Hidden Markov Models vs. Conditional Random Fields bei der Named Entity Recognition

Till Blume
AG Knowledge Discovery
Institut für Informatik
Christian-Albrechts-Universität zu Kiel
tbl@informatik.uni-kiel.de

ABSTRACT

Conditional Random Fields erfreuen sich immer größerer Beliebtheit und erzielen bei einigen Machine Learning Problemen zunehmend bessere Ergebnisse als die Hidden Markov Models. Dies lässt sich vermutlich darauf zurückführen, dass mit Conditional Random Fields besser komplexe Abhängigkeiten innerhalb der Eingaben modelliert und ausgenutzt werden können. In dieser Arbeit werden die grundlegenden Konzepte der beiden Modelle untersucht und einige ausgewählte experimentelle Analysen verglichen. Dabei werden die Stärken und Schwächen von Hidden Markov Models und Conditional Random Fields im Allgemeinen und im speziellen Kontext der Named Entity Recognition herausgearbeitet. Dieses Ergebnis kann als Orientierung dienen, inwieweit eine Hidden Markov Model Lösung durch Anwenden von Conditional Random Fields verbessert werden kann.

1. EINLEITUNG

In der Informationsverarbeitung müssen immer wieder Daten analysiert, Abhängigkeiten gefunden und so Modelle für den vorliegenden Sachverhalt entwickelt werden. Hierbei ist es ein immer wiederkehrender Wunsch dies maschinell erledigen zu lassen. Es stehen dem Entwickler heutzutage unzählige Möglichkeiten aus dem Bereich der künstlichen Intelligenz zur Verfügung. Insbesondere beim Machine Learning gab es in den letzten Jahren viele Entwicklungen, sodass es einem Einsteiger schwer erscheinen mag, die für ihn passende Technologie zu finden [1].

In dieser Arbeit werden zwei Modelle besonders hervorheben und ihre Stärken und Schwächen gegeneinander abgewogen. Das sind zum einen die Hidden Markov Models (HMM) und zum anderen die Conditional Random Fields (CRFs). Conditional Random Fields erfreuen sich immer größerer Beliebtheit und erzielen bei einigen Machine Learning Problemen zunehmend bessere Ergebnisse als die Hidden Markov Model. [2, 3, 4, 5]. Die beiden graphentheoretischen Modelle werden in Kapitel 2 und 3 kurz vorgestellt. Es wird sich darauf beschränkt, die Aspekte zu erläutern, die in Kapitel 4 bei der Gegenüberstellung aufgegriffen werden. Diese sind das Lernen des Modells und die Lösung für das Problem der Klassifikation, insbesondere die Aufgabe der Named Entity Recognition (NER). Anschließend werden einige ausgewählte experimentelle Analysen in Kapitel 5 ausgewertet und in Bezug zueinander gesetzt. Außerdem wird überprüft, wie stark sich die in Kapitel 4 erläuterten Stärken und Schwächen in realen Versuchen auswirken. Die untersuchten Analysen wurden so ausgewählt, dass sie sowohl eine direkte Gegenüberstellung zulassen, als auch einen Einblick in unterschiedlich komplexe Anwendungsgebiete geben. Einige Arbeiten, die sich mit dem Vergleich von HMM und CRFs befasst haben, kamen ebenfalls zu dem Schluss, dass CRFs besser komplexe Abhängigkeiten innerhalb der Eingabe handhaben können und deshalb häufig den HMM überlegen sind [6, 7]. Diese Erkenntnis konnte durch die Auswertung mehrerer experimenteller Analysen bestärkt werden.

2. HIDDEN MARKOV MODELS

Mit Hilfe von graphentheoretischen Modellen können Wahrscheinlichkeitsverteilungen in Daten aufgedeckt und ausgenutzt werden. Sieht man Daten als endliche, zeitlich geordnete Sequenz von Beobachtungen $\mathtt{x}$, so kann eine Beobachtung zum Zeitpunkt t als $\mathtt{x}_t$ bezeichnet werden. Die Beobachtung $\mathtt{x}_t$ kann dabei einen beliebigen Wert aus einem vordefinierten Alphabet annehmen [8]. Mit Hilfe graphentheoretischer Modelle möchte man zu jeder Beobachtung $\mathtt{x}_t$ eine Aussage $\mathtt{y}_t$ treffen. Bei der Aufgabe der Named Entity Recognition (NER) aus dem Bereich des Natural Language Processing (NLP) geht es beispielsweise darum, jedes Wort $\mathtt{x}_t$ innerhalb eines Textes $\mathtt{x}$ zu identifizieren und anhand einer vordefinierten Menge $\mathtt{y}$ von benannten Entitäten zu klassifizieren. Der Zeitpunkt t ist dann die Position des Wortes $\mathtt{x}_t$ innerhalb des Satzes $\mathtt{x}$, wobei $\mathtt{x}_0$ den Anfang und $\mathtt{x}_T$ das Ende des Satzes markieren. T bezeichnet die Länge der Beobachtungssequenz. Eine benannte Entität $\mathtt{y}$ wird in diesem Kontext häufig auch Label (eng. Etikett) genannt. Benannte Entitäten könnten zum Beispiel `Person` [*I-PER*] und `Miscellaneous` [*I-MISC*] oder `Other` [*O*] (keine Entität) sein.

Wie in Beispiel 2.1 zu sehen ist, kann die Postion eines Wortes Einfluss auf die Aussage eines Satzes haben. Daher ist es für Aufgaben wie der NER keine Einschränkung, dass die Unabhängigkeitsannahme unter den Beobachtungen sowohl bei den Hidden Markov Models als auch den linear-

verketteten Conditional Random Fields gelockert wurde.

Beispiel 2.1.

x	*Herr*	*Vogel*	*sucht*	*den*	*Wolf.*
y	[*I-PER*]	[*I-PER*]	[*O*]	[*O*]	[*I-MISC*]
x	*Herr*	*Wolf*	*sucht*	*den*	*Vogel.*
y	[*I-PER*]	[*I-PER*]	[*O*]	[*O*]	[*I-MISC*]

