Proceedings

Tagung der Computerlinguistik-Studierenden 2005

(TaCoS '05)

03.–05. Juni 2005

Universität Stuttgart
Institut für Maschinelle Sprachverarbeitung

Inhaltsverzeichnis

Vorwort

Liebe Leserin, lieber Leser,

wir freuen uns sehr, Ihnen diesen Vortragsband anlässlich der 15. Tagung der Computerlinguistikstudierenden (TaCoS) vorlegen zu können. Die TaCoS ist eine jährlich stattfindende Tagung von und für Studierende der Computerlinguistik und verwandter Felder. Sie bietet die Möglichkeit der Präsentation von wissenschaftlichen Arbeiten, die während oder kurz nach dem Studium entstanden sind, und gibt ferner die Gelegenheit, Kontakte zu Studierenden anderer Universitäten und zu Vertretern der fachnahen Industrie zu knüpfen.

Im Jahre 2005 fand die TaCoS zum dritten Mal in Stuttgart statt. Es war uns, dem Organisationsteam, ein Vergnügen die Veranstaltung vorzubereiten und nicht zuletzt mitzuerleben. Es gab eine Vielzahl interessanter Vorträge, die unserer Meinung nach nicht in der Versenkung verschwinden sollten. Aus diesem Grund haben wir den Plan gefasst, den Vortragenden die Veröffentlichung ihrer vorgestellten Arbeiten zu ermöglichen. Für die erhebliche Verspätung bei der Umsetzung dieses Vorsatzes möchten wir uns bei allen Autoren, die einen Beitrag zu dieser Veröffentlichung geleistet haben, vielmals entschuldigen.

Der Inhalt dieses Bandes ist, wie auch die Themengebiete der 15. TaCoS, weit gefächert. Es gab Vorträge aus den Bereichen der theoretischen und angewandten Computerlinguistik, der allgemeinen Linguistik und Sprachwissenschaften sowie Ausblicke auf potenzielle Arbeitsfelder auch außerhalb der Universitäten. Durch das vielfältige Abendprogramm mit dem Stadtrundgang, dem gemeinsamen Abendessen und dem traditionellen Grillabend kam auch das Zwischenmenschliche nicht zu kurz.

Abschließend möchten wir allen Teilnehmern, Vortragenden, der Vielzahl ungenannter Helfer im Hintergrund und den Sponsoren, der Trados GmbH und der Gesellschaft für linguistische Datenverarbeitung (GLDV), danken, die alle zum gelungenen Ablauf der Tagung beigetragen haben.

Vielen Dank auch an Sabine Schulte im Walde, die in der Vergangenheit schon eine Stuttgarter TaCoS mitorganisiert hat und bereit war, ein Grußwort zu verfassen.

Wir wünschen viel Spaß bei der Lektüre und hoffen dass sich die TaCoS auch in Zukunft so großer Beliebtheit erfreuen wird.

Die Organisatoren der TaCoS 2005:
Tolga Ergin, Fabienne Fritzinger, Silvana Hartmann,
Ralf Jankowitsch, Andreas Madsack & Alexander Valet

Stuttgart, 03. Juni 2007

Grußwort

Man kann es sicher als eine Ehre für das IMS betrachten, bereits zum dritten Mal in der Geschichte der TaCoS-Veranstaltungen als Gastgeber zu fungieren. 1992 haben Studenten der Fachschaft Computerlinguistik am IMS die Tagungsreihe ins Leben gerufen, 1996 war ich selbst Teil des Organisationsteams, und im Jahre 2005 gibt es offensichtlich und erfreulicherweise wieder eine Gruppe von Studenten, die das TaCoS-Erlebnis zu schätzen wissen. Die TaCoS stellt einen wichtigen Teil des fachlichen und sozialen Austausches im deutschsprachigen Raum der Computerlinguistik dar: Die Studenten bekommen einen ersten Eindruck vom Konferenzleben, sie tragen über eigene Forschung vor oder diskutieren die Forschung von Kollegen; eingeladene CLer aus dem akademischen Bereich oder aus der Industrie präsentieren Zukunftsmöglichkeiten in unterschiedlichem Umfeld und mit verschiedenen Anwendungsperspektiven; es gibt Potenzial für erste Netzwerke zwischen Instituten verschiedener Universitäten; und die privaten Unterkünfte und sozio-kulturellen Unternehmungen tragen zu persönlichen Kontakten bei. Ich halte die TaCoS für eine sehr wertvolle Einrichtung und freue mich, dass sie nun schon über 14 Jahre lang an vielen verschiedenen Universitäten wertgeschätzt wird, dass die Teilnehmerzahlen nicht gesunken, sondern gestiegen sind, und natürlich dass das IMS offensichtlich immer noch eine tragende Rolle spielt!

Sabine Schulte im Walde

Stuttgart, 07. Dezember 2006

Tagungsprogramm

	Freitag, 03. Juni 2005	**Samstag, 04. Juni 2005**	**Sonntag, 05. Juni 2005**
08:30 – 09:30	**Frühstück**	**Frühstück**	**Frühstück**
09:30 – 10:30	Prof. Hinrich Schütze, Ph.D. [Universität Stuttgart] – **Begrüßungsrede** [de]	Halyna Seniv [Universität Stuttgart] – **Wortbasierte Rechtschreibprüfung mit bedingter Kompositazerlegung** [de]	Marthe Dekker [Universiteit Utrecht] – **Recognizing Textual Entailment using Common Sense Knowledge** [en]
10:30 – 11:30	Augustin Speyer [University of Pennsylvania] – **A Phonological Factor for the Decline in Topicalization in English** [en]	Caroline Clemens, Dr. Sascha Fagel [Technische Universität Berlin] – **Die Nutzung experimentalphonetischer Messdaten zur audiovisuellen Sprachsynthese** [de]	Charlotte Wollermann [Rheinische Friedrich-Wilhelms-Universität Bonn] – **Perzeption der kongruenten Kopfbewegung bei der multimodalen Sprachsynthese** [de]
11:30 – 12:30	Andrea Schuch [Universität des Saarlandes] – **Sprachsteuerung für Fahrkartenautomaten** [de]	Michael Wetzel [TRADOS] – **. . . and does all this theory matter in business?** [en]	Julia Ritz [Universität Stuttgart] – **Ein Named-Entity-Recognition-System fürs Deutsche** [de]
12:30 – 14:30	**Mittagessen**	**Mittagessen**	**Abschlussdiskussion**
14:30 – 15:30	Jörg Didakowski [Universität Potsdam] – **Robustes Parsing und Disambiguierung mit gewichteten Transduktoren** [de]	Michael Kaisser [Universität des Saarlandes] – **Question Answering with Linguistic Methods** [de]	**Heimreise**
15:30 – 16:30	Charles Yee [Universität Stuttgart] – **A Lexical Approach to Presupposition and Meaning** [en]	Michael Poprat [Friedrich-Schiller-Universität Jena] – **Computerlinguistik in Jena: Lehre, Forschung und Perspektiven** [de]	
16:30 – 17:30	Eva Anderl [Ludwig-Maximilians-Universität München] – **Analyse bibliographischer Referenzen** [de]	**Abendveranstaltung**	
ab 18:00	**Abendveranstaltung**		

Robustes Parsing und Disambiguierung von Konstituentensätzen mit gewichteten Transduktoren

Jörg Didakowski
Universität Potsdam
didakowski@ling.uni-potsdam.de

Zusammenfassung

In diesem Aufsatz soll das in Didakowski (2005) entwickelte Verfahren für robustes Parsing von Dependenzstrukturen auf die Verarbeitung von Konstituentensätzen erweitert werden. In Didakowski (2005) werden die linguistischen Theorien des Chunking (Abney, 1990) und syntaktischen Tagging (Karlsson, 1999) mit Hilfe von gewichteten Transduktoren realisiert. Allerdings bleibt dieser Ansatz auf einfache Sätze[1] beschränkt.

1 Parsing von einfachen Sätzen

1.1 Einführung

In Didakowski (2005) wird ein Verfahren vorgestellt, mit dem es möglich ist, einfache Sätze mit gewichteten Transduktoren zu analysieren.

Es werden zwei grundsätzliche linguistische Theorien, das Chunking (Abney, 1990) und das syntaktische Tagging (Karlsson, 1999), als Grundlage für die syntaktische Analyse angenommen. Beide Theorien werden unter der Betrachtung von lokalen Dependenzstrukturen zusammengeführt und angewendet. Die Analyse bewegt sich zwischen einem Low- und High-Level-Parsing.

Mit Hilfe von gewichteten Transduktoren ist es möglich, Dependenzstrukturen über natürlichsprachliche Texte zu erstellen. Die Eingabe selbst muss als ein gewichteter Automat dargestellt werden. Eine Grammatik, die durch gewichtete Transduktoren realisiert ist, kann dann eine Eingabe auf einen gewichteten Automaten abbilden, der die Dependenzstrukturen zu der entsprechenden Eingabe enthält. Hierzu dient die Komposition von gewichteten Transduktoren. Es werden alle möglichen Strukturen generiert, ob sie grammatisch sind oder nicht. Die Strukturen werden durch Klammerungen (Chunks) und Tags angezeigt. Danach wird die beste Lesart durch einen Single-Source-Shortest-Path-Algorithmus ermittelt. Dieser Schritt kann als Disambiguierung angesehen werden.

Das paarweise Vergleichen von Lesarten wird durch eine Reihe von Kriterien realisiert, die als Bewertungssemiring[2] vorliegen und daher leicht ergänzt werden können. Kriterien können nach ihrer Relevanz geordnet werden. So ist es z.B. möglich, bei einer Longest-Match-Strategie, die durch bestimmte Kriterien realisiert ist, einen Garden-Path-Effekt[3] zu unterbinden, indem das Füllen eines Subkategorisierungsrahmens als wichtiger eingestuft wird.

Durch Kriterien ist es dann möglich, Lesarten nach ihrer Relevanz zu ordnen; hierbei soll die Relevanz mit der Grammatikalität syntaktischer Strukturen korrespondieren. Es müssen aber nicht unbedingt grammatische Strukturen generiert werden. Dieses Verfahren generiert alle im Allgemeinen gültigen aber nicht unbedingt grammatischen Strukturen. Die potenziellen Analysen werden paarweise verglichen. Hierbei beruht die Grammatikalität auf dem Prinzip des Vergleiches.

Letztlich ist so eine äußerst robuste und effiziente Verarbeitung möglich.

Hier werden im Folgenden die Notationen von Karttunen (1995) verwendet.

[1] Als einfacher Satz werden hier Sätze bezeichnet, die keine Konstituentensätze enthalten.

[2] Ein additiv idempotenter Semiring $(S, \oplus, \otimes, \bar{0}, \bar{1})$ mit $\bar{0} \neq \bar{1}$, bei dem die Operation $\oplus$ eine lineare Ordnung über S definiert, wird hier als *Bewertungssemiring* bezeichnet.

[3] Bei einem Garden-Path-Effekt kommt es zu Verarbeitungsschwierigkeiten, wenn z.B. ein Subjekt im Satz fehlt, was durch eine Longest-Match-Strategie (late-closure) ausgelöst werden kann (Frazier & Clifton, 1996).

1.2 Die Eingabe

Die Eingabe beinhaltet in Didakowski (2005) am Anfang alle möglichen morphologischen Lesarten eines einfachen Satzes. Hierbei können Wörter kategoriale oder paradigmatische Ambiguitäten enthalten. Es ist also eine morphologische Analyse jedes natürlichsprachlichen Wortes notwendig. Die Eingabe wird ähnlich zum syntaktischen Tagging in Koskenniemi (1990) durch eine Abfolge von natürlichsprachlichen Wörtern mit ihren morphologischen Analysen dargestellt. Kategorien, die nicht durch das Chunking behandelt werden, oder Kategorien, die zur Klasse der verwaisten funktionalen Elemente gehören (vgl. Abney 1990), werden mit potenziellen syntaktischen Funktionen ausgezeichnet. Vor jedem Wort befindet sich das Wortgrenzen-Tag @. Ein Satz wird letztlich durch das Satzgrenzen-Tag @@ umschlossen. Es soll ein Beispiel für den Satz *die Männer lieben Eva* gegeben werden, wobei das STTS[4] und die syntaktischen Tags aus dem ENGCG Tagset (siehe Halteren 1999) verwendet werden:

@@
@die [[ART @DN>]|PRELS]
@Männer [NN]
@lieben [[VVFIN @+FMAINV]|
[VVINF @−FMAINV]]
@Eva [NE]
@@

1.3 Erstellen einer Grammatik

Um eine Grammatik zu erstellen, werden in Didakowski (2005) spezielle Operatoren eingeführt.

