellen, probabilistischen Verfahren. Im Gegensatz zu den generativen HMMs basiert bei den diskriminativen CRFs die Berechnung der Labelsequenz Y auf der Wahrscheinlichkeit $P(Y|X, \vec{\lambda})$.

Bei einem CRF auf (X, Y) handelt es sich um einen ungerichteten Graphen $G = (V, E)$. Dieser wird durch die Sequenz Y initiiert, d.h. jede Position in Y entspricht einem Knoten V in G . Im vorliegenden Fall des Sequenzlernens handelt es sich dabei um eine Kette, d.h. für alle benachbarten y_{t-1} und y_t existiert eine Kante $E = \{y_{t-1}, y_t\}$. Die Beobachtungsfolge X ist ebenfalls Bestandteil des Graphen. Dies bedeutet, dass in G auch Kanten der Form $E = \{y_t, x_{t \pm s}\}$ mit $s = 0, 1, 2, 3, \dots$ existieren (vgl. *Abbildung 3.5*).

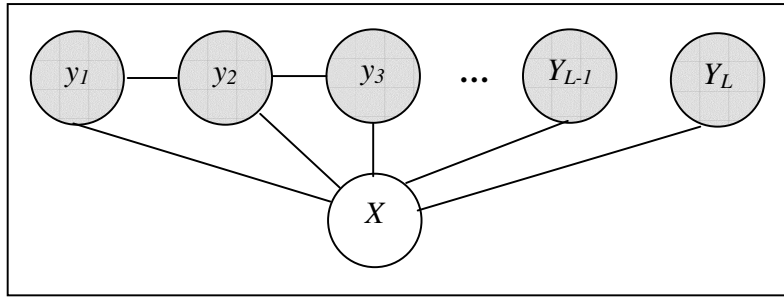


Abbildung 3.5: Der Graph G visualisiert die Abhängigkeiten der beteiligten Variablen.

Für den weiteren Verlauf ist es von besonderer Bedeutung, die maximalen 2er-Cliquen in G zu betrachten, da aus ihnen die Observationsfeatures $f(Y, X|t)$ und die Transitionsfeatures $g(Y|t)$ hervorgehen. Der daraus resultierende Featurevektor (vgl. (5)) wird bei der Bestimmung der Wahrscheinlichkeit durch den Vektor $\vec{\lambda}$ des CRF gewichtet.

Für das CRF auf (X, Y) ergibt sich damit folgende Wahrscheinlichkeit:

$$P(Y | X, \vec{\lambda}) = \frac{1}{Z(X, \vec{\lambda})} \sum_L \exp \left[\vec{\lambda} \cdot \sum_{m+n} (f, g)^T \right], \text{ wobei} \quad (29)$$

$$Z(X, \vec{\lambda}) = \sum_{\sigma \in \Sigma} \sum_L \exp \left[\vec{\lambda} \cdot \sum_{m+n} (f, g)^T \right], \quad (30)$$

der Normalisierung der Wahrscheinlichkeiten dient.

Das Optimierungsproblem für ein CRF besteht darin, den globalen Vektor $\vec{\lambda}$ so zu bestimmen, dass die Wahrscheinlichkeit einer gegebenen Menge von Trainingsbeispielen $T = \{(X^i, Y^i)\}$ mit $i = 1 \dots n$ maximiert wird. Dieses Verfahren wird auch als *Maximum-Log-Likelihood-Methode* bezeichnet, da die bedingten Wahrscheinlichkeiten logarithmiert werden.

Das Optimierungsproblem lautet demnach: Maximiere bzgl. $\vec{\lambda}$:

$$Z(X^i, Y^i, \vec{\lambda}) = \sum_i \log P_{\vec{\lambda}}(Y^i | X^i), \text{ für } i = 1 \dots n, \text{ so dass gilt: } \sum_Y P(Y | X, \vec{\lambda}) = 1. \quad (31)$$

Ein Verfahren, welches den optimalen Gewichtsvektor $\bar{\lambda}$ bestimmt, ist das in [Laf01] vorgestellte *Iterative-Scaling-Verfahren*, welches mit einem hohen Aufwand bezüglich der Laufzeit aufwartet. Ein alternativen Ansatz bildet das *Gradient-Tree-Boosting-Verfahren*, das in [Die02] beschrieben wird.