Das HMM ist ein gerichtetes graphentheoretisches Modell, das auf dem Bayesschen Netz basiert. Mit einem Bayesschen Netz ist es möglich, die Zuordnung zu einem Label abhängig von einem oder mehreren Featuren zu bestimmen. Mit Hilfe von Featuren werden die Eigenschaften einer Beobachtung beschrieben. So besteht ein Wort zum Beispiel aus seinen Buchstaben. Das HMM ist eine Erweiterung des Bayesschen Netz für sequentielle Daten [9]. Es modelliert die Abhängigkeiten zwischen den Beobachtungen $\mathtt{x}$ und den Labeln $\mathtt{y}$ durch die gemeinsame Wahrscheinlichkeitsverteilung $p(\mathtt{y}, \mathtt{x})$. Außer der gelockerten Unabhängigkeitsbedingung zwischen den Beobachtungen werden noch zwei weitere Annahmen getroffen. Zum einen geht man davon aus, dass jede Beobachtung in einem nicht mehr sichtbaren Prozess generiert wurde. Dieser Prozess befindet sich zu jedem Zeitpunkt t in einem Zustand $\mathtt{y}_t$. Wenn ein Label einer Beobachtung bestimmt wird, wird also tatsächlich der versteckte Zustand ermittelt. Zum anderen ist eine wichtige Annahme, dass jeder Übergang des Prozesses von Zustand $\mathtt{y}_t$ zu $\mathtt{y}_{t+1}$ einer Logik folgt, die die Markov Bedingung erfüllt. Diese gibt an, dass jeder Zustand $\mathtt{y}_{t+1}$ unabhängig von allen Zuständen vor $\mathtt{y}_t$ ist. Versteckt hinter der Beobachtungen $\mathtt{y}_t$ existiert also eine Kausalkette, die den Prozess veranlasst hat, von $\mathtt{y}_t$ zu $\mathtt{y}_{t+1}$ zu wechseln. Vereinfacht ausgedrückt: Die Tatsache, dass *Till* eine `Person` ist, hat zu der Beobachtung der Buchstaben *T i l l* geführt.

Diese Eigenschaften lassen sich formal aufschreiben als

$$p(\mathtt{y}_{1:T}, \mathtt{x}_{1:T}) = \pi(\mathtt{y}_1) P(\mathtt{x}_1|\mathtt{y}_1) \prod_{t=2}^{T} p(\mathtt{y}_t|\mathtt{y}_{t-1}) p(\mathtt{x}_t|\mathtt{y}_t), \quad (1)$$

mit $\mathtt{x}_{1:T}$ als die Sequenz $\mathtt{x}_1 ... \mathtt{x}_T$ von Beobachtungen und $\mathtt{y}_{1:T}$ als die Sequenz $\mathtt{y}_1 ... \mathtt{y}_T$ von Labeln [8]. Mit $\pi(\mathtt{y}_1)$ wird dabei die initiale Wahrscheinlichkeit eines Zustandes angegeben. Statt $p(\mathtt{y}_{1:T}, \mathtt{x}_{1:T})$ wird im Folgenden auch die die Kurzschreibweise $p(\mathtt{y}, \mathtt{x})$ genutzt, um eine Abhängigkeitsmodellierung zwischen $\mathtt{x}$ und $\mathtt{y}$ zu beschreiben.

Die Abhängigkeiten lassen sich graphisch wie in Abbildung 1 darstellen. Die blauen Knoten repräsentieren die internen Zustände $\mathtt{y}$, die grauen die Beobachtungen $\mathtt{x}$. Jede Kante repräsentiert eine Abhängigkeit. Fehlt ein Kante, so existiert auch keine Abhängigkeit.

Die entscheidenden versteckten Zustände $\mathtt{y}$ werden durch das HMM als diskrete Zufallsvariable modelliert. Es gibt nur K verschiedene Belegungen für eine Zustandsvariable $\mathtt{y}_t$, z.B. K verschiedene Label bei der NER Aufgabe. Ein fundamentaler Bestandteil ist die Markov Kette innerhalb des HMM. Mit ihr werden Abhängigkeiten zwischen den möglichen Belegungen der Zustände modelliert. Dazu wird eine sogenannte $K \times K$ Transitionsmatrix der Form

$$\mathcal{T}_{ij} = p(\mathtt{y}_t = j | \mathtt{y}_{t-1} = i)$$

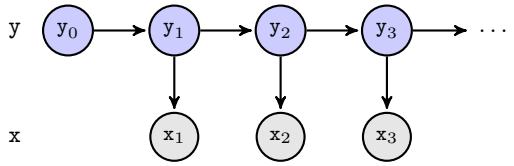

Abbildung 1: Versteckte Zustände $\mathtt{y}$ generieren beobachtbare Sequenz $\mathtt{x}$ [6].

gebildet [10]. Mit $\mathcal{T}_{ij}$ wird also die Wahrscheinlichkeit für einen Übergang von Zustand i zu j angegeben. Wählt man Indizes der Matrix wie folgt: Index $0 \triangleq$ Label **I-PER**, Index $1 \triangleq$ Label **O** und Index $2 \triangleq$ Label **I-MISC**, so ergibt sich eine Transitionsmatrix mit beispielhafter Belegung der Form

$$\mathcal{T} = \begin{pmatrix} 0.4 & 0.4 & 0.2 \\ 0.5 & 0.1 & 0.6 \\ 0.1 & 0.5 & 0.2 \end{pmatrix}.$$

Zusätzlich muss noch angegeben werden, mit welcher initialen Wahrscheinlichkeit die Zustände belegt werden. Diese gibt an, mit welcher Wahrscheinlichkeit der Zustand ein Startzustand einer Sequenz ist.

$$\pi = \begin{pmatrix} 0.6 \\ 0.3 \\ 0.1 \end{pmatrix}$$

Eine Markov Kette wird somit durch ihre initialen Zustandswahrscheinlichkeiten π und der Transitionsmatrix $\mathcal{T}$ definiert. Die Zustände $\mathtt{y}$ werden nur implizit angegeben.
Zu beachten ist, dass alle ausgehenden Transitionen eines Zustandes sich zu 1 summieren. Ebenso alle initialen Wahrscheinlichkeiten.

Mit Hilfe der Markov Kette kann bereits berechnet werden wie wahrscheinlich eine Zustandssequenz $\mathtt{y}$ zu einer Beobachtungssequenz $\mathtt{x}$ passt (vgl. Beispiel 2.2).

Beispiel 2.2.

$\boldsymbol{x}$	*Till*	*Blume*	*mag*	*Blumen.*
$\boldsymbol{y}$	*I-PER*	*I-PER*	*O*	*I-MISC*

$p(0012|\boldsymbol{x}) = 0.6 \cdot 0.4 \cdot 0.5 \cdot 0.5 = 0.06$

$\boldsymbol{x}$	*Till*	*Blume*	*mag*	*Blumen.*
$\boldsymbol{y}$	*I-PER*	*I-MISC*	*O*	*I-MISC*

$p(0212|\boldsymbol{x}) = 0.6 \cdot 0.1 \cdot 0.6 \cdot 0.5 = 0.018$

Das Modell in seiner jetzigen Form kann jedoch nur die Wortidentität benutzen. Ist dem Modell nur dieser eine Satz aus Beispiel 2.2 bekannt, so wäre es unter anderem nicht in der Lage zu erkennen, wenn ein anderer Name benutzt wird. Man müsste es mit allen existierenden Namen trainieren, um eine 100%ige Abdeckung zu erzielen. Um dies zu vermeiden ist es gängige Praxis sogenannte Featurefunktionen f_k zu definieren, die bestimmten Eigenschaften einer Beobachtung einer diskreten Zufallsvariable zuordnen. Ein Feature könnte die Länge (**L**) des Wortes, die Summe über alle ASCII Zeichen (**A**) oder Anzahl der Silben (**S**) sein. Wie relevant das jeweilige Feature für einen bestimmten Zustand ist, muss