Es ist ein Operator definiert, der es möglich macht, einen gewichteten Transduktor zu erstellen, der Chunks und ihre Struktur klammert. Der Optional-Criterion-Insertion-Operator (OCI) klammert optional Chunks, die durch einen Ausdruck A beschrieben werden, durch die Ausdrücke P und S. Zudem ist es möglich, Kriterien K_1,K_2, ...,K_n zur Disambiguierung anzugeben:

OCI-Operator:
$A(\rightarrow)P \ldots S//K_1,K_2, \ldots,K_n$

Der Ausdruck A kann einen gewichteten oder ungewichteten Transduktor denotieren; P, S, K_1, K_2, ..., K_n hingegen müssen einen gewichteten oder ungewichteten Automaten bezeichnen. Hier können auch potenzielle syntaktische Funktionen von Chunks angegeben werden. Diese können z.B. zur linken Satzklammer hinzugefügt werden.

Weiter ist ein Operator angegeben, der das syntaktische Tagging mit gewichteten Transduktoren ermöglicht. Mit dem *Optional-Restriction-Operator* ist es möglich, optional Tagabfolgen (Symbolabfolgen) zu gewichten. So können Constraints erstellt werden, mit denen das syntaktische Tagging realisiert wird:

Optional-Restriction-Operator:
$A(\Rightarrow)B\_C$
$A(\Rightarrow)\_C$
$A(\Rightarrow)B\_$

Die Ausdrücke A, B und C können gewichtete und ungewichtete Transduktoren denotieren. Es können also zusätzlich zur Gewichtung Transduktionen durchgeführt werden. So können innerhalb einer komplexeren Constraint-Grammatik Marker eingeführt werden, die das Formulieren einer Grammatik vereinfachen.

[4] `www.sfs.nphil.uni-tuebingen.de/Elwis/stts/stts-guide.ps.gz`

Hier soll der Optional-Restriction-Operator um eine Variante erweitert werden. Manchmal soll ein Ausdruck gewichtet oder eine Transduktion durchgeführt werden, ohne dass ein bestimmter Kontext angegeben wird:

$$\text{A}(\Rightarrow)\ \_ =_{def} \text{ ?*A?*}$$

Diese Variante soll auch für den obligatorischen Fall[5] gelten. Hierfür muss vorab ein Operator definiert werden, der den Support eines gewichteten Automaten bzw. Transduktors bestimmt (vgl. Kuich & Salomaa 1985):

$$\textit{Support:}\ \text{Supp(A)}$$

Bei der Anwendung auf einen ungewichteten Transduktor bzw. Automaten soll diese Funktion keinen Effekt haben. Dann ist folgende Variante des Restriction-Operators definiert, wobei `Dom` den Definitionsbereich und `Range` den Wertebereich eines Transduktors denotiert:

$$\begin{gathered}\text{A}\Rightarrow\ \_ =_{def}\\ [\sim \$\text{Supp(A)}\ \text{A}]^* \sim \$\text{Supp(A)}\end{gathered}$$

Um diese Variante auch auf Transduktoren anwenden zu können, ist die Definition wie folgt erweitert:

$$\begin{gathered}\text{A}\Rightarrow_{Dom}\ \_ =_{def}\\ [\sim \$\text{Dom(Supp(A))}\ \text{A}]^* \sim \$\text{Dom(Supp(A))}\end{gathered}$$

$$\begin{gathered}\text{A}\Rightarrow_{Range}\ \_ =_{def}\\ [\sim \$\text{Range(Supp(A))}\ \text{A}]^* \sim \$\text{Range(Supp(A))}\end{gathered}$$

Diese Varianten werden in diesem Aufsatz zwar nicht exzessiv verwendet, sie sind aber für das allgemeine Regelschreiben sehr nützlich.

1.4 Analyse eines einfachen Satzes

Die Analyse eines einfachen Satzes in Didakowski (2005) vollzieht sich in folgenden Schritten, wenn eine morphologisch analysierte Eingabe, die unter Umständen höchst ambig ist, angenommen wird:

$$\phi(\text{Range}\begin{pmatrix}\text{id(Eingabe)}\\ \text{.o.}\\ \text{Chunk-Grammatik}\\ \text{.o.}\\ \text{Constraint-Grammatik}\end{pmatrix})$$

Durch die Bestimmung des besten Pfades, die durch den Operator ϕ denotiert wird, wird durch einen Single-Source-Shortest-Path-Algorithmus genau eine Lesart der Eingabe ermittelt. Es soll ein Beispiel für eine Analyse gegeben werden. Der einfache Satz „Ich liebe die Frau in der Küche“ ist wie folgt analysiert:

@@ Ich liebe die Frau in der Küche @@
$\Downarrow$
@@ {$_{np}$ich}$_{np}$@SUBJ liebe@+FMAIN
{$_{np}$die Frau}$_{np}$@OBJ {$_{pp}$in{$_{np}$der
Küche}$_{np}$}$_{pp}$@<NOM-ADVL @@

Es werden alle möglichen Lesarten generiert, ob sie nun grammatisch sind oder nicht. Chunks können verschiedene Ausdehnungen haben und leere funktionale Elemente annehmen. Chunks und Wörter, die nicht durch das Chunking behandelt werden, können verschiedene potenzielle syntaktische Funktionen in einem Satz haben. Aus diesen Lesarten wird nun eine (hoffentlich) grammatische Lesart herausgefiltert.

[5] Der Restriction-Operator ist in Karttunen et al. (1996) definiert.

1.5 Probleme bei globalen Ambiguitäten

Das in Didakowski (2005) beschriebene Verfahren kann jedoch nur unter Zuhilfenahme von zusätzlichen Symbolen globale Ambiguitäten ausdrücken. Das Tag @<NOM-ADVL drückt z.B. aus, dass es sich um ein Adverbial oder ein nominales Attribut handelt. Beim syntaktischen Tagging nach Karlsson (1999) würden hier die syntaktischen Tags @<NOM und @ADVL in Optionalität stehen (vgl. Koskenniemi 1990). Durch die Ermittlung von genau einem besten Pfad ist dies bei diesem Verfahren jedoch nicht möglich.

Mit Hilfe eines Single-Source-Shortest-Distance-Algorithmus (Mohri, 2002) soll hier aus diesem Grund nicht nur ein bester Pfad, sondern alle besten Pfade mit dem gleichen Gewicht ermittelt werden. So ist es möglich, Ambiguitäten und damit auch die globalen Ambiguitäten zu erhalten:

@@ Ich liebe die Frau in der Küche @@
$\Downarrow$
@@ $\{_{np}$ich$\}_{np}$@SUBJ liebe@+FMAIN
$\{_{np}$die Frau$\}_{np}$@OBJ $\{_{pp}$in$\{_{np}$der
Küche$\}_{np}\}_{pp}$[@<NOM|@ADVL] @@

Das Erhalten von Ambiguitäten ist später für die Analyse von Konstituentensätzen relevant.

2 Analyse von Konstituentensätzen

Bei dem Verfahren in Didakowski (2005) werden nur einfache Sätze behandelt. Hier sollen aber auch Konstituentensätze analysiert werden.

Abney (1996) analysiert Konstituentensätze als Schwestern. So wird die Einbettung durch Iteration ersetzt. Falls die Konstituentensätze keine Zentraleinbettung darstellen, macht diese Betrachtungsweise Sinn. Jedoch stößt man bei Zentraleinbettungen auf Probleme, da hier ein Konstituentensatz nicht mehr als Schwester angesehen werden kann.

Koskenniemi (1990) behandelt Konstituentensätze durch das syntaktische Tagging. Als Erstes werden überall Konstituentensatzgrenzen angenommen. Diese werden dann durch Constraints festgesetzt. Auch hier ist die Einbettungstiefe beschränkt.

In Joshi & Hopely (1999) werden Konstituentensätze bottom-up aufgebaut. Zuerst werden die am tiefsten eingebetteten Konstituentensätze geklammert, dann die am nächsttiefsten usw., bis der Matrixsatz erreicht ist. Dieser Ansatz ist jedoch nicht finite-state, da beliebig viele Einbettungen erlaubt sind.

Hier soll ein Verfahren ähnlich dem von Joshi & Hopely (1999) vorgestellt werden. Konstituentensätze können ähnlich wie Chunks geklammert werden, um eine Struktur sichtbar zu machen. Die Einbettungstiefe ist hierbei theoretisch unbegrenzt[6]. Hier sollen zuerst k Einbettungen möglich sein. Später wird das Verfahren auf beliebig viele Einbettungen erweitert.

Es soll ein Operator entwickelt werden, der Konstituentensätze klammert. Im Deutschen ist es notwendig, den Kontext eines Satzes zu betrachten. Bei Relativsätzen herrscht z.B. eine Kasus-, Numerus- und Genus-Kongruenz zwischen dem Relativpronomen und seinem vorangehendem Bezugswort. Es sollen daher nicht wie beim Chunking nur bestimmte Abfolgen geklammert werden, sondern es sollen bestimmte Abfolgen (A) in einem bestimmten linken und rechten Kontext (L, R) durch Klammern (P, S) geklammert werden. Um Disambiguierungsstrategien zu realisieren, muss es zudem möglich sein, Kriterien (K1, K2 ... K3) anzuwenden:

[6] Psycholinguistisch sind Einbettungen natürlich begrenzt, da das menschliche Satzanalysesystem große Probleme damit hat, Satzteile miteinander in Verbindung zu bringen, die sehr weit voneinander entfernt sind (vgl. Dijkstra & Kempen 1993).

Optional-Criterion-Subclause-Operator (OCS-Operator):
$L \dots A \dots R(\rightarrow) \dots P \dots S \dots //K_1,K_2, \dots,K_n =_{def}$

$$[[\sim \$[< \mid >]L[0.x.<]A[0.x.>]R]^* \sim \$[< \mid >]]$$
.o.
$$id(K_1).o.id(K_2).o. \dots .o.id(K_n)$$
.o.
$$[<\rightarrow P].o.[>\rightarrow S]$$

Es lassen sich Varianten des OCS-Operators definieren, bei denen die Kontexte weggelassen werden können:

- $\dots A \dots R(\rightarrow) \dots P \dots S \dots //K_1,K_2, \dots,K_n =_{def} 0 \dots A \dots R(\rightarrow) \dots P \dots S \dots //K_1,K_2, \dots,K_n$
- $L \dots A \dots (\rightarrow) \dots P \dots S \dots //K_1,K_2, \dots,K_n =_{def} L \dots A \dots 0(\rightarrow) \dots P \dots S \dots //K_1,K_2, \dots,K_n$
- $\dots A \dots (\rightarrow) \dots P \dots S \dots //K_1,K_2, \dots,K_n =_{def} 0 \dots A \dots 0(\rightarrow) \dots P \dots S \dots //K_1,K_2, \dots,K_n$

Durch den OCS-Operator kann nun ein gewichteter Transduktor erstellt werden, der Konstituentensätze klammert. Zudem können Kriterien angegeben werden, die z.B. die Ausdehnung der Konstituentensätze durch eine Longest-Match-Strategie disambiguieren können.

Für Kopffunktionen gilt das Eindeutigkeitsprinzip. Wenn ein einfacher Satz analysiert wird, darf eine Kopffunktion genau einmal vorkommen. Das Eindeutigkeitsprinzip gilt hierbei unabhängig für alle Konstituentensätze. Also können Constraint-Grammatiken unabhängig für die Konstituentensätze entwickelt werden.