Das Verfahren der Matrixberechnung, mit deren Hilfe die Wahrscheinlichkeit $P(Y|X, \lambda)$ für kettenförmige CRFs effizient berechnet werden kann, ist in [Wal04] dargestellt.

4. Experimente und Ergebnisse

Der nachfolgende Teil der Arbeit beschreibt die durchgeführten Experimente und die daraus gewonnenen Ergebnisse. In Abschnitt 4.1 werden die Daten und die Datenaufbereitung näher erläutert. Kapitel 4.2 beschreibt die Durchführung der Experimente und in Abschnitt 4.3 erfolgt die Auswertung der Experimente und der Vergleich der 3 verwendeten Verfahren unter verschiedenen Evaluierungskriterien.

4.1 Datenaufbereitung

Die Experimente wurden auf Artikeln des Volltext Dokumenten Servers (VDS) des KOBV durchgeführt. Dieser verfügt über mehr als 1 Million Artikel der Verlage Kluwer, Springer und Elsevier, die zwischen den Jahren 1997 und 2003 publiziert wurden. Aus dieser Gesamtmenge wurde eine zufällige Teilmenge von insgesamt 79 Artikeln erzeugt, welche insgesamt 828 Zitationen beinhaltet. Die Artikel stehen als PDF-Dateien zur Verfügung und wurden mit dem Programm `pdf2txt` in Volltext-Dateien umgewandelt. Anschließend wurde der Referenzblock jedes Artikels per Hand extrahiert. Dieser Schritt lässt sich jedoch automatisieren [Din99]. Da Anfang und Ende einer Referenz nicht immer eindeutig sind, insbesondere gilt dies für mehrzeilige Referenzen, wurden die einzelnen Referenzen manuell durch zusätzliche Leerzeilen voneinander getrennt.

Im Anschluss daran wurde jede einzelne Referenz von Hand mit semantischen Labels in IOB-Notation getaggt. IOB ist die Abkürzung für *inner*, *outer* und *beginning*. Dabei wird jedem Token zunächst eine semantische Klasse zugeordnet, bzw. die Klasse

outer, wenn es sich keiner gegebenen Klasse zuordnen lässt. Aus Gründen der Übersicht, werden im weiteren Text Abkürzungen der Klassen verwendet. Die Menge aller semantischen Klassen und die jeweilige Abkürzung ist in *Abbildung 4.1* zu sehen.

Kürzel	Klasse	Kürzel	Klasse
ART	Artikel	NUM	Nummer d. Referenz
AUTF	Vorname e. Autors	PAGE	Seitenangabe
AUTL	Nachname e. Autors	PLACE	Ort d. Herausgebers
BOOK	Buch	PUB	Name d. Herausgebers
CONF	Konferenz	VOL	Band
ISS	Heft	YEAR	Jahr
JOUR	Zeitschrift	OUT	keiner Klasse zuzuordnen

Abbildung 4.1.: Verwendete Semantische Klassen und ihre Abkürzung.

```
[1] T. Davenport, D. DeLong, and M. Beers, "Successful knowledge management
projects," Sloan Management Review, vol. 39, no. 2, pp. 43-57, 1998.

<OUT> <B_NUM> <OUT> <B_AUTF> <OUT> <B_AUTL> <OUT>
[ 1 ] T . Davenport ,
<B_AUTF> <OUT> <B_AUTL> <OUT> <OUT> <B_AUTF> <OUT>
D . DeLong , and M .
<B_AUTL> <OUT> <OUT> <B_ART> <I_ART> <I_ART> <I_ART>
Beers , " Successful knowledge management projects
<I_ART> <OUT> <B_JOUR> <I_JOUR> <I_JOUR> <OUT> <OUT>
, " Sloan Management Review , vol
<OUT> <B_VOL> <OUT> <OUT> <OUT> <B_ISS> <OUT> <OUT>
. 39 , no . 2 , pp
<OUT> <B_PAGE> <I_PAGE> <I_PAGE> <OUT> <B_YEAR> <OUT>
. 43 - 57 , 1998 .
```

Abbildung 4.2.: Eine Referenz mit IOB-Notation.