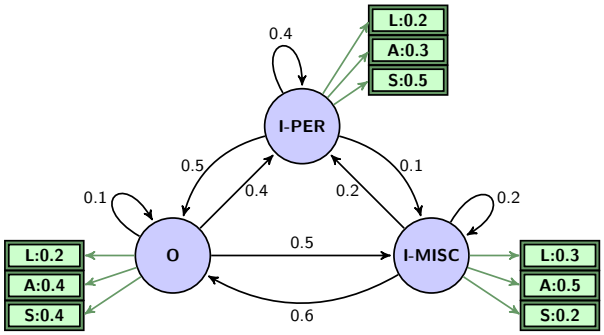

Abbildung 2: NER mittels HMM für I-PER, I-MISC und O (Darstellung inspiriert durch [11])

durch Training gelernt werden. Definiert man M Featurefunktionen, so lässt sich der Zustand über die Zusammenführung dieser Funktionen eindeutig definieren. Das HMM erlaubt jedoch lediglich Zustände, die durch eine einzige Variable angegeben werden. Daher müssen alle Featurevariablen in einer Variable zusammen gefasst werden. Diese Zuordnung erfolgt mittels einer Featurematrix $\mathcal{F}$, die zu jedem möglichen Zustand $\mathtt{y}_i$, mit $1 \leq i \leq K$ die Wahrscheinlichkeit des Features k angibt. Der Wert der Zustandsvariable $\mathtt{y}_i$ wird dann durch die gemeinsame Wahrscheinlichkeit über alle Feature f_{ij} mit $0 \leq j < M$ berechnet. In unserem Beispiel wäre folgende Featurematrix denkbar:

$$\mathcal{F} = \begin{pmatrix} 0.2 & 0.2 & 0.3 \\ 0.3 & 0.4 & 0.5 \\ 0.5 & 0.4 & 0.2 \end{pmatrix}.$$

Eine solche Markovkette kann graphisch wie in Abbildung 2 dargestellt werden. Blaue Knoten repräsentieren hier die verschiedenen Belegungen für einen Zustand $\mathtt{y}_t$, die aus ihren Featurefunktionen (Grüne Knoten) zusammengesetzt gesetzt sind. Die Kanten zeigen die Übergangswahrscheinlichkeiten, die initialen Wahrscheinlichkeiten wurden ausgespart.

Definition 2.3.
Ein Hidden Markov Model wird bestimmt durch $\pi, \mathcal{T}, \mathcal{F}$, hat K verschiedene Zustände und M verschiedenen Feature. Seien π die initialen Zustandswahrscheinlichkeiten $\pi = (\pi_1, .., \pi_K)$ der möglichen Zustände, $\mathcal{T}$ eine $K \times K$ Transistionsmatrix und $\mathcal{F}$ eine $M \times K$ Featurematrix. Mit $p(\boldsymbol{y}, \boldsymbol{x})$ kann einer zufälligen Sequenz von Beobachtungen $\boldsymbol{x}$ eine Sequenz $\boldsymbol{y}$ von Labeln zugeordnet werden. Als die Parameter θ eines Modells werden die konkreten Belegungen $\mathcal{T}$ bezeichnet.

Lernen

Das Lernen ist ein fundamentaler Bestandteil und trägt maßgeblich zur Qualität der Ergebnisse bei. Durch das Lernen werden die optimalen Parameter θ ermittelt. Dazu trainiert das Modell mit Daten $\mathcal{D} = \{\mathtt{x}^{(i)}, \mathtt{y}^{(i)}\}_{i=1}^{N}$, wobei jedes $\mathtt{x}^{(i)} = \{\mathtt{x}_1^{(i)}, \mathtt{x}_2^{(i)}, \ldots, \mathtt{x}_T^{(i)}\}$ eine Sequenz von Beobachtungen und jedes $\mathtt{y}^{(i)} = \{\mathtt{y}_1^{(i)}, \mathtt{y}_2^{(i)}, \ldots, \mathtt{y}_T^{(i)}\}$ eine Sequenz der korrekten Label ist [8]. Diese Daten werden häufig als der Korpus des Modells bezeichnet.

Um eine optimale Parameterverteilung zu ermitteln, wird die *maximum log likelihood* über

$$\log \mathcal{L}(\theta) = \log \prod_{i=1}^{N} p(\mathtt{x}^{(i)}; \theta) \tag{2}$$

gebildet [12]. Diese wird mit Hilfe des EM-Algorithmus berechnet, auf den in Kapitel 4 noch genauer eingegangen wird. In Gleichung 2 wird jedoch schon ersichtlich, dass beim Lernen die Beobachtungen die entscheidende Rolle spielen. Es muss also eine konkrete Modellierung von $p(\mathtt{x})$ vorgenommen werden.

3. LINEAR VERKETTETE CONDITIONAL RANDOM FIELDS

Die linear-verketteten Conditional Random Fields (im folgenden kurz CRFs) sind grundsätzlich den HMM sehr ähnlich. Sie unterscheiden sich eigentlich nur darin, das HMM generativ und CRFs diskriminativ sind [6]. Generative Modelle basieren auf einer gemeinsamen Wahrscheinlichkeitsverteilung $p(\mathtt{y}, x)$, diskriminative Modelle auf einer bedingten Wahrscheinlichkeitsverteilung $p(\mathtt{y}|x)$. Es wird deutlich, dass sich, wie schon bei den HMM, auch hier die wesentlichen Eigenschaften bereits im Namen finden. Die Zustandssequenzen werden also nicht durch das Produkt der Einzelwahrscheinlichkeiten gebildet, sondern durch Aggregation der bedingten Wahrscheinlichkeiten.

Ein Conditional Random Field ist eine bedingte Wahrscheinlichkeitsverteilung $p(\mathtt{y}|\mathtt{x})$ verknüpft mit einem Graphen. Durch die bedingte Wahrscheinlichkeit müssen Abhängigkeiten zwischen den Beobachtungen $\mathtt{x}$ nicht explizit modelliert werden, was ein Ausnutzen von aufwendigen und weitverzweigten Abhängigkeiten ohne unberechenbar zu werden möglich macht. Die bedingte Wahrscheinlichkeit $p(\mathtt{y}|\mathtt{x})$ beinhaltet kein Modell von $p(\mathtt{x})$.

Dies führt zu der Definition von Charles Sutton und Andrew McCallum [6].