Das syntaktische Tagging innerhalb von Konstituentensätzen kann durch ein Kriterium realisiert werden. Constraints beinhalten zu präferierende Abfolgen. Diese Präferierungen innerhalb von Konstituentensätzen sind durch das Kriterium $\mathcal{K}_{Tag\_Pref\_KS}$ realisiert:

$\mathcal{K}_{Tag\_Pref\_KS}$**:**
Wörter in einem gewichteten Transduktor A werden durch (i) geordnet.
(i) Vordefinierte Abfolgen von Zeichen innerhalb eines Konstituentensatzes werden präferiert.

Dieses Kriterium kann durch den Bewertungssemiring $(\mathbf{N} \cup \{-\infty\}, max, +, -\infty, 0)$ beschrieben werden. Der gewichtete Automat, der Klammerinhalte nach $\mathcal{K}_{Tag\_Pref\_KS}$ gewichtet, ist folgendermaßen definiert, wobei A die durch Constraints definierten präferierten Abfolgen denotiert:

$$filter(A) =_{def} [\sim \$[< \mid >] < A >]^* \sim \$[< \mid >]$$

Dieses Kriterium bezieht sich auf die vom OCS-Operator eingefügten spitzen Klammern (<,>).

Hier sollen zudem Kriterien angenommen werden, die für die Klammerung von Konstituentensätzen eine Longest-Match-Strategie realisieren. Diese sollen in dem gewichteten Automaten *LongestMatch* zusammengefasst sein. Auf den Bewertungssemiring, der die Kriterien realisiert, soll hier nicht eingegangen werden.

Nun kann ein gewichteter Transduktor erstellt werden, der Konstituentensätze klammert. Es sind die Kontexte L und R, der Konstituentensatz A selbst und die Constraint-Grammatik F gegeben:

$$L,A,R(\rightarrow) \dots \{_s \dots \}_s \dots //id(LongestMatch),filter(F)$$

Es soll ein Satz mit einem enthaltenen Konstituentensatz analysiert werden. Im ersten Schritt wird eine ambige Eingabe als gewichteter Automat erstellt. Nun wird die Eingabe durch eine Chunk-Grammatik (CG) mit syntaktischer Struktur angereichert. Die Eingabe enthält nun alle morphologischen Lesarten, alle möglichen Klammerungen von Chunks und alle potenziellen syntaktischen Funktionen. Dann werden die am weitesten eingebetten Konstituentensätze durch eine Konstituentensatzgrammatik (KSG) geklammert, dann die am zweittiefsten usw. Als Letztes wird die Matrixsatzgrammatik (MSG) angewendet, die das syntaktische Tagging des Matrixsatzes realisiert.

$$\phi(\text{Range}\left(\begin{array}{c}\text{id(Eingabe)}\\ .o.\\ \text{CG}\\ .o.\\ \text{KSG}_1\\ .o.\\ \text{KSG}_2\\ .o.\\ \dots\\ .o.\\ \text{KSG}_n\\ .o.\\ \text{MSG}\end{array}\right))$$

Es wäre auch möglich, die Analyse-Transduktoren vor der Anwendung durch die Komposition zusammenzufassen. Das wäre aber nicht ratsam. In der Praxis werden die gewichteten Transduktoren, wenn sie zu einem Transduktor komponiert werden, zu groß. Zudem tritt dann auch ein Problem auf, das im Folgenden erläutert werden soll.

2.1 Redundanzen bei der Analyse

Bei der hier aufgeführten Analyse treten Redundanzen auf (überflüssiger Ballast). Es werden manche Teile mehrfach analysiert. Folgende Skizze für eine Zentraleinbettung soll den Sachverhalt verdeutlichen.

$$\left[\begin{array}{ll}\text{Matrix} & \left[\begin{array}{c}\text{Konstituentensatz}_{L_1}\\ \text{Konstituentensatz}_{L_2}\\ \dots\\ \text{Konstituentensatz}_{L_3}\end{array}\right] \text{satz}_{L_1}\\ \text{Matrix} & \left[\begin{array}{c}\text{Konstituentensatz}_{L_1}\\ \text{Konstituentensatz}_{L_2}\\ \dots\\ \text{Konstituentensatz}_{L_3}\end{array}\right] \text{satz}_{L_2}\\ & \dots\\ \text{Matrix} & \left[\begin{array}{c}\text{Konstituentensatz}_{L_1}\\ \text{Konstituentensatz}_{L_2}\\ \dots\\ \text{Konstituentensatz}_{L_3}\end{array}\right] \text{satz}_{L_3}\end{array}\right]$$

Jede Lesart des Matrixsatzes (L1, L2,...) enthält alle Lesarten der enthaltenen Konstituentensätze (L1, L2,...). Bei mehreren Einbettungen steigt die Anzahl enthaltener Lesarten rapide an. Es könnte jedoch teilweise früher disambiguiert werden, ohne dass spätere, für die Analyse notwendige Lesarten wegfallen. Lokale Ambiguitäten lassen sich innerhalb eines Satzes oder Konstituentensatzes auflösen. Daher könnten Lesarten früher eliminiert werden, nämlich genau dann, wenn vorher ein Konstituentensatz geklammert wurde.

Es wäre hierbei möglich, nach jedem Klammern eines Konstituentensatzes den besten Pfad zu berechnen. In diesem Fall würden jedoch auch Chunks disambiguiert werden, und es könnte zu einem Garden-Path-Effekt kommen. Zudem könnte es verschiedene Klammerungsmöglichkeiten der eingebetteten Sätze geben. So könnte durch die Bestimmung des besten Pfades eine potenzielle Konstituentensatzklammerung wegfallen, die später relevant ist.

@@ der Mann, der die Frau, die schläft, liebt. @@
*@@ der Mann{, der die Frau, } die schläft, liebt. @@
@@ der Mann, der die Frau{, die schläft,} liebt. @@

Es soll daher direkt innerhalb der Konstituentensätze disambiguiert werden. Auf diese Weise kann viel Mehrarbeit erspart werden. Hierzu muss ein Kriterium als Bewertungssemiring und als gewichteter Automat entwickelt werden, mit deren Hilfe es möglich ist, lokal zu disambiguieren. Dieses Kriterium ist als $\mathcal{K}_{Loc\_Dis}$ folgendermaßen formuliert:

$\mathcal{K}_{Loc\_Dis}$:
Wörter in einem gewichteten Transduktor A werden durch (i) ausgeschlossen.
(i) Innerhalb von Satzklammern wird disambiguiert.

Dieses Kriterium wird mit Hilfe des Bewertungssemirings $(\mathbf{N} \cup \{-\infty\}, max, +, -\infty, 0)$ realisiert. Eine öffnende Konstituentensatzklammer erhält das Gewicht 1 und eine schließende das Gewicht -1. Um dieses Kriterium auszudrücken, ist ein gewichteter Automat durch Local_Dis definiert, der Klammern gewichtet[7]:

$$\text{Local\_Dis} =_{def}$$
$$[\sim \$[< \mid >][<] < \mathcal{K}_{Loc\_Dis}, 1 > \sim \$[< \mid >][>] < \mathcal{K}_{Loc\_Dis}, -1 >]^* \sim \$[< \mid >]$$

Durch das Kartesische Produkt von Semiringen wird nun ein Semiring gebildet, mit dem eine lokale Disambiguierung möglich wird. $(A, \oplus, \otimes, \bar{0}, \bar{1})$ soll für den Bewertungssemiring stehen, mit dem die Disambiguierung realisiert ist. S $= (\mathbf{N} \cup \{-\infty\}, max, +, -\infty, 0) \times (A, \oplus, \otimes, \bar{0}, \bar{1})$ bildet nun einen neuen Semiring.

Um die gewichteten Transduktoren zu kompilieren, wird die durch $\oplus$ definierte Ordnung über S verwendet:

$$S_1 \leq S_2 \Longleftrightarrow S_1 \oplus S_2 = S_1$$

Der Semiring S stellt keinen Bewertungssemiring dar, da er nicht idempotent ist und so keine lineare Ordnung über S gebildet werden kann (vgl. Bistarelli 2004).

Um nun das lokale Disambiguieren zu realisieren, muss die lineare Ordnung der Elemente des Bewertungssemirings $(A, \oplus, \otimes, \bar{0}, \bar{1})$ vergröbert werden, indem Äquivalenzklassen gebildet werden. Wörter sollen nur noch anhand der Gewichtungen innerhalb von Satzklammern geordnet werden. Hierzu soll die Ordnung $\leq_{local}$ über S definiert werden. Gegeben sind (A_1, B_1) und $(A_2, B_2) \in$ S, dann ist die Ordnung $\leq_{local}$ wie folgt definiert:

$$(A_1, B_1) \leq (A_2, B_2)$$
$$\Longleftrightarrow$$
$$A_1 > \text{o und } B_1 \oplus B_2 = B_2$$
$$\text{oder}$$
$$A_2 > \text{o und } B_1 \oplus B_2 = B_2$$

Es wird erst angefangen zu disambiguieren, wenn eine öffnende Konstituentensatzklammer auftritt, und erst dann aufgehört, wenn genauso viele öffnende wie schließende Klammern innerhalb eines Pfades auftreten. Auf diese Weise wird gewährleistet, dass grammatische Hypothesen so früh wie möglich isoliert und nicht blockiert werden:

$$\left[\begin{array}{l} \text{Matrix}\left[\ \text{Konstituentensatz}_{Lbest}\ \right] \text{satz}_{L_1} \\ \text{Matrix}\left[\ \text{Konstituentensatz}_{Lbest}\ \right] \text{satz}_{L_2} \\ \qquad \ldots \\ \text{Matrix}\left[\ \text{Konstituentensatz}_{Lbest}\ \right] \text{satz}_{L_3} \end{array}\right]$$

Es ist ersichtlich, dass die Satzgrenzenmarker @@ beim syntaktischen Tagging des Matrixsatzes auch die Gewichte 1 und -1 für das Kriterium $\mathcal{K}_{Loc\_Dis}$ erhalten müssen.

[7]Die Gewichte werden hier durch eine Indexfunktion zugewiesen, die hier im Gegensatz zu Didakowski (2005) auch für Tupel aus Tupeln funktioniert.

$$\text{Matrixsatz\_Disambiguierung} =_{def}$$
$$@@<\mathcal{K}_{Loc\_Dis}, 1> \sim \$@@\ @@<\mathcal{K}_{Loc\_Dis}, -1>$$

Jedoch können immer noch Redundanzen auftreten, was mit dem optionalen Klammern von Konstituentensätzen zu tun hat. Konstituentensätze können wiederum Konstituentensätze enthalten. Die Klammerungssymbole können hierbei identisch sein. Nun können aus diesem Grund gleiche Strukturen doppelt generiert bzw. vorhergesagt werden.

Erste Konstituentensatzebene:

- @@ Hans, der Eva, weil sie {, die schläft,} blond ist, liebt, schläft auch. @@
- @@ Hans, der Eva, weil sie, die schläft, blond ist, liebt, schläft auch. @@

Zweite Konstituentensatzebene:

- @@ Hans, der Eva, weil sie {, die schläft,} blond ist, liebt, schläft auch. @@
- @@ Hans, der Eva, weil sie, die schläft, blond ist, liebt, schläft auch. @@
- @@ Hans, der Eva{, weil sie {, die schläft,} blond ist,} liebt, schläft auch. @@
- @@ Hans, der Eva, weil sie {, die schläft,} blond ist, liebt, schläft auch. @@ ⇒ Doppelung

Dritte Konstituentensatzebene:

- @@ Hans, der Eva, weil sie {, die schläft,} blond ist, liebt, schläft auch. @@
- @@ Hans, der Eva, weil sie, die schläft, blond ist, liebt, schläft auch. @@
- @@ Hans, der Eva{, weil sie {, die schläft,} blond ist,} liebt, schläft auch. @@
- @@ Hans, der Eva, weil sie {, die schläft,} blond ist, liebt, schläft auch. @@ ⇒ Doppelung
- @@ Hans, der Eva, weil sie {, die schläft,} blond ist, liebt, schläft auch. @@ ⇒ Doppelung
- @@ Hans, der Eva{, weil sie {, die schläft,} blond ist,} liebt, schläft auch. @@ ⇒ Doppelung
- @@ Hans{, der Eva{, weil sie {, die schläft,} blond ist,} liebt,} schläft auch. @@

Konstituentensätze, die optional geklammert wurden, können in der nächsten Ebene ein zweites Mal geklammert werden usw. So verdoppeln sich die Analysen Einbettungsebene für Einbettungsebene. Das soll durch eine Kompilationstechnik verhindert werden.