Da sich einige Entitäten wie der Titel eines Artikels oft über mehrere Token erstrecken, ist es notwendig, den Anfang der Entität besonders zu kennzeichnen. Dieser wird dabei mit einem BEGIN-Tag markiert, z.B. <B_ART>, die restlichen Token der selben Entität mit einem INSIDE-Tag, z.B. <I_ART>. Dadurch ist es möglich, aufeinander folgende Entitäten derselben Klasse eindeutig voneinander zu unterscheiden. *Abbildung 4.2* zeigt ein Beispiel Referenz mit IOB-Notation.

Observations-Feature	Bedeutung
Word-Feature	Betrachtet für alle Token der Trainingsmenge, ob Sie mit dem aktuellen Token übereinstimmen
Unknown-Feature	Ist gesetzt, wenn das aktuelle Token nicht in der Trainingsmenge enthalten ist
Regular-Expression-Feature	Prüft für 9 unterschiedliche reguläre Ausdrücke, ob diese auf das aktuelle Token zutreffen, z.B.: <ul style="list-style-type: none"> - Token beginnt mit Buchstaben - Token enthält Zahl - Token ist Sonderzeichen
Prefix-Feature	Betrachtet für alle Präfixe der Trainingsmenge der Länge 1-4, ob diese auch Präfix des aktuellen Tokens sind
Suffix-Feature	analog zu Prefix-Feature
2-Gramm-Feature	Betrachtet und prüft alle Folgen von 2 Buchstaben der Trainingsmenge mit dem aktuellen Token
3-Gramm-Feature	analog zu 2-Gramm-Feature
Transitions-Feature	Bedeutung
Start-Feature	Betrachtet das Startlabel der Referenz
End-Feature	Betrachtet das letzte Label der Referenz
Edge-Feature	Betrachtet den Übergang zwischen zwei Labels

Abbildung 4.3.: Die 10 verschiedenen Arten von Features und die Aufteilung in Observations- und Transitions-Features.

4.2 Durchführung

Die SVM Experimente wurden mit SVM^{light} von Thorsten Joachims² durchgeführt. Für die Experimente der HMMs nutzte der Autor eine eigene Implementierung. Die Versuche der CRFs basieren auf dem Programmpaket von Sunita Sarawagi³. Letztgenanntes wurde von Felix Jungermann im Rahmen einer Diplomarbeit [Jun06] durch eine Vielzahl neuer Features erweitert. Die Anpassung und Weiterentwicklung des Programms für die spezifische Problemstellung wurde vom Autor vorgenommen.

Die Experimente für alle drei Verfahren wurden in mehreren Läufen durchgeführt, wobei jeder Lauf aus einer Trainings-, einer Test- und einer Evaluierungsphase besteht. In der Trainingsphase wurde aus den 828 getaggten Referenzen zunächst eine Trainingsmenge generiert, aus der eine Menge von Features extrahiert wurden und anhand derer das jeweilige Modell gelernt wurde. Auf die Generierung der Trainingsmengen wird bei der Evaluierung genauer eingegangen. *Abbildung 4.3* gibt einen Überblick über alle Arten von Features, die berücksichtigt wurden sowie deren Bedeutung.

Bei der Extraktion der Features wurden Läufe mit 4 unterschiedlichen Feature-Sets durchgeführt, die in *Abbildung 4.4* aufgeführt sind.

Name	Enthaltene Features
FS1	Word-Feature, Unknown-Feature, Regular-Expression-Feature, Start-Feature, End-Feature, Edge-Feature (Fenstergröße 1)
FS2	FS1 und zusätzlich: Prefix-Feature, Suffix-Feature, 2Gramm-Feature, 3Gramm-Feature (Fenstergröße 1)
FS3	FS2 (Fenstergröße 3)
FS4	FS2 (Fenstergröße 5)

Abbildung 4.4.: Die 4 unterschiedlichen Feature-Sets.