Definition 3.1. *Es seien $\mathtt{y}$ und $\mathtt{x}$ zufällige Sequenzen, $\theta = \{\lambda_k\} \in \mathbb{R}^K$ der Parametervektor und $\{f_k(y, y', \boldsymbol{x}_t)\}_{k=1}^{K}$ eine Menge von kontinuierlichen Featurefunktionen. Dann ist ein lineare-verkettetes Conditional Random Field eine Verteilung $p(\boldsymbol{y}|\boldsymbol{x})$ der Form*

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(\boldsymbol{y}_t, \boldsymbol{y}_{t-1}, \boldsymbol{x}_t)\right), \tag{3}$$

mit $Z(\boldsymbol{x})$ als einer instanzspezifische Normalisierungsfunktion

$$Z(\boldsymbol{x}) = \sum_{y} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(\boldsymbol{y}_t, \boldsymbol{y}_{t-1}, \boldsymbol{x}_t)\right). \tag{4}$$

$Z(\mathtt{x})$ zu bestimmen ist sehr aufwendig und praktisch häufig nicht durchführbar. Daher wird hier häufig auf eine Approximation zurückgegriffen [6]. Der Parametervektor wurde als Transitionsmatrix bei HMM erklärt. Alle angelernten

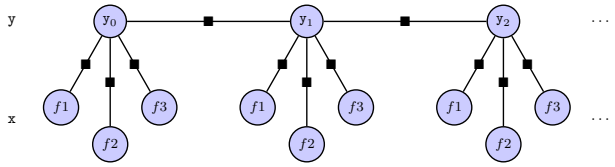

Abbildung 3: Graphisches Modell eines HMM-ähnlichen linear-verketteten CRF [6]

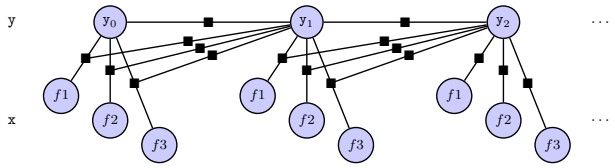

Abbildung 4: Graphisches Modell eines linear-verketteten CRF bei denen die Übergangswahrscheinlichkeiten nur von der aktuellen Beobachtung abhängen [6].

Transitionswahrscheinlichkeiten werden im Parametervektor angegeben.

Das Modell ist also eine Familie von Wahrscheinlichkeitsverteilungen, die anhand eines Graphen faktorisiert werden. Faktorisierungen werden durch Funktionen der Form

$$\Psi_A(\mathtt{x}_A, \mathtt{y}_A) = \exp\left(\sum_k \theta_{Ak} f_{Ak}(\mathtt{x}_A, \mathtt{y}_A)\right) \tag{5}$$

angegeben, wobei $\mathtt{x}_A$ eine A-elementige Teilsequenz von $\mathtt{x}$ ist und analog auch $\mathtt{y}_A$ eine A-elementige Teilsequenz von $\mathtt{y}$. Diese Faktorisierungen werden in Graphen durch schwarze Boxen angezeigt. Die Grundidee ist eine Verteilung über eine große Anzahl an zufälligen Variablen zu schaffen, indem ein Produkt nur über eine kleine Teilmenge der Variablen gebildet wird. In Abbildung 3 ist die graphische Repräsentation eines CRFs zu sehen, dass ein HMM nachbildet.

Wie auch in dem Beispiel aus Kapitel 2 ist es häufig der Fall, dass HMM sich fast ausschließlich auf die Identität des Wortes verlassen. Selbst wenn unterschiedlichste Featurefunktionen genutzt werden, so sind sie durch die Einschränkungen häufig auf die Identität des Wortes zurückzuführen. Gerade aber Namen sind teilweise sehr selten und tauchen sie nicht in dem Trainingsdaten auf, so sind solche Feature irrelevant. Um noch nicht gesehene Wörter zu erkennen, müssten andere Eigenschaften ausgenutzt werden, wie z.B. Groß- und Kleinschreibung, Vorgänger und Nachfolger oder die Position im Text. Viele Eigenschaften sind möglich, sogar das Einbinden eines Wörterbuchs von Namen. Im Gegensatz zum HMM lässt sich das CRF aus Abbildung 3 einfach erweitern, sodass jede Featurefunktion von Beobachtungen von irgendeinem Zeitpunkt profitieren kann. Es können also zum Zeitpunkt t nicht nur alle Beobachtungen von $\mathtt{x}_t$ benutzt werden (vgl. Abbildung 4), sondern auch zukünftige Beobachtungen $\mathtt{x}_{t+1}$. Ein solches CRF wird häufig für Arbeiten in Texten, wie z.B bei der NER genutzt.

Es muss jedoch immer berücksichtigt werden, dass zwar die Qualität gesteigert werden kann, zu komplexe Abhängigkeiten jedoch die Berechenbarkeit beeinflussen. So könnte man modellieren wollen, dass alle Wörter in einem gesamten Text Einfluss auf die Label eines Wortes haben. Da dies die Berechenbarkeit stark beeinflusst, ein solches Feature aber durchaus die Qualität steigern kann, wird darauf noch einmal gesondert am Ende dieses Kapitels eingegangen.

Lernen

Um ein Modell zu generieren, müssen auch hier wieder die Parameter $\theta = \{\lambda_k\}$ bestimmt werden. Die Trainingsdaten, seien gegeben durch $\mathcal{D} = \{\mathtt{x}^{(i)}, \mathtt{y}^{(i)}\}_{i=1}^N$, wobei jedes $\mathtt{x}^{(i)} = \{\mathtt{x}_1^{(i)}, \mathtt{x}_2^{(i)}, \ldots, \mathtt{x}_T^{(i)}\}$ eine Sequenz von Beobachtungen und jedes $\mathtt{y}^{(i)} = \{\mathtt{y}_1^{(i)}, \mathtt{y}_2^{(i)}, \ldots, \mathtt{y}_T^{(i)}\}$ eine Sequenz der korrekten Label ist. Da es sich um bedingte Wahrscheinlichkeiten handelt, wird eine modifizierte log-likelihood maximiert, die sogenannte *conditional log likelihood* [6]. Die *conditional likelihood* $p(\mathtt{y}|\mathtt{x};\theta)$ lässt sich über die gemeinsame Wahrscheinlichkeitsverteilung herleiten. Kombiniert man die *conditional likelihood* mit einer beliebigen initialen Wahrscheinlichkeit $p(\mathtt{x};\theta')$ zu einer gemeinsamen Verteilung $p(\mathtt{y},\mathtt{x})$, so ergibt sich die *joint likelihood*

$$\log p(\mathtt{y},\mathtt{x}) = \log p(\mathtt{y},\mathtt{x};\theta) + \log p(\mathtt{x};\theta'). \tag{6}$$

Die beiden Summanden sind voneinander unabhängig, da die Parameter θ' keinen Einfluss auf die Optimierung von θ haben. Daher kann der zweite Summand weggelassen werden. Es gilt also die *conditional log likelihood*

$$\log \mathcal{L}(\theta) = \log \prod_{i=1}^{N} p(\mathtt{y}^{(i)}|\mathtt{x}^{(i)}) \tag{7}$$

zu maximieren [6].
Um sogenanntes *overfitting* zu vermeiden, das immer dann auftreten kann, wenn jede einzelne Beobachtung in das Modell aufgenommen wird, auch wenn sie keinen positiven Nutzen für das Modell hat, wird noch ein Regulierungsparameter $1/2\sigma^2$ eingeführt. Für einen tiefer gehenden Einstieg die die Optimierungsschritte wird auf [6] verwiesen.