Die nicht am tiefsten eingebetteten Konstituentensätze müssen Konstituentensätze enthalten. Daher muss bei diesen im Gegensatz zu den am weitesten eingebetteten Konstituentensätzen ein Konstituentensatz im Muster vorkommen. So wird verhindert, dass die am tiefsten eingebetteten Konstituentensätze mehrere Male analysiert werden. Es sind folgende Constraints für beide Fälle definiert:

$$\text{Contain\_No\_Brackets(P,S)} =_{def}$$
$$\sim \$[P|S]$$

$$\text{Contain\_Brackets(P,S)} =_{def}$$
$$\$[P|S]$$

Konstituentensätze, die in einer Konstituentensatzebene n nicht Teil eines anderen Konstituentensatzes sein können, sind auch in einer Ebene $n+1$ kein Teil eines anderen Konstituentensatzes (vorausgesetzt, es wird immer die gleiche KSG benutzt). Aus diesem Grund werden die Konstituentensätze, die nicht in einer höheren Ebene innerhalb eines Konstituentensatzes geklammert werden, markiert. Die markierten Konstituentensätze werden dann nicht mehr in andere Konstituentensätze eingebettet. So wird verhindert, dass Sätze mehrere Male analysiert werden.

Daher soll ein externer Filter für den OCS-Operator definiert werden:

$$\text{ext\_filter(A)} =_{def}$$
$$[[\sim \$[< \mid >].o.A][<] \sim \$[< \mid >][>], -1 >]^*[\sim \$[< \mid >].o.A]$$

Mit Hilfe von ext_filter(A) können nun Statusinformationen außerhalb eines Konstituentensatzes geändert werden. Es sollen Symbole als Statusinformation eingeführt werden. „yes“ bedeutet aktiv und „no“ bedeutet inaktiv. Außerhalb von Satzklammerungen sollen andere Satzklammern von „yes“ auf „no“ gesetzt werden:

$$\text{marker(P,S)} =_{def} [[\text{P}|\text{S}]\text{yes:no}] \Rightarrow_{Dom \_}$$

Konstituentensatzklammerungen mit dem Status „no“ sollen nun kein Teil einer anderen Konstituentensatzklammerung werden können.

Um das Grammatikschreiben zu Konstituentensätzen zu vereinfachen, können Wortgrenzen ein Merkmal bekommen, das angibt, ob es schon innerhalb eines Konstituentensatzes eingeschlossen ist und daher außerhalb des Konstituentensatzes nicht mehr relevant ist. Hierfür sollen auch die Symbole „yes“ und „no“ dienen. „yes“ wird dann innerhalb von Konstituentensätzen auf „no“ geändert.

Nun kann ein gewichteter Transduktor erstellt werden, der Konstituentensätze klammert. Der Ausdruck F denotiert hier einen Transduktor, der das syntaktische Tagging innerhalb von Konstituentensätzen realisieren soll:

- **Am weitesten eingebettete Konstituentensätze:**

  KSG1 $=_{def}$
  L,A,R($\rightarrow$) $\ldots\{\ldots\}\ldots //$
  LongestMatch,
  filter(id(Contain_No_Brackets({,})).o.F)

- **Alle anderen Konstituentensätze:**

  KSG2 $=_{def}$
  L,A,R($\rightarrow$)$\ldots\{_s\ldots\}_s\ldots //$
  LongestMatch,
  ext_filter(marker({,})),
  filter(id(Contain_Brackets({,})).o.F)

Die Analyse eines Satzes vollzieht sich nun in folgenden Schritten:

$$\phi(\text{Range}\left(\begin{array}{c}\phi\left(\begin{array}{c}\ldots\phi\left(\begin{array}{c}\phi\left(\begin{array}{c}\text{id(Eing.)}\\ .\text{o.}\\ \text{CG.}\\ .\text{o.}\\ \text{KSG1}\end{array}\right)\\ .\text{o.}\\ \text{KSG2}_1\end{array}\right)\\ .\text{o.}\\ \ldots\\ .\text{o.}\\ \text{KSG2}_n\end{array}\right)\\ .\text{o.}\\ \text{MSG}\end{array}\right))$$

2.2 Beliebige Einbettungen

Es wird vorausgesetzt, dass es genau eine rekursive Konstituentensatzgrammatik gibt. Von dieser Grammatik soll eine Variante existieren, die ausschließlich die am weitesten eingebetteten Konstituentensätze klammert (KSG1). Die andere Variante klammert alle anderen Konstituentensätze, wobei der Status von Konstituentensätzen außerhalb von Klammerungen auf „no“ gesetzt wird (KSG2). Durch eine einfache Abbruchbedingung lassen sich nun auch beliebig viele Einbettungen behandeln.

Ein Satz wird analysiert, indem sukzessive mögliche Strukturen generiert und gewichtet werden. Hierbei vergrößert sich der gewichtete Transduktor bei jeder Ebene, die die Klammerung von Konstituentensätzen betrifft, genau dann, wenn ein Konstituentensatz enthalten ist, der noch nicht geklammert wurde. Wenn sich die Größe (Zustands- und Übergangsanzahl) des Transduktors nicht verändert, ist auch kein zu klammernder Konstituentensatz mehr enthalten. So könnte leicht nach jeder entsprechenden Komposition die Größe überprüft werden. Wenn sich diese nicht verändert hat, kann angenommen werden, dass keine Konstituentensätze mehr zu analysieren sind und der Matrixsatz analysiert werden kann. Die Ermittlung der Größe kann in konstanter Zeit ermittelt werden (dabei wird angenommen, dass über die Zustands- und Übergangsanzahl Buch geführt wird), so dass die Vergleichsoperation in konstanter Zeit durchführbar ist[8].

Diese Analyse hat den Vorteil, dass nicht immer k Konstituentensatzebenen analysiert werden, obwohl ein Satz keine oder weniger Einbettungen enthält. Hier wird nur eine Kompositionsebene zu viel durchlaufen. Dieser Ansatz ist jedoch nicht finite-state. Es können kontextfreie Strukturen analysiert werden, die Komplexität ist hierbei O(n^3), da die Anzahl der Einbettungen von der Länge der Eingabe abhängen. Es handelt sich hierbei also um einen kontextfreien Bottom-Up-Parser.

Input: Eingabe,CG,KSG1,KSG2,MSG

$$\text{analyse}_1 =_{def} \phi \begin{pmatrix} \text{id(Eing.)} \\ \text{.o.} \\ \text{CG.} \\ \text{.o.} \\ \text{KSG1} \end{pmatrix}$$

while (Analyse$_1$!=

$$(\text{Analyse}_2 =_{def} \phi \begin{pmatrix} \text{analyse}_1 \\ \text{.o.} \\ \text{KSG2} \end{pmatrix}))$$

$\text{Analyse}_1 =_{def} \text{Analyse}_2$

Output: $\phi(\text{Range} \begin{pmatrix} \text{Analyse}_1 \\ \text{.o.} \\ \text{MSG} \end{pmatrix})$

Jedoch liegt es hier an der Grammatik, ob der Algorithmus terminiert. Nur mit den speziellen Eigenschaften, die hier vorausgesetzt werden, ist eine erfolgreiche Analyse möglich.

3 Zusammenfassung

Es wurde der Ansatz aus Didakowski (2005) um die Verarbeitung von Konstituentensätzen erweitert. Konstituentensätze werden bottom-up geklammert, dabei vollzieht sich die Analyse in mehreren Schritten. Nach jedem Schritt wird innerhalb von Konstituentensätzen disambiguiert, um so früh wie möglich Lesarten auszuschließen. Hierfür wurde ein spezieller Semiring entwickelt, der es möglich macht, lokal zu disambiguieren. Weiterführend wurden durch eine Kompilationstechnik weitere Redundanzen vermieden, die aus dem optionalen Klammern von Konstituentensätzen resultieren. Es wurde zuerst eine Analyse mit k Einbettungen vorgestellt. Diese wurde dann um unbegrenzt viele Einbettungen erweitert, dabei ist die Komplexität jedoch kubisch.

[8]Eine ähnliche Abbruchbedingung verwendet Roche in Roche (1997) bei einem Top-Down Parsing-Verfahren mit Finite-State-Maschinen.

Literatur

Abney, S., 1990. *Syntactic Affixation and Performance Structures*. In: D. Bouchard, K. Leffel (eds.), *Views on Phrase Structure*. Kluwer Academic Publishers, Dordrecht.

Abney, S., 1996. *Partial Parsing via finite-state cascades*. In: Proceedings of the ESSLI workshop on robust parsing. Prague.

Bistarelli, S., 2004. *Semirings for Soft Constraint Solving and Programming*. Vol. 2962 of *Lecture Notes in Computer Science*. Springer, Berlin.

Didakowski, J., 2005. Robustes Parsing und Disambiguierung mit gewichteten Transduktoren. *LiP* **23**.

Dijkstra, T., Kempen, G., 1993. *Einführung in die Psycholinguistik*. Huber, Bern.

Frazier, L., Clifton, C. J., 1996. *Construal*. MIT Press, Cambridge, Massachusetts.

Halteren, H. v. (ed.), 1999. *Syntactic Wordclass Tagging*. Vol. 9 of *Text, speech and language technology series*. Kluwer Academic Publishers, Dordrecht.

Joshi, A. K., Hopely, P. D., 1999. *A parser from antiquity*. In: A. Kornai (ed.), *Extended Finite State Models of Language*. Cambridge University Press, Cambridge, 6–15.

Karlsson, F., 1999. *Constraint grammar as a framework for parsing running text*. In: Proceedings of the 13th International Conference on Computational Linguistics (COLING-90). Vol. 3. Helsinki, 168–173.

Karttunen, L., 1995. *The Replace Operator*. In: Meeting of the Association for Computational Linguistics. 16–23.

Karttunen, L., Chanod, J.-P., Grefenstette, G., Schiller, A., 1996. Regular expressions for language engineering. *Natural Language Engineering* **2** (4), 305–338.

Koskenniemi, K., 1990. *Finite-state parsing and disambiguation*. In: H. Karlgren (ed.), *COLING-90. Papers presented to the 13th International Conference on Computational Linguistics*. Vol. 2. Helsinki, 229–232.

Kuich, W., Salomaa, A., 1985. *Semirings, automata, languages*. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo.

Mohri, M., 2002. Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics* **7** (3), 321–350.

Roche, E., 1997. *Parsing with Finite-State Transducers*. In: E. Roche, Y. Schabes (eds.), *Finite-State Language Processing*. MIT Press, Cambridge, Massachusetts, Ch. 8.

Analyse bibliographischer Referenzen

Eva Anderl
Centrum für Informations- und Sprachverarbeitung
Ludwig-Maximilians-Universität München

Zusammenfassung

Bibliographische Referenzen haben eine inhärente Struktur, werden allerdings meist als wenig strukturierter, zusammenhängender Text wiedergegeben. Das Fehlen von einheitlichen und verbindlichen Normen macht die maschinelle Verarbeitung von derartigen Literaturangaben zu einem keineswegs trivialen Problem, das in den letzten Jahren zum Gegenstand intensiver Forschung geworden ist.
Dieser Artikel stellt typische Problemstellungen vor, die bei der Verarbeitung bibliographischer Referenzen auftreten, und gibt einen Überblick über den aktuellen Stand der Forschung.

1 Einleitung

Der Umgang mit bibliographischen Referenzen ist ein Grundbestandteil wissenschaftlichen Arbeitens: Für Autoren ist es aus Gründen der Ethik und des Urheberrechts verpflichtend, verwendete Quellen eindeutig anzugeben, und Leser sollen mithilfe der Angaben in der Lage sein, die referenzierten Dokumente zu identifizieren und zu lokalisieren.