² <http://svmlight.joachims.org/>

³ <http://crf.sourceforge.net/>

Sämtliche Versuche wurden sowohl auf der gesamten Labelmenge (vgl. *Abbildung 4.1*) als auch auf einer reduzierten Labelmenge durchgeführt. Es handelt sich dabei um die minimale Menge von Labeln, die nötig ist, um den in der Einführung beschriebenen Zitationsgraphen zu generieren. Diese Menge besteht aus den 5 Klassenlabeln ART, AUTF, AUTL, JOUR*, YEAR und dem Label OUT*, wobei die Zuordnung der Klassen zu den Labeln in *Abbildung 4.5* dargestellt ist.

Im Anschluss an die Trainingsphase schließt sich die Testphase an, in welcher die Testmenge, das sind alle Referenzen, die nicht Bestandteil der Trainingsmenge sind, mithilfe des gelernten Modells getaggt wird. Für jedes Token in der Testmenge existiert nun ein manuell und ein automatisch getaggt Label. Beide Label sind notwendig, um das Verfahren zu evaluieren.

Kürzel	Enthält Klassen:
ART	Artikel
AUTF	Vorname e. Autors
AUTL	Nachname e. Autors
JOUR*	Buch (BOOK), Zeitschrift (JOUR), Konferenz (CONF)
YEAR	Jahr
OUT*	Heft (ISS), Nummer d. Referenz (NUM), Seitenangabe (PAGE), Ort d. Herausgebers (PLACE), Name d. Herausgebers (PUB), Band (VOL), keiner Klasse zuzuordnen (OUT)

Abbildung 4.5.: Die reduzierte Labelmenge besteht aus 6 Elementen.

Die Verfahren wurden mit zwei unterschiedlichen Methoden evaluiert: einer 5-fachen Kreuzvalidierung und einer strategischen Kreuzvalidierung. Bei der 5-fachen Kreuzvalidierung wurden aus der Menge aller Referenzen 5 gleichgroße, disjunkte Teilmengen gebildet. Evaluiert wurden pro Feature-Set und Labelmenge 5 Durchläufe, wobei in jedem Lauf 4 Teilmengen dem Training dienen, anhand derer das Verfahren ein Modell lernt. Auf der verbleibenden Teilmenge wurde das Modell getestet, und auf den

Ergebnissen wurden die Standardmaße der Evaluierung angewendet. Bei der 5-fachen Kreuzvalidierung wird ein bereits bekannter Zitationsstil evaluiert.

Für die strategische Kreuzvalidierung wurden alle 828 Zitationen auf 6 verschiedene, disjunkte Referenzgruppen aufgeteilt, die sich eindeutig in der Art und Weise der Zitation voneinander unterscheiden. Trainiert wurde das Modell mit den Referenzen von 5 Gruppen, die verbleibende Gruppe bildete die Testmenge. Die Gruppierung basiert auf dem Vorhandensein und der Reihenfolge der wichtigsten semantischen Klassen: AUTL, AUTF, ART, JOUR, BOOK, CONF, YEAR. Die anderen Klassen wurden aufgrund geringem Vorhandensein (VOL, ISS, PUB, PLACE) innerhalb der Referenzen bzw. Irrelevanz bzgl. der Eindeutigkeit eines Artikels (NUM, PAGE, OUT) ignoriert. Ebenso spielen unterschiedliche Klammerung und das Vorhandensein von Interpunktionszeichen keine Rolle. Das Ziel war es, möglichst heterogene Trainingsmengen und in sich homogene Testmengen zu bilden, um die Verfahren auf bisher ungesehene Zitationsstile zu testen.

Gruppe A (109 Referenzen):					
AUTL	AUTF	ART	JOUR BOOK CONF	YEAR	
Gruppe B (30 Referenzen):					
AUTL	AUTF	JOUR BOOK CONF	YEAR		
Gruppe C (245 Referenzen):					
AUTL	AUTF	YEAR	ART	JOUR BOOK CONF	
Gruppe D (79 Referenzen):					
AUTL	AUTF	YEAR	JOUR BOOK CONF		
Gruppe E (50 Referenzen):					
AUTF	AUTL	ART	JOUR BOOK CONF	YEAR	
Gruppe F (315 Referenzen):					
AUTF	AUTL	JOUR BOOK CONF	YEAR		

Abbildung 4.6: 6 Referenzgruppen für die strategische Kreuzvalidierung.