Skip-Chain CRFs

In diesem Abschnitt möchte ich einen kurzen Einblick geben, in wie weit etablierte Modelle wie die CRFs erweitert und angepasst werden können. Die bisher beschrieben linear-verketteten CRFs benutzen bei der NER in unstrukturiertem Text häufig ein sogenanntes n-gram Feature. Dabei werden die unmittelbaren Nachbarn mit maximaler Entfernung n für die Bestimmung des Labels genutzt. Erweitert man diese Nachbarschaft unbegrenzt, so wird eine Bestimmung unberechenbar [6]. Um dennoch ausnutzen zu können, dass eine bestimmte Entität innerhalb eines Textes mehrfach in unterschiedlichem Kontext vorkommt, wurden in [6] die Skip-Chain CRFs vorgeschlagen. Die grundsätzliche Funktionsweise kann sehr gut aus Abbildung 5 abgeleitet werden. Wird eine gleiche Entität im Text wiedererkannt (String Vergleich), so können alle Informationen ausgetauscht und für eine bessere Klassifizierung genutzt werden. Ein solches Feature lässt sich verhältnismäßig leicht auf CRFs anwenden, nicht jedoch auf generative Modelle wie die HMM. Dies ist wieder darauf zurückzuführen, dass die HMM $p(\mathtt{x})$ explizit modellieren müssen.

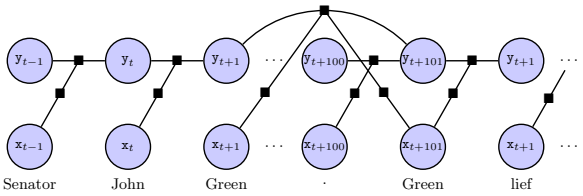

Abbildung 5: Graphisches Modell eines skip-chain CRF. Identische Wörter werden verbunden, da sie wahrscheinlich das gleiche Label bekommen [6].

4. VERGLEICH

Nachdem nun ein kleiner Einblick in die beiden Modelle gewonnen wurde und an einigen Stellen schon Vergleiche gezogen wurden, soll in diesem Kapitel die Gegenüberstellung für diese Arbeit komplettiert werden. Das Hidden Markov Model auf der einen Seite als Repräsentant für generative Modelle, die Conditional Random Fields für die diskriminativen Modelle auf der anderen.

Der offensichtlichste und ausschlaggebendste Unterschied ist der, dass die HMM eine gemeinsame Wahrscheinlichkeitsverteilung über Abhängigkeiten zwischen Zuständen $\mathtt{y}$ und Beobachtungen $\mathtt{x}$ angeben, die CRFs hingegen eine bedingte Wahrscheinlichkeitsverteilung [6].

Bei einem HMM ist die zugrunde liegende Annahme, dass jede Beobachtung von einem versteckten Zustand generiert wurde. Um die wahrscheinlichste Erklärung (Most Likely Explanation) herauszufinden, wird für jede Beobachtung die Wahrscheinlichkeit ermittelt, von einem bestimmten Zustand generiert worden zu sein. Dazu generiert das HMM alle möglichen Beobachtungssequenzen. Werden vielen interagierenden Featurefunktionen oder weit verteilte Abhängigkeiten innerhalb der Eingabe verwendet, so führt dies zu praktischer Unberechenbarkeit [2]. Die Berechnung der wahrscheinlichsten Erklärung $\mathtt{y}*$ erfolgt durch

$$\mathtt{y}* = arg\ max_{\mathtt{y}}\ p(\mathtt{y}|\mathtt{x}). \tag{8}$$

Hierbei wird das sogenannte Inferenz Problem gelöst. Die Lösung dieses Problems ist von entscheidender Bedeutung, da es nicht nur für die wahrscheinlichste Erklärung gelöst werden muss, sondern auch beim Lernen. Allerdings ist das Inferenz Problem NP-schwer, weshalb spezielle Algorithmen wie der Baum-Welch Algorithmus benutzt werden, der auf dem Konzept der dynamischen Programmierung basiert. Wie bereits in Kapitel 2 dargestellt, erfolgt das Lernen durch Bildung der *maximum log likelihood* über

$$\log \mathcal{L}(\theta) = \log \prod_{i=1}^{N} p(\mathtt{x}^{(i)}; \theta)$$

Dabei basiert dies auf der expliziten Modellierung von $p(\mathtt{x})$. Deshalb müssen in HMM strenge Unabhängigkeitsbedingen unter den Beobachtungen $\mathtt{x}$ getroffen werden, z.B. dass jeder Zustandswechsel ausschließlich von der aktuellen Beobachtung und dem letzten Zustand abhängt. Dies macht es jedoch schwieriger aufgrund eines neuen Kontextes eine Beobachtung neu zu klassifizieren. Die Relevanz eines solchen Features wird besonders deutlich mit Hinblick auf Beispiel 2.2. In einem Satz ist *Blume* sowohl der Name einer `Person`, als auch eine Pflanze (`Miscellaneous`).

Die Parameter können mit dem Expectation-Maximization Algorithmus (EM-Algorithmus) optimiert werden. Dabei wird in jedem E-Schritt die Erwartung maximiert und in jedem M-Schritt das Modell (Smoothing). Beide Schritte müssen dabei das Inferenz Problem lösen. Die CRFs hingegen maximieren die sogenannte *conditional log likelihood* über

$$\mathcal{L}(\theta) = \log \prod_{i=1}^{N} p(\mathtt{y}^{(i)}|\mathtt{x}^{(i)}).$$

Hier wird keine explizite Modellierung von $p(\mathtt{x})$ benötigt.

Mit dem Baum-Welch Algorithmus kann die Gleichung 8 unabhängig von dem verwendeten Modell gelöst werden. Der Baum-Welch Algorithmus wird auch als Forward-Backward-Algorithmus oder im Kontext von HMM als EM-Algorithmus bezeichnet [12]. Die Laufzeit beider Optimierungen liegt so in $\mathcal{O}(TK^2NG)$ [6], wobei T die Länge der Sequenzen, K die Anzahl der verschiedenen Zustände, N die Anzahl der Trainingsdaten und G die Anzahl berechneten Gradienten ist. Die Anzahl der verschiedenen Zustände K gehen quadratisch in die Laufzeit ein, weshalb bei beiden Modellen darauf zu achten ist, nicht zu viele Label zu erlauben. Die Laufzeit steigt mit der Anzahl der Gradienten G linear. Bei genügend komplexen Abhängigkeiten bei den Featurefunktionen kann die Laufzeit also erheblich verschlechtert werden. Hier liegt der entscheidende Unterschied zwischen den HMM und den CRFs, der sich sowohl auf das Lernen, also auch auf das Finden der wahrscheinlichsten Erklärung auswirkt. Dies kann im Extremfall dazu führen, dass das gesamte Klassifizierungsproblem NER mit HMM praktisch nicht mehr zu lösen ist.