Quellenangaben setzen sich aus einzelnen Teilinformationen wie Autor, Titel oder Seitenzahl zusammen, werden aber in der Regel als durchgehende Textstrings repräsentiert. Durch diese Art der Repräsentation geht die Information über die Funktion der einzelnen Bestandteile verloren und muss von einem menschlichen Leser erst wieder erschlossen werden. Ein Nebeneinander von zahlreichen Normen und Zitierkonventionen[1] gekoppelt mit idiosynkratischen Varianten und die daraus resultierende uneinheitliche Formatierung machen es bisweilen schon für den Menschen nicht einfach, den Aufbau einer bibliographischen Referenz vollständig zu verstehen, vor allem aber wird eine maschinelle Verarbeitung dadurch erheblich erschwert. Zur Illustration zeigt Abbildung 1 neun Quellenangaben, die alle den Artikel *Digital Libraries and Autonomous Citation Indexing* von Lawrence et al. (1999b) referenzieren: Jede der ausgewählten Quellenangaben hat eine andere Form; die Unterschiede sind zum Teil beträchtlich.

Sowohl bei der Verlinkung elektronischer Publikationen[2] als auch im Rahmen der Digitalisierung von Bibliotheken gewinnt eine maschinelle Verarbeitung von Literaturangaben immer mehr an Bedeutung. Aufgrund der oben dargestellten Situation stellt diese allerdings keineswegs ein triviales Problem dar, weshalb dieses Gebiet in den letzten Jahren Gegenstand zahlreicher Publikationen war. Im Rahmen dieses Artikels werden die wichtigsten Problemstellungen im Zusammenhang mit der Verarbeitung bibliographischer Angaben eingeführt und ein Überblick über den Stand der Forschung gegeben.

Grundsätzlich lassen sich zwei große Bereiche unterscheiden: Strukturanalyse und Datenabgleich. Ziel der Strukturanalyse ist es, aus einer gegebenen bibliographischen Referenz die einzelnen logischen Teileinheiten (*Felder*) wie Autor oder Titel zu extrahieren. Im Falle des Datenabgleichs werden die Quellenangaben dagegen nicht für sich betrachtet, sondern sollen mit anderen Daten z.B. aus bibliographischen Datenbanken abgeglichen und verknüpft werden.

[1] Hier sind unter anderem die großen nationalen und internationalen Normungsinstitutionen (ISO, 1987; ANSI, 2005; DIN, 1984, 1995), die sogenannten *Style Manuals* (CMS, 2003; APA, 2001; MLA, 2003) und Vorschriften wissenschaftlicher Zeitschriften und großer Verlage zu nennen.

[2] Für elektronische Publikationen gibt es – unter anderem zur Umgehung der oben genannten Probleme – Initiativen zur eindeutigen Identifizierung von Dokumenten über *Digital Object Identifiers* (Doi, 2005; Crossref, 2005); die entsprechenden Systeme befinden sich jedoch erst im Aufbau.

S.Lawrence, C. Giles, and K. Bollacker, „Digital libraries and autonomous citation indexing,“ IEEE Computer, vol. 32, no. 6, pp. 67–71, 1999.	(Yin et al., 2004)
S. Lawrence, C.L. Giles, and K. Bollacker. 1999. Digital Libraries and Autonomous Citation Indexing. *IEEE Computer*, 32(6): 67–71.	(Peng & McCallum, 2004)
Lawrence, S., Giles, C. L. & Bollacker, K. (1999). Digital libraries and autonomous citation indexing. *IEEE Computer, 32*(6), 67–71.	(McCallum et al., 2000b)
S. Lawrence, C.L. Giles, and K. Bollacker, „Digital Libraries and autonomous Citation indexing“, *IEEE Computer*, vol. 32(6), 1999, pp.67-71.	(Besagni & Belaïd, 2004)
S. Lawrence, and C.L. Giles, „Digital Libraries and Autonomous Citation Indexing,“ IEEE Computer, Volume 32, Number 6, 1999, pp.67-71.	(Berkowitz & Elkhadiri, 2004)
Steve Lawrence, C. Lee Giles, and Kurt Bollacker. Digital libraries and Autonomous Citation Indexing. *IEEE Computer*, 32(6):67-71, 1999.	(Geng, 2002)
Steve Lawrence, C. Lee Giles, and Kurt Bollacker. Digital libraries and autonomous citation indexing. *IEEE Computer*, 32(6):67-71, 1999. <http://www.researchindex.com>	(Bergmark & Lagoze, 2001)
Lawrence, S., Giles, C. L., and Bollacker, K. (1999), „Digital Libraries and Autonomous Citation Indexing,“ *IEEE Computer* 32(6), 67–71	(Demleitner et al., 2004)
S. Lawrence, C. L. Giles, and K. D. Bollacker. Digital libraries and autonomous citation indexing. *IEEE Computer*, 32(6):67–71, June 1999.	(Takasu, 2003)

Abbildung 1: Unterschiedliche Zitierweisen am Beispiel von Referenzen auf Lawrence et al. (1999b).

2 Strukturanalyse

Bei der Strukturanalyse oder Strukturextraktion soll die rein textuell repräsentierte Quellenangabe segmentiert und um Informationen zur Funktion der einzelnen Felder angereichert werden. Strukturextraktion wird üblicherweise als Vorverarbeitungsschritt des Datenabgleichs eingesetzt, aber in der Literatur meist als eigenständiges Problem beschrieben, weshalb sie auch hier gesondert betrachtet wird.

Zur Strukturanalyse existiert eine Vielzahl konkurrierender Ansätze, von denen die meisten auf wahrscheinlichkeitstheoretischen Verfahren beruhen.

2.1 Wahrscheinlichkeitstheoretische Verfahren

Das unter den stochastischen Methoden am weitesten verbreitete Konzept sind Hidden-Markov-Modelle (HMMs), andere Ansätze sind Support-Vector-Machines, Conditional-Random-Fields und probabilistische Modelle erster Ordnung. Die einzelnen Verfahren werden im Folgenden überblicksweise vorgestellt:

HMMs lassen sich als probabilistische endliche Automaten beschreiben, die in jedem Zustand gemäß einer Wahrscheinlichkeitsverteilung Ausgabesymbole emittieren.[3]

HMMs können auf unterschiedliche Weisen zur Analyse der Struktur von bibliographischen Referenzen eingesetzt werden. Beispielsweise versuchen Connan & Omlin (2000) mithilfe von manuell entwickelten HMMs den Zitierstil einer bibliographischen Referenz zu erkennen. Kann die Referenz einer eindeutigen Zitierkonvention zugeordnet werden, können damit – sofern der genaue Aufbau von Referenzen gemäß dieser Konvention bekannt ist – die einzelnen Felder extrahiert werden.

[3] Eine gute Einführung in Hidden-Markov-Modelle bietet Rabiner (1989).

Geläufiger sind jedoch Ansätze, in denen die HMMs direkt die Struktur der bibliographischen Referenzen repräsentieren. McCallum et al. (2000b) vergleichen verschiedene mögliche Strukturen für HMMs. Die besten Ergebnisse können erzielt werden, wenn zunächst ein sehr spezifisches HMM mit hoher Zustandszahl automatisch anhand eines Trainingskorpus erstellt wird und dieses anschließend durch verschiedene Merging-Techniken vereinfacht wird. Außerdem untersuchen McCallum et al. die Frage, in welcher Form geeignete Trainingsdaten vorliegen sollten. Außer manuell gelabelten Daten eignen sich auch Daten, die nicht speziell für diesen Zweck markiert wurden, aber eine ausreichend große Zahl an Überschneidungen aufweisen. Als Beispiel für bibliographische Referenzen werden BibTEX-Datenbanken angeführt.

Weitere Untersuchungen zu geeigneten Strukturen von einfachen HMMs finden sich in Geng (2002) und Geng & Yang (2004). Eine Erweiterung des Ansatzes von McCallum et al. (2000b) wird in Yin et al. (2004) beschrieben: die Struktur des HMMs wird wie oben beschrieben ermittelt, es werden aber nicht nur die Frequenzen einzelner Wörter, sondern auch die von Wort-Bigrammen miteinbezogen.

Eines der meistzitierten Verfahren ist das von Borkar et al. (2001), die ein sogenanntes geschachteltes HMM (*„nested HMM“*) verwenden. Ein *äußeres* HMM bildet die Abfolge der einzelnen Felder ab und enthält pro Feld genau einen Zustand. Jeder dieser Zustände enthält selbst wieder ein *inneres* HMM, das die interne Struktur des entsprechenden Feldes repräsentiert. Auf diese Weise können strukturelle Abhängigkeiten einzelner Felder genauer abgebildet werden.

Geschachtelte HMMs sind Spezialfälle von hierarchischen HMMs (Fine et al., 1998). In diese Gruppe können auch die von Takasu (2003) entwickelten *Dual and Variable-length Output Hidden-Markov-Models* eingeordnet werden, die auch fehlerhafte Referenzen verarbeiten können.

Agichtein & Ganti (2004) verfolgen einen Ansatz, der keine (manuell) getaggten Quellenangaben als Trainingsdaten benötigt, sondern seine Informationen aus bestehenden bibliographischen Datenbanken bezieht. Für jedes Feld wird anhand der Datenbank ein eigenes Hidden-Markov-Modell berechnet. Bei der Analyse wird für eine Gruppe von Referenzen durch Maximierung der Gesamtwahrscheinlichkeit die beste Abfolge der einzelnen Modelle ermittelt. Dabei wird davon ausgegangen, dass die Reihenfolge von Feldern innerhalb eines Dokuments gleich bleibt – ein Ansatz, der außerdem nur noch von Besagni et al. (2003) und Besagni & Belaïd (2004) verfolgt wird.

Bei der Extraktion von Titelinformationen aus wissenschaftlichen Publikationen liefern *Support-Vector-Machines* (SVMs) bessere Ergebnisse als klassische Hidden-Markov-Modelle (Han et al., 2003). SVMs sind überwachte Lernmethoden zur Klassifikation, die darauf abzielen, eine optimal trennende Hyperebene in einem hochdimensionalen Merkmalsraum zu finden (Schölkopf & Smola, 2002). Die für die Titelinformationen verwendeten Methoden dürften auf bibliographische Referenzen übertragbar sein. Okada et al. (2004) kombinieren SVMs mit Hidden-Markov-Modellen: Die Quellenangabe wird an möglichen Trennzeichen aufgesplittet und die erhaltenen Teilstrings werden mit Support-Vector-Machines klassifiziert. Ist die Klassifikation durch SVMs nicht eindeutig, werden über HMMs Informationen zur Abfolge von Feldern miteinbezogen.

Peng & McCallum (2004) verwenden *Conditional-Random-Fields* (CRFs) und erzielen damit sehr gute Ergebnisse. CRFs sind ein probabilistischer Ansatz zur Segmentierung sequenzieller Daten, der es erlaubt, schwächere Unabhängigkeitsannahmen zu verwenden als bei Hidden-Markov-Modellen (Wallach, 2004).

Einen weiteren neuen Ansatz zur Strukturanalyse stellt die Verwendung von probabilistischen Modellen erster Ordnung dar, mit deren Hilfe Wahrscheinlichkeitsverteilungen spezifiziert werden können. Die Erweiterung auf Prädikatenlogik erster Ordnung ermöglicht es, auch komplexe Objekte abzubilden. Im konkreten Fall werden die einzelnen Klassen (Autor, Herausgeber, etc.) mit ihren Attributen genau modelliert; Inferenz und Modellierung der einzelnen Parameter erfolgt mithilfe eines effizienten Verfahrens zur Approximation von Wahrscheinlichkeiten (*Markov-Chain-Monte-Carlo*, Pasula et al. 2002; Marthi et al. 2003).

2.2 Weitere Verfahren

Neben wahrscheinlichkeitstheoretischen Verfahren existieren zahlreiche weitere Methoden zur Strukturanalyse.