Die Größe der Gruppen ist dabei sehr unterschiedlich und liegt zwischen 30 und 315. Die Gruppe A, bestehend aus 109 unterschiedlichen Referenzen, enthält Referenzen, die mit

einem oder mehreren Autoren beginnen, wobei die Abfolge AUTL AUTF ist. Danach schließt sich ART an, gefolgt von JOUR, BOOK oder CONF. Den Abschluss bildet YEAR. Die Referenzgruppen, die jeweilige Reihenfolge der wichtigen semantischen Klassen und die Anzahl der Referenzen ist in *Abbildung 4.6* zu sehen. Bei der strategischen Kreuzvalidierung wird ein unbekannter Zitationsstil evaluiert.

4.3 Ergebnisse

Für alle drei Verfahren wurden verschiedene Testreihen durchgeführt. Dabei wurde zum einen die allgemeine Performance der Verfahren untersucht. Weiterhin wurde analysiert, wie genau der Verfahren bei ungesesehenen Zitationsstilen arbeiten. Die dritte Versuchsreihe basierte auf einer reduzierten Labelmenge (vgl. *Abbildung 4.5*). Jede der drei Testreihen wurde für 4 unterschiedliche Feature-Sets durchgeführt (vgl. *Abbildung 4.4*), wodurch der Einfluss der Fensterbreite und die Auswahl der Features auf die Performance der Verfahren geprüft wurde. Die dabei ermittelten Micro-F1-Werte (8) geben die durchschnittliche Token-Genauigkeit, die Macro-F1-Werte (9) die durchschnittliche Klassen-Genauigkeit an.

4.3.1 Allgemeine Performance

Die allgemeine Performance eines Verfahrens gibt Auskunft darüber, mit welcher Genauigkeit korrekte Klassenlabel zugeordnet werden. Diese bezieht sich auf bereits bekannte Zitationsstile, was vom Autor als Normalfall angenommen wird. Die Evaluierung basiert auf einer 5-fachen Kreuzvalidierung, deren Ergebnis in *Abbildung 4.7* dargestellt.

Bei allen drei Verfahren liegen die Micro-F1-Werte über den jeweiligen Macro-F1-Werten, was den niedrigen Werten einiger kleiner Klassen zuzuschreiben ist. CRFs erzielen sowohl für Micro-F1 (0,92) als auch für die Macro-F1 (0,76) die höchsten Werte. Alle Verfahren profitieren von einer größeren Anzahl an Features. Ein breiteres Beobachtungsfenster bewirkt besonders bei den SVMs einen starken Anstieg der Werte, während die Werte der HMMs dabei nur bedingt ansteigen.

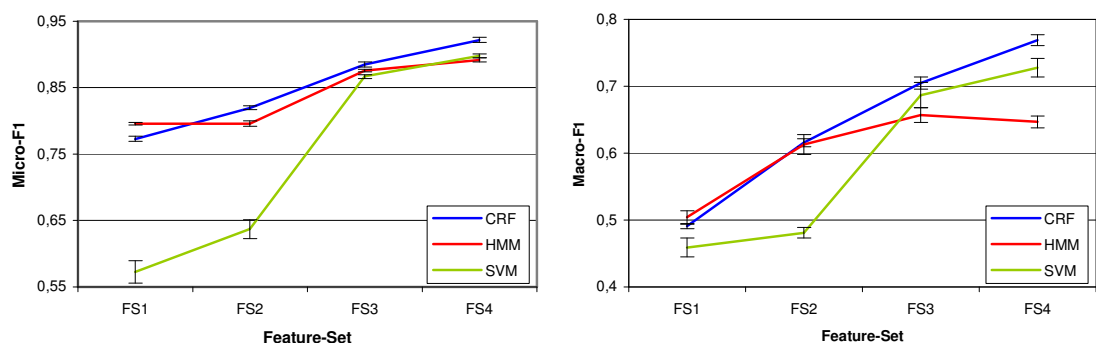


Abbildung 4.7 Ergebnis einer 5-fachen Kreuzvalidierung anhand der durchschnittlichen Micro-F1-Werte (links) und Macro-F1-Werte (rechts) sowie der jeweils auftretende Standardfehler.