Ein diskriminatives Modell wie die CRFs hingegen spezifiziert die bedingten Wahrscheinlichkeiten für mögliche Sequenzen von Labeln bei gegebener Sequenz von Beobachtungen. Daher ist kein zusätzlicher Aufwand nötig, um die Beobachtungen zu modellieren. Das Modell $p(\mathtt{y}|\mathtt{x})$ beinhaltet also kein Modell $p(\mathtt{x})$. Bei der Erstellung von Featurefunktionen sind also weniger Grenzen gesetzt, es können nahezu beliebige Abhängigkeiten innerhalb der Beobachtungen ausgenutzt werden. Auch ist es möglich Features mit unterschiedlichem Detailgrad auszunutzen (Tabelle → Wörter → Zeichen) oder zusammenfassende Features wie das Text Layout oder erkannte Label von zukünftigen Beobachtungen zu benutzen.
Die fehlende Modellierung von $p(\mathtt{x})$ wird als Grund dafür vermutet, dass CRFs robuster gegenüber Verletzungen von Unabhängigkeitsbedingungen sind [2]. Die Annahmen über die Unabhängigkeit von Variablen wird bei CRFs nicht über die Beobachtungen $\mathtt{x}$ getroffen, sondern über die Label $\mathtt{y}$. Dadurch haben die CRFs mehr Freiheiten, sich an die Daten anzupassen [6].

Trotz dieser Einschränkungen können mit den HMM qualitativ hochwertige Ergebnisse erreicht werden, die sich durch experimentelle Analysen zeigen lassen. In Kapitel 5 werden hierzu einige ausgewählte Analysen präsentiert. Allerdings sind häufig die Wahrscheinlichkeitsbestimmungen schlecht. Die Begründung hierfür liegt im Baysschen Netz. Wenn beim Training sich häufig wiederholende Daten vorkommen,

so wird das *confidence level* für die Wahrscheinlichkeitsbestimmung hoch gesetzt, obwohl keine neuen Informationen hinzugefügt wurden. Besonders problematisch wird es bei der Erweiterung zu den sequentiellen HMM, da bei der Inferenzbestimmung viele Teile des Modells miteinander kombiniert werden. Ist ein Teil *overconfident*, so wird es schwer ihn sinnvoll in das gesamte Modell zu integrieren [6].

Es gibt außerdem noch ein weiteres Problem, welches von John Lafferty, Andrew McCallum und Fernando Pereira als das *Label Bias Problem* beschrieben wird [2]. In einem generativen Modell werden nur alle ausgehenden Transitionen eines Zustandes verglichen, z.B. $A \xrightarrow{0.7} B$ und $A \xrightarrow{0.3} C$. Die Entscheidung hängt nur vom aktuellen Zustand und von der aktuellen Beobachtung ab. Dies kann Ergebnisse beeinflussen, da, wenn ein Zustand beispielsweise nur eine ausgehende Transition hat, die Beobachtung prinzipiell ignoriert wird. Es wird unabhängig von der vom Modell berechneten Wahrscheinlichkeit für diese Beobachtung immer die gleiche Transition gewählt. Allgemein gilt, je weniger ausgehende Transitionen ein Zustand hat, desto weniger Einfluss haben Beobachtungen auf die Entscheidung. Bei einer geringen Abweichung von einer Gleichverteilung der Trainingsdaten, kann dies dazu führen, dass immer eine Entscheidung bevorzugt wird, was zu einer Verschlechterung der Qualität führt.

Mit diskriminativen Modellen ist es möglich, in den Übergangswahrscheinlichkeiten alle ausgehenden Transitionen aller Zustände zu berücksichtigen und auch eine Art Eigenbewertung (mass) mitzugeben. Ein CRF hat ein einziges exponentielles Modell für die Wahrscheinlichkeit der gesamten Sequenz von Labeln bei gegebener Sequenz von Beobachtungen. Daher können alle Gewichte (Übergangswahrscheinlichkeiten) gegeneinander aufgerechnet werden und Transitionen bewertet werden (vgl. auch Gleichung 4).

5. ANWENDUNGSFÄLLE

In diesem Kapitel werden einige Anwendungsfälle von HMM und CRFs aufgezeigt und verschiedene experimentelle Analysen verglichen, um Ansätze mit HMM und CRFs direkt vergleichen zu können. Dazu werden häufig drei verschiedene Angaben zur Qualität gemacht:

1. Genauigkeit (Precision **P**): Prozentuales Verhältnis zwischen korrekt Erkannten und allen Erkannten. (Label A wurde in 7 von 15 Fällen korrekt gesetzt)
2. Trefferquote (Recall **R**): Prozentuales Verhältnis zwischen den korrekt Erkannten und allen möglichen korrekten. (Label A wurde in 7 von 10 Fällen gesetzt)
3. F-Maß (F-score **F**): Gewichtetes Mittel aus Trefferquote und Genauigkeit. ($F = \frac{P \cdot R}{P+R}$)

Eine nicht nur im Kontext von Semantic Web immer beliebter werdende Disziplin ist die der Informationsextraktion und zwar möglichst automatisch oder semiautomatisch. Eine Teildisziplin ist dabei die NER. Dabei gilt es in einem gegebenen Text bestimmte Entitäten aufzuspüren und mit einem korrekten Label zu versehen.

Tabelle 1: Biomedizinische NER

	Quelle	P	R	F
CRF	[13]	71.1	66.4	68.7
	[14]	72.8	69.1	70.5
	[15]	85.09	79.06	81.96
	∅	76.33	71.52	73.72
HMM	[13]	62.4	69.4	65.7
	[16]	62.98	69.41	66.04
	∅	62.69	69.41	65.87
CRF/HMM	∅	1.22	1.03	1.12

Biomedizinische NER

Natalia Ponomareva e.a. haben die Aufgabe der NER in der Molekular Biologie experimentell sowohl mit HMM als auch mit CRFs gelöst [13]. Durch die experimentelle Analyse kamen sie zu dem Schluss, dass HMM im Vergleich zu den CRFs eher dazu tendieren ein Label trotz einer möglichen Fehleinschätzung zu setzen. Daher ist die Trefferquote der HMM höher, als bei den CRFs, die Genauigkeit leidet jedoch darunter. Daher erzielte die CRF Implementierung ein höheres F-Maß, das jedoch mit 68.7 gegenüber 65.7 natürlich höher, jedoch immer noch ungenügend für eine automatische Erkennung ist.