Programme, die Informationen aus Webseiten extrahieren und in ein passendes Format umwandeln, werden als *Wrapper* bezeichnet. Einen guten Überblick über dieses Thema gibt ein Artikel von Laender et al. (2002). Die meisten existierenden Wrapper-Systeme unterstützen lediglich Daten im HTML-Format und sind somit für rein textuelle bibliographische Referenzen nicht einsetzbar;[4] einige Systeme wie NoDoSe, RAPIER oder WHISK (Adelberg, 1998; Califf & Mooney, 1999; Soderland, 1999) können aber auch für Quellenangaben verwendet werden, die nur als reine Textstrings vorliegen.

Ein speziell für bibliographische Daten manuell entwickeltes System wird in Ding et al. (1999) vorgestellt: Basierend auf einer ausführlichen Korpusanalyse wurden Schablonen, sogenannte *Templates*, erstellt, die den Aufbau von bibliographischen Referenzen abbilden. Ein ähnlicher Ansatz wird in dem Perl-Modul Biblio::Citation::Parser verfolgt. Reguläre Ausdrücke, die die einzelnen Felder repräsentieren, können gemäß festgelegten Mustern kombiniert werden (Jewell, 2003).[5]

Parmentier setzt zur Strukturextraktion neuronale Netze ein (Parmentier & Belaïd, 1997; Parmentier, 1998), Besagni et al. verwenden eine auf Part-of-Speech-Tagging basierende Methode (Besagni et al., 2003; Besagni & Belaïd, 2004): In einem Bottom-Up-Verfahren werden anhand eines Lexikons zunächst einzelne Token getaggt, die dann durch syntaktische Regeln zu Feldern zusammengefasst werden. Felder, die nicht eindeutig oder nur inkonsistent aufgelöst werden können, werden anhand der Struktur der eindeutig gelabelten Referenzen erschlossen. Damit werden strukturelle Gemeinsamkeiten von Referenzen innerhalb eines Dokuments berücksichtigt.

Day et al. (2005) verwenden zur Abbildung der Regularitäten bibliographischer Referenzen eine ontologische Wissensrepräsentation, mithilfe derer weitere Referenzen analysiert werden können.

3 Datenabgleich

Häufig sollen bibliographische Referenzen nicht nur analysiert, sondern sollen mit anderen Daten abgeglichen und verknüpft werden. Dabei kann sowohl ein Abgleich mit anderen Referenzen als auch mit bibliographischen Datenbanken erfolgen.

Beim Datenabgleich ist zu beachten, dass für Entitäten wie Personen, Zeitschriften, Konferenzen, etc. zahlreiche Bezeichnungsvarianten bestehen, die geeignet aufeinander abgebildet werden müssen. Zudem soll häufig auch für fehlerhafte Referenzen der entsprechende Eintrag ermittelt werden können, weshalb teilweise Techniken zum approximativen Matching eingesetzt werden.

3.1 Abgleich von Referenzen mit Referenzen

Beim Abgleich von Referenzen mit Referenzen (*Citation-Matching*) sollen Angaben ermittelt werden, die auf dieselbe Quelle verweisen. Dies ist vor allem bei der Durchführung von bibliometrischen Analysen von Bedeutung. Unter Bibliometrie versteht man die quantitative Analyse von wissenschaftlicher Kommunikation mit statistischen Methoden.[6] Im Zusammenhang mit bibliographischen Referenzen ist besonders die sogenannte Zitationsanalyse von Bedeutung, bei der beispielsweise die Bedeutung einer Publikation anhand der Zahl ihrer Zitierungen ermittelt wird oder Dokumente anhand gemeinsamer bibliographischer Referenzen thematisch gruppiert werden.

Zu den Verfahren, die beim *Citation-Matching* eingesetzt werden, gehören Editierdistanzmessungen und Frequenzanalysen. Eine vorausgehende Strukturanalyse der bibliographischen Referenzen kann das

[4] Zur Extraktion von bibliographischen Daten aus HTML-Seiten vgl. z.B. Ortyl & Pfingstl (2004).

[5] Eine Implementierung des Verfahrens ist unter `http://www.cpan.org` frei verfügbar.

[6] Ein Standardwerk zur Bibliometrie ist Garfield (1979).

Matching der Quellenangaben erheblich vereinfachen (Lawrence et al., 1999a). Der vom vermutlich bekanntesten Web-Zitationsindex Citeseer/ResearchIndex (Citeseer, 2005) verwendete Algorithmus basiert auf einer Normalisierung der Referenzen, Sortierung nach Länge und Wortmatching innerhalb von einzelnen heuristisch erkannten Feldern wie Jahr und Seitenzahlen (Lawrence et al., 1999b).

3.2 Abgleich von Referenzen mit Datenbankeinträgen

Häufiger als das Problem, Referenzen untereinander abzugleichen, ist die Aufgabe, herauszufinden, ob das von einer Quellenangabe referenzierte Dokument in einer gegebenen Datenbank enthalten ist. Ziel des Abgleichs kann sowohl eine Erweiterung bzw. Ergänzung der Datenbasis (*Data Cleaning*)[7] oder eine Korrektur der bibliographischen Referenz sein. Letzteres ist vor allem im Rahmen der OCR-Nachkorrektur von Bedeutung. Außerdem können Online-Dokumente über den entsprechenden Datenbankeintrag direkt verlinkt werden (*Citation-Linking*).

In den meisten Verfahren zum Datenbankabgleich wird regelbasiert versucht, einzelne Felder der bibliographischen Referenz zu extrahieren und diese auf die Datenbank abzubilden. Das System von Kratzer (2002) versucht mithilfe von Regeln Autor, Jahr, Bandnummer und Seitenzahl zu erkennen und ermittelt die Zeitschrift durch Abgleich mit einer Liste von Zeitschriften und deren Abkürzungen. Ein Treffer wird ausgegeben, wenn ein Datenbankeintrag in bestimmten Feldern übereinstimmt. Ein sehr ähnliches Verfahren verwenden Claivaz et al. (2001), die Internetadressen, Nummern und Zeitschriften extrahieren (letztere ebenfalls über eine entsprechende Wissensbasis). Auch Bergmark (2000) geht von einer heuristischen Extraktion einzelner Felder aus, beginnend mit Jahr und Seitenzahl.[8]

Das vom *NASA Astrophysics Data System* verwendete Verfahren (Demleitner et al., 2004) basiert auf einem theoretischen Konzept der maschinellen Sprachverarbeitung, den Dependenzgrammatiken (Heringer, 1993). Der Kopf einer Phrase eröffnet bestimmte Leerstellen, sogenannte *Slots*, die geeignet gefüllt werden müssen. Für bibliographische Referenzen wird als Kopf in der Regel die Quelle der Publikation gewählt, die mit einem regulären Ausdruck extrahiert und über ein Kodierungsverfahren mit der Datenbasis abgeglichen wird. Ausgehend von den durch die Quelle eröffneten Slots werden mögliche Kandidaten durch approximativen Abgleich mit der Datenbank validiert. Auf diese Weise eignet sich das Verfahren auch zur Verarbeitung fehlerhafter Referenzen, wie sie bei der Verwendung von OCR-Daten auftreten können.

Ein anderer Ansatz zum Abgleich fehlerhafter Quellenangaben (Anderl, 2005) verwendet eine – speziell auf die approximative Suche mehrerer Suchterme zugeschnittene – Indexstruktur, um aus einer Datenbank mögliche Matching-Kandidaten für die gegebene bibliographische Referenz zu extrahieren. Für jeden gefundenen Kandidaten wird durch ein approximatives Matching auf Felderebene ein Konfidenzwert berechnet.

3.3 Abgleich von Datenbankeinträgen mit Datenbankeinträgen

Die Aufgabe, Einträge von bibliographischen Datenbanken untereinander abzugleichen, steht indirekt mit bibliographischen Referenzen in Verbindung: Findet vor dem Abgleich einer Referenz mit einer Datenbank eine erfolgreiche Strukturanalyse statt, wird das Problem auf einen Abgleich zweier Datenbankeinträge reduziert. Die analysierte Referenz kann in diesem Fall als strukturierter Eintrag aufgefasst werden.

Das Matching von Datenbankeinträgen ist Gegenstand zahlreicher Untersuchungen und unter verschiedensten Namen bekannt, u. a. *Merge/Purge* (Hernandez & Stolfo, 1998), *Record-Linkage* (Winkler, 1995) oder *Detecting duplicate database records* (Monge & Elkan, 1997). Eine ausführliche Behandlung dieses Themas ist im Rahmen dieses Artikels nicht möglich; einen guten Überblick gibt Gu et al.

[7] Ein Beispiel für eine solche Anwendung findet sich in McCallum et al. (2000a).

[8] Dieses Verfahren weist große Ähnlichkeiten mit dem *Citation-Matching*-Algorithmus von Citeseer (Lawrence et al., 1999b) auf.

(2003). Mit speziell in bibliographischen Datenbanken auftretenden Problemen beschäftigen sich Ayres et al. (1988) und Hylton (1996).

4 Zusammenfassung und Ausblick

Wie dieser kurze Überblick zeigt, birgt das Gebiet der bibliographischen Referenzen interessante Probleme für die maschinelle Sprachverarbeitung. Dabei können Techniken aus unterschiedlichsten Bereichen der Computerlinguistik und der Informationsextraktion eingesetzt werden. Auch wenn bereits zahlreiche Lösungsansätze für die Verarbeitung von Quellenangaben existieren, lassen Anwendungsmöglichkeiten unter anderem im Bereich der Datenintegration und der OCR-Nachkorrektur eine weitere Erforschung dieses Themengebiets erhoffen.

Literatur

Adelberg, B., 1998. *NoDoSE - A Tool for Semi-Automatically Extracting Semi-Structured Data from Text Documents*. In: ACM International Conference on Management of Data (SIGMOD). Seattle, 283–294.

Agichtein, E., Ganti, V., 2004. *Mining Reference Tables for Automatic Text Segmentation*. In: 10th ACM Conference on Knowledge Discovery and Data Mining (SIGKDD). Seattle, 20–29.
URL `citeseer.ist.psu.edu/agichtein04mining.html`

Anderl, E., 2005. *Computerlinguistische Analyse bibliographischer Referenzen*. Magisterarbeit, Ludwig-Maximilians-Universität München.

ANSI, 2005. *Bibliographic References – ANSI/NISO Z39.29-2005*. American National Standards Institute: NISO Press, Bethesda.

APA, 2001. *Publication Manual of the American Psychological Association*, 5th Edition. American Psychological Association, Washington, D.C.

Ayres, F. H., Huggill, J. A. W., Yannakoudakis, E. J., 1988. The Universal Standard Bibliographic Code (USBC): Its Use for Clearing, Merging and Controlling Large Databases. *Program – Automated Library and Information Systems* **22** (2), 117–132.

Bergmark, D., 2000. *Automatic Extraction of Reference Linking Information from Online Documents*. Tech. Rep. CSTR 2000-1821, Cornell University.

Bergmark, D., Lagoze, C., 2001. *An Architecture for Automatic Reference Linking*. In: 5th European Conference on Research and Advanced Technology for Digital Libraries (ECDL). Darmstadt, 115–126.
URL `citeseer.ist.psu.edu/bergmark01architecture.html`

Berkowitz, E., Elkhadiri, M. R., 2004. *Creation of a Style Independent Intelligent Autonomous Citation Indexer to Support Academic Research*. In: 15th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS). Chicago, 68–73.

Besagni, D., Belaïd, A., 2004. *Citation Recognition for Scientific Publications in Digital Libraries*. In: 1st International Workshop on Document Image Analysis for Libraries (DIAL). Palo Alto, 244–252.

Besagni, D., Belaïd, A., Benet, N., 2003. *A Segmentation Method for Bibliographic References by Contextual Tagging of Fields*. In: 7th International Conference on Document Analysis and Recognition (ICDAR). Edinburgh, 384–388.

Borkar, V., Deshmukh, K., Sarawagi, S., 2001. *Automatic Segmentation of Text into Structured Records*. In: ACM International Conference on Management of Data (SIGMOD). Santa Barbara, 175–186.

Califf, M. E., Mooney, R. J., 1999. *Relational Learning of Pattern-Match Rules for Information Extraction*. In: 16th National Conference on Artificial Intelligence and 11th Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI). Orlando, 328–334.