4.3.2 Performance auf ungesehenen Zitationsstilen

Die Performance auf ungesehenen Zitationsstilen wird mittels der strategischen Kreuzvalidierung untersucht. Es handelt sich hierbei um einen Spezialfall, der zeigen soll, wie flexibel die Verfahren auf neue und bisher unbekannte Zitationsstile reagieren. Das Ergebnis ist in *Abbildung 4.8* zu sehen.

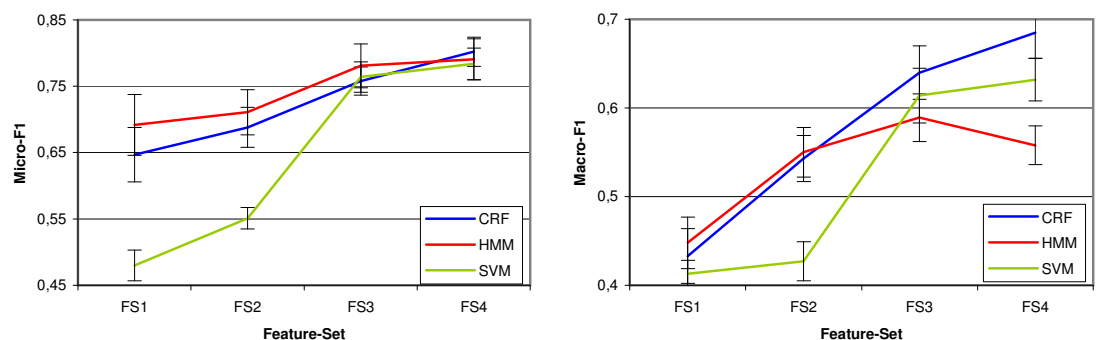


Abbildung 4.8 Durchschnittliche Micro-F1 Werte (links) und Macro-F1 Werte (rechts) der strategischen Kreuzvalidierung sowie der jeweilige Standardfehler.

Erwartungsgemäß liegen die Werte der strategischen Kreuzvalidierung deutlich unter denen der 5-fachen Kreuzvalidierung. Die Kurven beider Evaluierungen weisen starke Ähnlichkeit auf. So die Micro-F1-Werte ebenfalls über den Macro-F1-Werten, was

wiederum durch einige kleine Klassen mit niedrigen Werten hervorgerufen wird. Auch bei unbekanntem Zitationsstil schneiden CRFs für Micro-F1 (0,80) und Macro-F1 (0,69) besser als die anderen beiden Verfahren ab. Ein breiteres Beobachtungsfenster bewirkt bei den SVMs einen besonders ausgeprägten Anstieg der Werte. HMMs erzielen bei niedriger Anzahl der Features und kleinem Fenster bessere Werte als CRFs, profitieren von einem größeren Fenster gar nicht oder weniger als die anderen beiden Verfahren.

4.3.3 Performance bei reduzierter Labelmenge

Die Verfahren wurden ebenfalls für den reduzierten Labelsatz (vgl. *Abbildung 4.5*), der für den Aufbau eines Zitationsgraphen verwendet werden kann, mittels der 5-fachen Kreuzvalidierung evaluiert. Ergebnisse sind in *Abbildung 4.9* dargestellt.