Vergleichbare Tests wurden von Settles [14] und von Leaman and Gonzalez [15] ausgeführt. In beiden Artikeln wurde jedoch der Fokus auf eine CRF Implementierung gelegt, ohne eine vergleichende HMM Implementierung, da die Autoren sich keine besseren Ergebnisse versprachen. Für alle Tests wurden Artikel aus der Biomedizin benutzt, jedoch nicht immer der exakt gleiche Korpus. Die möglichen Label waren jedoch bei allen Tests identisch. Auch in ihren Ergebnissen spiegelt sich wieder, dass CRF Implementierungen im Bereich der biomedizinischen NER eine höhere Genauigkeit als Trefferquote aufweisen können. Eine reine HMM Implementierung von Zhao und Shaojun [16] aus der Schweiz, die ebenfalls für diesen Anwendungsfall entwickelt wurde, erzielte ziemlich genau die gleichen Ergebnisse bei Genauigkeit, Trefferquote und F-Maß wie bei [13]. Die genauen Ergebnisse sind in Tabelle 1 abgebildet. Anhand von Tabelle 1 ist ebenfalls ersichtlich, dass alle gelisteten CRF Implementierungen bei Genauigkeit und im F-Maß besser abgeschnitten haben, als die HMM Implementierungen. Durchschnittlich konnte ein ca. 12% höheres F-Maß erzielt werden.

Informationsextraktion aus Tabellen

Liegen die Informationen schon in einer einigermaßen strukturierten Form vor (Semi Structured Data), so können diese zusätzlichen Informationen genutzt werden. David Pinto e.a. haben mit Hilfe von CRFs und HMM Informationen, die in Tabellenform vorliegen, verarbeitet [4].

Sie konnten beobachten, dass das Rauschen (Noise) von CRFs besser abgefangen wurde als von dem HMM. Rauschen bedeutet hier, dass leere Tabellenreihen von den HMM fälschlich für Informationen gehalten wurden, während CRFs sie korrekter Weise ignorierten.

Informationsextraktion aus Tabellen benötigt viele komplexe Feature, die voneinander abhängig sind und

vom Detailgrad verschiedene Eigenschaften kombinieren müssen. Dazu gehört der Mix aus sprachabhängigen und formatabhängigen Features. Wie in Kapitel 4 bereits dargelegt wurde, ist dies ein entscheidender Vorteil von CRFs. Ihr Ansatz war es, jede Zeile einer Tabelle zu finden und mit den entsprechenden Beschriftungen zu verknüpfen. Dabei zeigte ihre CRF Implementierung deutlich bessere Ergebnisse als ihre HMM Entsprechung. Während die HMM Variante nur ein F-Maß von 65.4 erreicht, steigert die CRF Implementierung dieses Ergebnis um ca. 42% auf 91.8.

Informationsextraktion aus Sozialen Netzen

Auch Informationsgewinnung in sozialen Netzwerken wird immer wichtiger. In der Universität für Elektro-Kommunikation in Tokyo wurde ein Notfallsystem für Erdbebenopfer gebaut. Dabei werden alle menschlichen Aktivitäten auf Twitter und in Blogs während und nach dem Erdbeben aufgezeichnet und die wichtigen Informationen mittels CRFs extrahiert, um eine Aktivitätenübersicht zu erstellen und so anderen Menschen sichere Fluchtwege etc. bereitzustellen [5]. Aufgrund von vorherigen Ergebnissen in vergleichbaren Aufgaben, bei denen sich CRFs als sehr zuverlässig und mit guten Ergebnissen hinsichtlich F-Maß herausgestellt haben, wurde in [5] kein Vergleich zu einer HMM Variante vorgenommen. In ihrem Experiment wurden aus Twitter- und Blogposts `Aktivitäten`, `Aktoren`, `Aktionen`, `Objekte`, `Zeiten` und `Orte` extrahiert. Insgesamt konnte in den Tests ein durchschnittliches F-Maß von 81.05 erreicht werden. Auch in diesem Anwendungsfall lag, wenn auch nur um knapp 4%, die Genauigkeit über der Trefferquote.

Informationsgewinnung durch probabilistische Crawler

Ein etwas anders gelagerter Anwendungsfall ist der der Informationsgewinnung. Während die bisher untersuchten Aufgaben sich damit befasst haben vorhandene Daten zu verarbeiten, liegt hier das Ziel darin, die Daten zu beschaffen. Mit Hilfe eines HMM basierten Crawlers sollen möglichst schnell zu einem Suchbegriff relevante Seiten gefunden werden [3]. In diesem Kontext bedeutet schnell, dass wenige Seiten besucht werden müssen, bis fast ausschließlich nur noch für ausreichend relevant befundene Seiten geliefert werden. Den HMM basierten Ansatz haben sie mit einem Best-First Crawler verglichen. Während der Best-First Crawler immer die maximal relevante Seite als nächsten Ausgangspunkt nimmt, nutzt der HMM Crawler inhaltlich irrelevante Seiten, die aber vielversprechend sind einen Link zu einer relevanten Seite zu beinhalten. Eine solche Abhängigkeit zwischen Seiten mit HMM zu modellieren führte schon zu sehr guten Ergebnissen. Da mit CRFs noch komplexere Abhängigkeiten modelliert werden können, erhoffen sich die Autoren mit einer geplanten CRFs Variante noch bessere Ergebnisse.

5.1 Zusammenfassung

In allen hier betrachteten Anwendungsfällen wird deutlich, dass beide Methoden durchaus ihre Berechtigung haben. Es scheint so, dass in der Praxis jedoch HMM eher dazu tendieren ein Label zu setzen, auch wenn es ein falsches Label sein sollte. Bei einer CRF Implementierung ist daher die Genauigkeit häufig höher, da sie vorsichtiger sind. So verliert man wichtige Treffer. Besonders bei der NER im biomedizinischen Kontext zeigen die Testergebnisse, dass zum jetzigen Zeitpunkt noch keine vollautomatische Erkennung möglich ist. Es muss sich erst noch zeigen, ob dies überhaupt möglich sein wird, oder ob sich die Einsatzmöglichkeit hier auf semi-automatische Unterstützung beschränkt. Die beste CRF Implementierung ist mit einem etwa 24% höherem F-Maß deutlich besser als die beste HMM Implementierung. Da beide analysierten HMM Varianten nahezu identische Ergebnisse erzielten, ist hier vermutlich weniger Optimierungspotential vorhanden. Alle Varianten sind jedoch mit ihren Ergebnissen nicht für die automatische NER geeignet. Bei einem semi-automatischen Einsatz müsste noch ermittelt werden, mit welchem Modell Menschen besser unterstützt werden können. Zu klären ist, ob es hilfreicher ist, wenn mehr gefundenen Entitäten bereits mit einem korrekten Label versehen wurden, oder wenn mehr Entitäten gefunden werden, deren Label aber noch häufiger verändert werden muss. Hier könnte sich abzeichnen, dass obwohl das F-Maß der CRF Varianten konstant besser war, der gesamte Arbeitsaufwand inklusive dem menschlichen Faktor durch eine HMM Unterstützung schneller und besser wird. Bei der Ausführung von komplexeren Aufgaben als der NER auf unstrukturierten Texten, z.B. der Informationsextraktion aus semi-strukturierten Daten, konnte das Experiment jedoch deutliche Vorteile der CRFs ausmachen. Mit einer Steigerung des F-Maßes um etwa 42% auf über 90 erscheint eine automatische Datenverarbeitung nicht ausgeschlossen. Diese Überlegenheit ist sehr wahrscheinlich auf die grundlegenden Unterschiede zwischen den Modellen zurückzuführen. Die Tatsache, dass einige Forscher [14] aufgrund anderer Ergebnisse eine HMM Variante nicht mehr in Betracht ziehen, zeigt außerdem zweifelsfrei den Trend hin zu CRFs.