Citeseer, 2005. *Citeseer*. `http://citeseer.ist.psu.edu` (Stand: 12.8.2005).

Claivaz, J.-B., Meur, J.-Y. L., Robinson, N., 2001. From Fulltext Documents to Structured Citations: CERN's Automated Solution. *High Energy Physics Libraries Webzine* **5**.
URL `http://library.cern.ch/HEPLW/5/papers/2/`

CMS, 2003. *The Chicago Manual of Style*, 15th Edition. University of Chicago Press, Chicago.

Connan, J., Omlin, C., 2000. *Bibliography Extraction with Hidden Markov Models*. Tech. Rep. US-CS-TR-00-6, Department of Computer Science, University of Stellenbosch.
URL `citeseer.ist.psu.edu/connan00bibliography.html`

Crossref, 2005. *crossref.org:: the reference linking backbone*. `http://www.crossref.org/` (Stand: 12.8.2005).

Day, M.-Y., Tsai, T.-H., Sung, C.-L., Lee, C.-W., Wu, S.-H., Ong, C.-S., Hsu, W.-L., 2005. *A Knowledge-Based Approach to Citation Extraction*. In: IEEE International Conference on Information Reuse and Integration (IEEE IRI), im Druck. Las Vegas.

Demleitner, M., Kurtz, M., Accomazzi, A., Eichhorn, G., Stern-Grant, C., Murray, S. S., 2004. Automated Resolution of Noisy Bibliographic References. *CoRR* **cs.DL/0401028**.

DIN, 1984. *DIN 1505-2 Titelangaben von Dokumenten: Zitierregeln*. Deutsches Institut für Normung e.V.

DIN, 1995. *DIN 1505-3 Titelangaben von Dokumenten Verzeichnisse zitierter Dokumente (Literaturverzeichnisse)*. Deutsches Institut für Normung e.V.

Ding, Y., Chowdhury, G., Foo, S., 1999. *Template Mining for the Extraction of Citation from Digital Documents*. In: 2nd Asian Digital Library Conference. Taiwan, 47–62.
URL `citeseer.ist.psu.edu/ding99template.html`

Doi, 2005. *The Digital Object Identifier System*. `http://www.doi.org/` (Stand: 12.8.2005).

Fine, S., Singer, Y., Tishby, N., 1998. The Hierarchical Hidden Markov Model: Analysis and Applications. *Machine Learning* **32** (1), 41–62.
URL `citeseer.ist.psu.edu/fine98hierarchical.html`

Garfield, E., 1979. *Citation Indexing: Its Theory and Application in Science, Technology, and Humanities*. Wiley, New York.

Geng, J., 2002. *Automatic Extraction and Integration of Bibliographic Information on the Web Using Hidden Markov Models*. Masterarbeit, Duke University.

Geng, J., Yang, J., 2004. *AUTOBIB: Automatic Extraction of Bibliographic Information on the Web*. In: 8th International Database Engineering and Applications Symposium (IDEAS). Coimbra, 193–204.

Gu, L., Baxter, R., Vickers, D., Rainsford, C., 2003. *Record Linkage: Current Practice and Future Directions*. Tech. Rep. 03/83, CSIRO Mathematical and Information Sciences.

Han, H., Giles, C. L., Manavoglu, E., Zha, H., Zhang, Z., Fox, E. A., 2003. *Automatic Document Metadata Extraction Using Support Vector Machines*. In: 3rd ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL). Houston, 37–48.

Heringer, H. J., 1993. *Dependency Syntax – Basic Ideas and the Classical Model*. In: J. Jacobs, A. von Stechow, W. Sternefeld, T. Vennemann (eds.), *Syntax – An International Handbook of Contemporary Research*. Vol. 1. Walter de Gruyter, Berlin, 298–316.

Hernandez, M. A., Stolfo, S. J., 1998. Real-world Data Is Dirty: Data Cleansing and the Merge/Purge Problem. *Data Mining and Knowledge Discovery* **2** (1), 9–37.
URL `citeseer.ist.psu.edu/article/hernandez98realworld.html`

Hylton, J. A., 1996. *Identifying and Merging Related Bibliographic Records*. Tech. Rep. MIT/LCS/TR-678, Massachusetts Institute of Technology.
URL `citeseer.ist.psu.edu/hylton96identifying.html`

ISO, 1987. *Documentation – Bibliographic references – Content, form and structure 690:1987*. International Organization for Standardization (ISO).

Jewell, M., 2003. *ParaTools 1.00 Documentation*. `http://paracite.eprints.org` (Stand: 12.8.2005).

Kratzer, M., 2002. *Automatic Reference Linking by Means of MR Look-up*. `http://www.exp-math.uni-essen.de/algebra/veranstaltungen/kratzer.pdf`.

Laender, A., Ribeiro-Neto, B., Silva, A., Teixeira, J., 2002. A Brief Survey of Web Data Extraction Tools. *SIGMOD Record* **31** (2), 84–93.
URL `citeseer.ist.psu.edu/laender02brief.html`

Lawrence, S., Bollacker, K., Giles, C. L., 1999a. *Autonomous Citation Matching*. In: 3rd International Conference on Autonomous Agents (AGENTS). Seattle, 392–393.

Lawrence, S., Giles, C. L., Bollacker, K., 1999b. Digital Libraries and Autonomous Citation Indexing. *IEEE Computer* **32** (6), 67–71.

Marthi, B., Milch, B., Russell, S., 2003. *First-order probabilistic models for information extraction*. In: IJCAI Workshop on Learning Statistical Models from Relational Data. Acapulco.
URL `citeseer.ist.psu.edu/619286.html`

McCallum, A., Nigam, K., Ungar, L. H., 2000a. *Efficient Clustering of High-Dimensional Data Sets with Application to Reference Matching*. In: 6th ACM Conference on Knowledge Discovery and Data Mining (SIGKDD). New York, 169–178.

McCallum, A. K., Nigam, K., Rennie, J., Seymore, K., 2000b. Automating the Construction of Internet Portals with Machine Learning. *Information Retrieval* **3** (2), 127–163.
URL `citeseer.ist.psu.edu/mccallum00automating.html`

MLA, 2003. *MLA Handbook for Writers of Research Papers*, 6th Edition. Modern Language Association of America, New York.

Monge, A. E., Elkan, C., 1997. *An Efficient Domain-Independent Algorithm for Detecting Approximately Duplicate Database Records*. In: 2nd Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD). Tucson, 23–29.
URL `citeseer.ist.psu.edu/monge97efficient.html`

Okada, T., Takasu, A., Adachi, J., 2004. *Bibliographic Component Extraction Using Support Vector Machines and Hidden Markov Models*. In: 8th European Conference on Research and Advanced Technology for Digital Libraries (ECDL). Bath, 501–512.

Ortyl, P., Pfingstl, S., 2004. *Extrahierung bibliographischer Daten aus dem Internet*. In: 34. GI Jahrestagung. Vol. 2. Ulm, 203–207.

Parmentier, F., 1998. *Spécification d'une architecture émergente fondée sur le raisonnement par l'analogie: Application aux références bibliographiques*. Ph.D. dissertation, Université Henri Poincaré – Nancy 1.

Parmentier, F., Belaïd, A., 1997. *Logical Structure Recognition of Scientific Bibliographic References*. In: 4th International Conference on Document Analysis and Recognition (ICDAR). Ulm, 1072–1076.
URL `citeseer.ist.psu.edu/parmentier97logical.html`

Pasula, H., Marthi, B., Milch, B., Russell, S., Shpitser, I., 2002. Identity uncertainty and citation matching. *Advances in Neural Information Processing* **15**, 1401–1408.
URL `citeseer.csail.mit.edu/pasula03identity.html`

Peng, F., McCallum, A., 2004. *Accurate Information Extraction from Research Papers using Conditional Random Fields*. In: Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT-NAACL). Boston, 329–336.

Rabiner, L., 1989. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. In: Proceedings of the IEEE. Vol. 77(2). 257–285.

Schölkopf, B., Smola, A., 2002. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. MIT Press, Cambridge.

Soderland, S., 1999. Learning Information Extraction Rules for Semi-Structured and Free Text. *Machine Learning* **34** (1-3), 233–272.
URL `citeseer.csail.mit.edu/soderland99learning.html`

Takasu, A., 2003. *Bibliographic Attribute Extraction from Erroneous References Based on a Statistical Model*. In: 3rd ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL). Houston, 49–60.

Wallach, H. M., 2004. *Conditional Random Fields: An Introduction*. Tech. Rep. MS-CIS-04-21, Department of Computer and Information Science, University of Pennsylvania.

Winkler, W. E., 1995. *Matching and Record Linkage*. In: B. G. Cox, D. A. Binder, B. N. Chinnappa, A. Christianson, M. J. Colledge, P. S. Kott (eds.), *Business Survey Methods*. John Wiley & Sons, New York, Ch. 11, 355–384.

Yin, P., Zhang, M., Deng, Z.-H., Yang, D., 2004. *Metadata Extraction from Bibliographies Using Bigram HMM*. In: 7th International Conference on Asian Digital Libraries (ICADL). Shanghai, 310–319.

Die Nutzung experimentalphonetischer Messdaten zur audiovisuellen Sprachsynthese

Caroline Clemens
Institut für Sprache und Kommunikation
Sascha Fagel
Technische Universität Berlin

Zusammenfassung

Die Autoren erläutern, wie die Artikulationsbewegungen eines Menschen erfasst und die Messergebnisse in der Entwicklung eines Talking Heads umgesetzt wurden. Die Qualität audiovisueller Sprachsynthese hängt stark davon ab, wie realitätsnah die Synthese gelingt. Während die Entwicklung auditiver Sprachsynthese stark fortgeschritten ist, ist die visuelle Synthese vieler Talking Heads unbefriedigend. MASSY ist ein audiovisueller Sprachsynthesizer, der aus einem beliebigen Text hör- und sichtbare Sprache erzeugt. Damit die Sprechbewegungen von MASSY realistisch aussehen, dienen die Sprechbewegungen eines echten Menschen als Vorbild. Die Messungen erfolgten mit einem optischen und einem elektromagnetischen Motion-Capture-Verfahren, der Elektromagnetischen Artikulographie.

1 Audiovisuelle Sprachsynthese

In der Mensch-Maschine-Kommunikation finden immer häufiger Schnittstellen mit Sprachausgabe Verwendung. Dabei wird die natürliche, mündliche Sprache des Menschen nachgeahmt. Audiovisuelle Sprachsynthese kommt bereits in zahlreichen Applikationen zur Anwendung:

- In der multimedialen Lehre: Im modernen Fremdsprachenunterricht unterstützt Lernsoftware beim Erlernen der richtigen Aussprache. Die Lernenden sprechen etwas ein und bekommen die richtige Aussprache als Verbesserungsvorschlag vorgespielt. Auch können Einzelheiten der Aussprache und Unterschiede zur Muttersprache des Lernenden verdeutlicht werden.
- In der Logopädie und Phoniatrie: Sofern der Patient in der Lage ist, Sprechbewegungen von einem Modell auf sich selbst zu übertragen, können Sprachtrainer zur Therapie eingesetzt werden. Die Software dient zum Lernen am Modell und der visuellen Rückmeldung (z.B. SpeechTrainer, Kröger 2006).
- In Avataren: Die Anwendungsmöglichkeiten reichen von Figuren in Computerspielen über virtuelle Agenten in audiovisuellen Informationssystemen (z.B. in Museen) bis hin zum E-Mail-Programm, das die E-Mails vorliest.
- Zur Übersetzung von Filmen: In hochwertigen Animationsfilmen sind die Sprechbewegungen der Figuren sehr realistisch – aber leider nur in der Originalsprache. Bei der Übersetzung in andere Sprachen könnten die Sprechbewegungen der Zielsprache angepasst werden.
- In der Bildtelefonie: Aus dem per Telefon übertragenen Audio-Sprachsignal können die Artikulationsbewegungen geschätzt und ein künstliches Gesicht animiert werden (Karlsson et al., 2003).