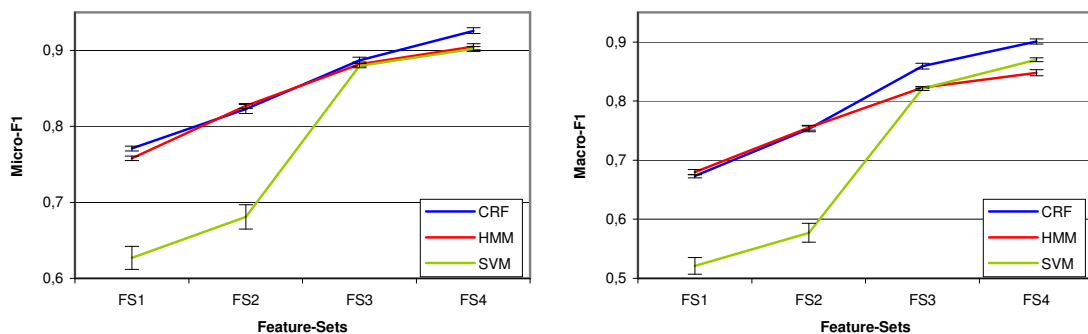


Abbildung 4.9 Ergebnis einer 5-fachen Kreuzvalidierung bei reduzierter Labelmenge mittels Micro-F1 Werte (links) und Macro-F1 Werte (rechts). Die Kurven basieren auf den Durchschnittswerten der 5 Läufe. Abgebildet ist ebenfalls der jeweilige Standardfehler.

Im Vergleich zur vollständigen Labelmenge sind die Micro-F1-Werte bei reduzierter Labelmenge auf ähnlichem Niveau, während die Macro-F1-Werte deutlich darüber liegen. Letzteres kann durch den Wegfall mehrerer kleiner Klassen mit niedrigen Werten begründet werden. Auch bei einer reduzierten Labelmenge schneiden CRFs bei Micro-F1 (0,93) und Macro-F1 (0,90) besser als die anderen Verfahren ab. Alle Verfahren

profitieren von einer größeren Anzahl von Features und einem breiteren Beobachtungsfenster, bei letztem genanntem die SVMs besonders stark.

4.3.4 Performancevergleich der Verfahren

Der Vergleich der Verfahren basiert auf dem paarweisen, zweiseitigem t-Test, welcher für jeweils zwei Verfahren feststellt, ob sich die Ergebnisse beider Verfahren signifikant voneinander unterscheiden. Das Signifikanzniveau α besagt, mit welcher Wahrscheinlichkeit die Werte beider Stichproben aus unterschiedlichen Verteilungen stammen. Es wurde für jede der Testreihen untersucht, ob in mehr als der Hälfte der Feature-Sets eines der Verfahren ein signifikant besseres Ergebnis als das andere liefert.

Für die allgemeine Performance gilt:

- Bei $\alpha = 97,5\%$ sind die Micro-F1-Werte der HMMs für 3 der 4 Feature-Sets besser als die der SVMs.
- Bei $\alpha = 97,5\%$ sind die Micro-F1-Werte der CRFs für alle Feature-Sets besser als die der SVMs.
- Bei $\alpha = 95\%$ sind die Macro-F1-Werte der CRFs für 3 der 4 Feature-Sets besser als die der SVMs.

Für die Performance auf ungesehenen Zitationsstilen ist beobachtet worden:

- Bei $\alpha = 95\%$ sind die Micro-F1-Werte der CRFs für 3 der 4 Feature-Sets besser als die der SVMs.
- Bei $\alpha = 90\%$ sind die Macro-F1-Werte der CRFs für 3 der 4 Feature-Sets besser als die der SVMs.

Für die Performance bei reduzierter Labelmenge wurde festgestellt:

- Bei $\alpha = 95\%$ sind die Micro-F1-Werte der HMMs für 3 der 4 Feature-Sets besser als die der SVMs.
- Bei $\alpha = 99,5\%$ sind die Micro-F1-Werte der CRFs für 3 der 4 Feature-Sets besser als die der SVMs und bei $\alpha = 95\%$ für alle Feature-Sets.
- Bei $\alpha = 99\%$ sind die Micro-F1-Werte der CRFs für alle Feature-Sets besser als die der SVMs.

Betrachtet man die Menge aller Experimente, so lassen sich folgende Aussagen machen:

- HMMs sind für FS1 in 5 von 6 Fällen und für FS2 in 6 von 6 Fällen zu 95% signifikant besser als SVMs und niemals schlechter.
- SVMs sind für FS4 in 4 von 6 Fällen für FS4 zu 95% signifikant besser als HMMs.
- CRFs sind für FS1 in 5 von 6 Fällen und für FS2 und FS4 in 6 von 6 Fällen zu 95% signifikant besser als SVMs und niemals schlechter.
- CRFs sind für FS4 in 5 von 6 Fällen zu 95% signifikant besser als HMMs und niemals schlechter.