6. FAZIT

Wie in mehreren Arbeiten [6, 9, 2, 17] bestätigt wird, eigenen sich CRFs deutlich besser für den Umgang mit komplexen und vielschichtigen Abhängigkeiten zwischen den Beobachtungen. Beim Trainieren von HMM muss vor allem darauf geachtet werden, dass die Daten gleichmäßig verteilt, also ohne hohe Redundanz sind. Dies führt sonst zu *overconfidence* eines Features und kann zu schlechteren Ergebnissen führen. Beide Modelle sind limitiert durch die Anzahl an verschiedenen Klassen von Zuständen, die quadratisch in die Laufzeit eingehen. Komplexe Abhängigkeiten innerhalb der Beobachtungen können in CRFs schneller verarbeitet werden. Dies führt zu mehr Freiheit in der Gestaltung der Modelle. Es hat sich außerdem gezeigt, dass HMM eher dazu tendieren eine Entscheidung zu treffen als die CRFs. Dies zeigt sich in der Genauigkeit und der Trefferquote. In einem semi-automatischen Kontext kann die Trefferquote ausschlaggebend für einen Erfolg sein, bei Daten mit viel Rauschen aber eher störend.

Es zeigt sich, dass der Trend zu den CRFs begründet ist, jedoch nur wenn die Aufgaben hinreichend komplex sind, lohnt sich auch ein Ansatz mit CRFs. Existierende Lösungen für simplere Aufgaben wie die NER, die hinreichend gute Ergebnisse liefern, werden diese vermutlich durch eine CRFs Variante nicht wesentlich verbessert werden können. Da

ein nahezu identischer Nachbau eines HMM durch CRFs möglich ist, kann es jedoch möglicherweise lohnenswert sein, in Zukunft CRFs zur Problemlösung heranzuziehen. Dies wird auch dadurch bestätigt, dass bei allen untersuchten experimentellen Analysen keine CRFs Implementierung gefunden wurde, die schlechter Abschnitt, als eine vergleichbare HMM.

Literatur

[1] M. I. Jordan und T. M. Mitchell. "Machine learning: Trends, perspectives, and prospects". In: *Science* 349.6245 (2015), 255–260. URL: http://www.sciencemag.org/content/349/6245/255.full.pdf.

[2] John D. Lafferty, Andrew McCallum und Fernando C. N. Pereira. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, S. 282–289. ISBN: 1-55860-778-1. URL: http://dl.acm.org/citation.cfm?id=645530.655813.

[3] Hongyu Liu, Evangelos Milios und Jeannette Janssen. "Probabilistic Models for Focused Web Crawling". In: *Proceedings of the 6th Annual ACM International Workshop on Web Information and Data Management*. WIDM '04. Washington DC, USA: ACM, 2004, S. 16–22. ISBN: 1-58113-978-0. DOI: 10.1145/1031453.1031458. URL: http://doi.acm.org/10.1145/1031453.1031458.

[4] David Pinto u. a. "Table Extraction Using Conditional Random Fields". In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. SIGIR '03. Toronto, Canada: ACM, 2003, S. 235–242. ISBN: 1-58113-646-3. DOI: 10.1145/860435.860479. URL: http://doi.acm.org/10.1145/860435.860479.

[5] The-Minh Nguyen e.a. "Building Earthquake Semantic Network by Mining Human Activity from Twitter". In: *Proceedings of the 2011 IEEE International Conference on Granular Computing*. Kaohsiung: IEEE, 2011, S. 496–501. ISBN: 978-1-4577-0372-0. DOI: 10.1109/GRC.2011.6122647. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6122647.

[6] Charles Sutton und Andrew McCallum. "Introduction to Conditional Random Fields for Relational Learning". In: *Introduction to Statistical Relational Learning*. Hrsg. von Lise Getoor und Ben Taskar. MIT Press, 2006.

[7] Charles Sutton und Andrew McCallum. "An Introduction to Conditional Random Fields". In: *Foundations and Trends in Machine Learning* 4.4 (2012), 267–373.

[8] Zoubin Ghahramani. "Hidden Markov Models". In: River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2002. Kap. An Introduction to Hidden Markov Models and Bayesian Networks, S. 9–42. ISBN: 981-02-4564-5. URL: http://dl.acm.org/citation.cfm?id=505741.505743.

[9] Roman Klinger und Katrin Tomanek. *Classical Probabilistic Models and Conditional Random Fields*. Techn. Ber. TR07-2-013. ISSN 1864-4503. Department of Computer Science, Dortmund University of Technology, 2007. URL: http://www.scai.fraunhofer.de/fileadmin/images/bio/data_mining/paper/crf_klinger_tomanek.pdf.

[10] Stuart J. Russell und Peter Norvig. *Artificial Intelligence: A Modern Approach*. 2. Aufl. Pearson Education, 2003. ISBN: 0137903952.

[11] Dr. Thomas Gottron. *Hidden Markov Models*. University Lecture. 2014.

[12] Jeff Bilmes. *A Gentle Tutorial of the EM algorithm and its application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*. Techn. Ber. TR-97-021. ICSI, 1997.

[13] Ferran Pla Natalia Ponomareva Paolo Rosso und Antonio Molina. "Conditional Random Fields vs. Hidden Markov Models in a biomedical Named Entity Recognition task". In: *Proceedings of the RANLP'07 conference*. 2007.

[14] B. Settles. "ABNER: An open source tool for automatically tagging genes, proteins, and other entity names in text". In: *Bioinformatics* 21.14 (2005), S. 3191–3192.

[15] Graciela Gonzalez Robert Leaman. "BANNER: An Executable Survey of Advances in Biomedical Named Entity Recognition". In: *Pacific Symposium on Biocomputing*. 2008, S. 652–663.

[16] Shaojun Zhao. "Named Entity Recognition in Biomedical Texts Using an HMM Model". In: *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*. JNLPBA '04. Geneva, Switzerland: Association for Computational Linguistics, 2004, S. 84–87. URL: http://dl.acm.org/citation.cfm?id=1567594.1567613.

[17] Andreas Stolcke Yang Liu Elizabeth Shriberg und Mary Harper. "Comparing HMM, maximum entropy, and conditional random fields for disfluency detection". In: *In Proceeedings of the European Conference on Speech Communication and Technology*. 2005.