5. Zusammenfassung

Die vorliegende Arbeit befasst sich mit dem Problem der Zitationsextraktion. Dabei wurden die drei Verfahren Support Vector Machine (SVM), Hidden Markov Modell (HMM) und Conditional Random Field (CRF) vorgestellt, die dieses Problem mit Hilfe der Textannotation lösen.

Es wurden verschiedene Experimente zur Ermittlung der Performance der drei Verfahren durchgeführt. Der paarweise Vergleich der Verfahren ergab, dass HMMs und CRFs bei mehreren Experimenten bessere abschnitten als SVMs. Die beste allgemeine Performance erbringen CRFs. Bei der Reduzierung der Klassenlabel auf die für einen Zitationsgraphen minimal notwendige Labelmenge liefern CRFs ebenfalls bessere Ergebnisse als die beiden anderen Verfahren.

6. Literaturangaben

- [Alt03] Y.Altun, I.Tsochantaridis, T.Hofmann: Hidden Markov Support Vector Machines. In Proceedings of the ICML, 2003.
- [Cal00] A.McCallum, K.Nigam, J.Rennie, K.Seymore: Automating the Construction of Internet Portals with Machine Learning. Information Retrieval, 3 (2), 2000, p.127-163.
- [Day05] M.Y.Day, T.H.Tsai, C.L.Sung, C.W.Lee, S.H.Wu, C.S.Ong, W.L.Hsu: A Knowledge-based Approach to Citation Extraction. In Proceedings of the IRI, Las Vegas, 2005, p.50-55.
- [Die02] T.G.Dietterich: Machine Learning for Sequential Data: A Review. In Proceedings of the Fourth International Workshop on Statistical Techniques in Pattern Recognition, 2002.
- [Din99] Y.Ding, G.Chowdhury, S.Foo: Template mining for the extraction of citation from digital documents. In Proceedings of the Second Asian Digital Library Conference, Taiwan, 1999, p.47-62.
- [Dur98] R.Durbin, S.Eddy, A.Krogh, G.Mitchison: Biological sequence analysis. Cambridge University Press, Cambridge, UK, 1998.
- [Jun06] F.Jungermann: Named Entity Recognition mit Conditional Random Fields. Diplomarbeit, Lehrstuhl für Künstliche Intelligenz, Universität Dortmund, 2006.
- [Laf01] J.Lafferty, A.McCallum, F.Pereira: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the ICML, 2001.
- [Law99] S.Lawrence, C.L.Giles, K.Bollacker: Autonomous Citation Matching. Proceedings of the Third International Conference on Autonomous Agents, Seattle, Washington, May 1-5, ACM Press, New York, NY, 1999.
- [Man99] C.D. Manning, H.Schütze: Natural Language Processing. MIT Press, Cambridge, Massachusetts, 1999.
- [Pen04] F.Peng, A.McCallum: Accurate Information Extraction from Research Papers using Conditional Random Fields. In Proceedings of the HLT-NAACL, 2004, p.329-336.

- [Sey99] K.Seymore, A.McCallum, R.Rosenfeld: Learning hidden Markov model structure for information extraction. AAAAI-99 Workshop on Machine Learning for Information Extraction, 1999, p.37-42.
- [Vap74] V.N.Vapnik, A.J.Chevonnkis: Theory of Pattern Recognition. Nauka, Moskau, 1974.
- [Wal04] H.M.Wallach: Conditional Random Fields: An Introduction. Technical Report No. MS-CIS-04-21, University of Pennsylvania Department of Computer and Information Science, 2004.
- [Wei04] B.Wellner, A.McCallum, F.Peng, M.Hay: An Integrated, Conditional Model of Information Extraction and Coreference with Application to Citation Matching. 2004.
- [Yia97] P.Yianlios: The LikeIt intelligent string comparison facility. Technical Report 97-093, NEC Research Institute, 1997.