2 MASSY

MASSY (Akronym aus **M**odular **A**udiovisual **S**peech **SY**nthesizer) wurde an der TU Berlin entwickelt (Fagel & Sendlmeier, 2003). Es ist ein Text-To-audiovisual-Speech-System (TTavS), das aus einem beliebigen Text synthetische Sprache erzeugt (Abbildung 1). Der Text für den Input kann aus einer Datei eingelesen oder in die Eingabemaske (Abbildung 6) von MASSY geschrieben werden.[1]

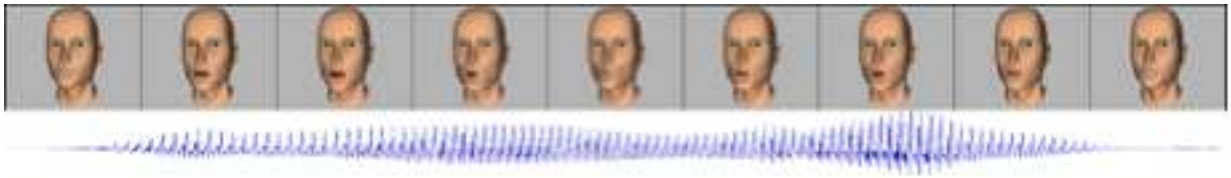

Abbildung 1: MASSY erzeugt aus einem beliebigen Text synthetische audiovisuelle Sprache.

Eine Besonderheit MASSYs ist sein modularer Aufbau (Abbildung 2). Üblicherweise können bei audiovisuellen Sprachsynthesesystemen bildgebende Verfahren, Audiosynthese und die Algorithmen zur Steuerung der Audio- und Videosynthese nicht ausgetauscht werden. MASSY bietet die Möglichkeit, Module auszutauschen, wie die Module zur Audio- und Videosynthese, die Vorverarbeitungsstufe und deren Submodule zur Erzeugung emotionaler Sprache und zur Integration nonverbaler Elemente. So ist es möglich, einzelne alternative Module miteinander zu vergleichen.

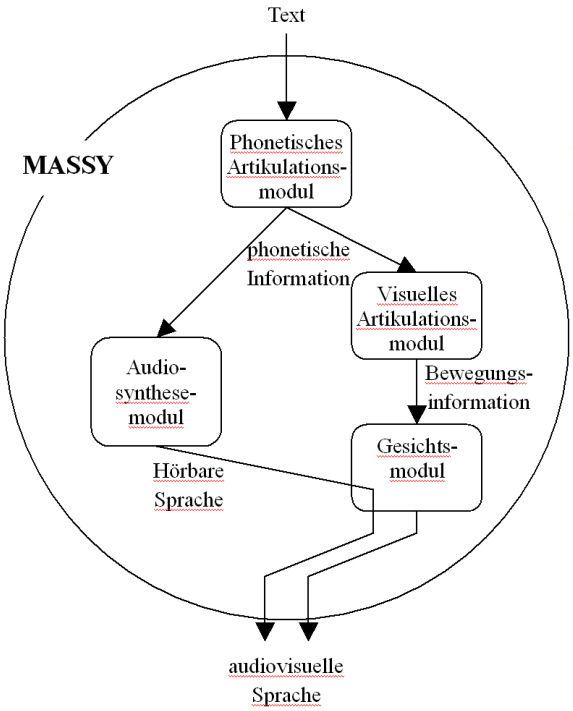

Abbildung 2: Systemübersicht über den modularen Aufbau von MASSY.

[1] MASSY online ausprobieren auf der Website: `http://avspeech.info`.

Massy kann unterschiedliche Gesichtsmodelle verwenden. Das Standard-Gesichtsmodell von MASSY ist ein dreidimensionaler künstlicher Kopf (Abbildung 3, 4).

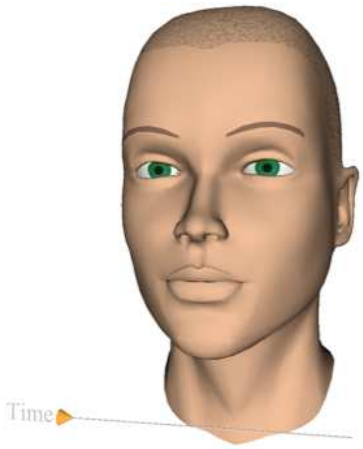

Abbildung 3: Das Standardgesichtsmodell von MASSY.

Dieses Standard-Gesichtsmodell besteht aus einem statischen und einem dynamischen Teil. Der statische Teil für den unbewegten Kopf enthält u.a. die 3.000 3D-Koordinaten der Eckpunkte der Gesichtshaut und der Zunge. Der dynamische Teil für die virtuellen Artikulatoren hat einen 3D-Vektor je Eckpunkt je Artikulationsparameter (siehe Kapitel „Verwendung der Messdaten“) zur Beschreibung der Bewegung.

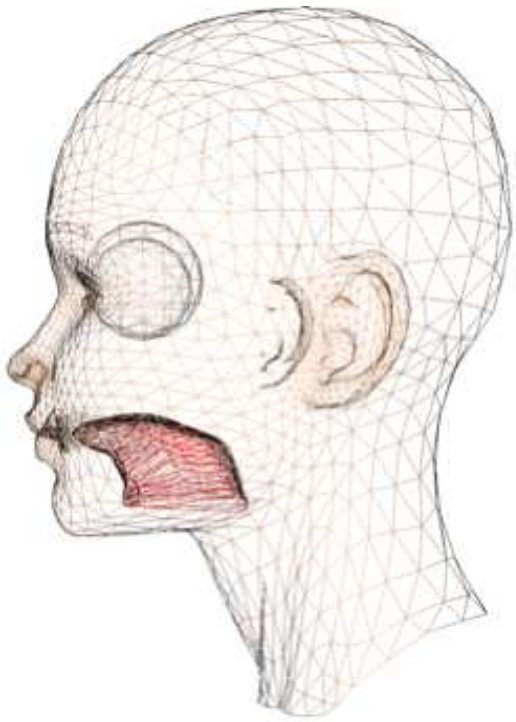

Abbildung 4: Im Standardgesichtsmodell von MASSY ist die Netzstruktur der 3D-Koordinaten gut erkennbar.

Zusätzlich wurde ein Gesichtsmodell zur Erzeugung eines Videos aus einem oder mehreren Fotos eines realen Gesichts implementiert (Abbildung 5). Hierfür werden zunächst die Artikulationsparameter in Eigenschaften des Gesichts umgerechnet, wobei eine Reduktion auf die zwei Eigenschaften Lippenbreite und Lippenrundung sowie Unterkieferhöhe und Lippenhöhe stattfindet. Aus diesen Eigenschaften des Gesichts werden für jedes Einzelbild des zu erzeugenden Videos bei einer wählbaren Bildrate (z.B. 25 Bilder pro Sekunde) die Werte der Lippenbreite/-rundung und Lippen-/Unterkieferhöhe linear interpoliert. Dann wird ein vorhandenes Bild so verformt, dass es die entsprechenden Eigenschaften annimmt. Details dieses Algorithmus' und eine Bewertung des Verfahrens finden sich in Fagel (2005).

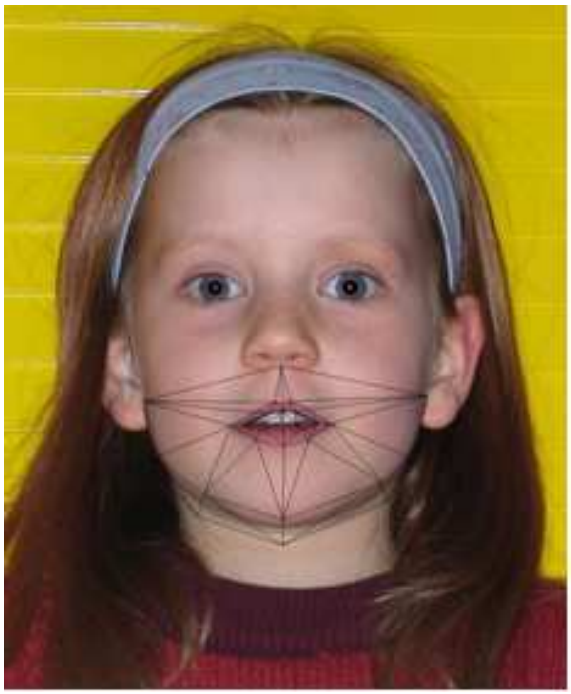

Abbildung 5: Foto als bildbasiertes Gesichtsmodell.

Die audiovisuelle Sprachsynthese von MASSY kann durch zahlreiche Auswahlfelder variiert werden (Abbildung 6):

- Sprache: Text-zu-Phonem-Übersetzung in *Deutsch* oder *Englisch*.

Audio-Einstellungen:

- Geschlecht: Grundfrequenzkontur *weiblich* oder *männlich*.
- Stimme: *Deutsche*, *englische* oder *tschechische weibliche* oder *männliche* Stimmen. Sprachen sind experimentell mischbar. *auto* wählt eine zu Sprache und Geschlecht passende Stimme. Es sind 14 Stimmen vorhanden.
- Sprechtempo: *Sehr langsam* bis *sehr schnell*. Bewirkt lineare Skalierung aller Lautdauern.
- F0-Umfang: Vergrößert den Grundfrequenzumfang linear.
- F0-Shift: Hebt oder senkt die Grundfrequenz prozentual.

Gesichtseinstellungen:

- Artikulation: *Dominanzmodell*[2] oder *Di-Visem* Algorithmus (siehe Kapitel „Steuerung der Sprachvisualisierung“).
- Hyperartikuliert: *Stark* bis *sehr schwach*. Vergrößert oder verkleinert die Bewegungsamplitude der Artikulatoren.
- Gesichtsmodell: *VRML* (Erzeugung dauert länger, mehr Daten) oder *VRML displacers* (geht schneller, weniger Bilder pro Sekunde) in 3D, *image* basiert auf einem Foto, das verformt wird, *image-db* verwendet Einzelbilder, die aus einer Datenbasis ausgewählt werden.
- Transparenz: *Opak* (= undurchsichtig) bis *transparent* (nur vrml und vrml-disp).
- Emotion: *Freudig*, *ärgerlich* oder *traurig* (nur vrml und vrml-disp).
- *jpg* kann ein selbst erstelltes Gesicht oder Foto sein (nur *image*).

Spezielle Einstellungen:

- Prüft 3D-plugin: *an* testet bei jedem Aufruf, ob das VRML-plugin installiert ist. *aus* ist schneller.
- Kompression: an (schneller, weil geringere Datenmengen) oder aus (VRML-Dateien sind im Text-Editor lesbar).

Spezielle Einstellungen (McGurk):

- ersetze Phone [ ] mit [ ]: Ersetzt die Laute im ersten Feld durch diejenigen im zweiten Feld – aber nur im Audiokanal!

[2] Erklärung siehe Fagel & Clemens (2004).

Abbildung 6: Auswahlfelder von MASSY.

3 Motion-Capture-Verfahren

Es sollen die sichtbaren Artikulationsbewegungen beim Sprechen erfasst werden, also solche im oberen Ansatzrohr. Die experimentelle Phonetik bietet unterschiedliche Verfahren zur Erfassung von Sprechbewegungen, die im Körper eines Sprechers stattfinden.

Röntgenaufnahmen müssen wegen der gesundheitlichen Risiken für den Sprecher ausgeschlossen werden. Auch ist das Bildmaterial nicht ausreichend aussagekräftig. Weichteile wie Zunge und Lippen sind nur undeutlich zu erkennen. Gerade diese Körperteile sollen bei den Messungen erfasst werden. Die Magnet-Resonanz-Tomographie (MRT; englisch MRI – Magnetic Resonance Imaging) wird zur Visualisierung von Strukturen im Körper verwendet. Die Geräte werden überwiegend in klinischen Einrichtungen verwendet. Gemessen wird die Kernspinresonanz der Atome, deren Kerne sich, wenn ein äußeres Magnetfeld angelegt wird, ausrichten. Im Gegensatz zum Röntgen-Verfahren kommt die MRT ohne gefährliche Strahlung aus. Der gesamte Vokaltrakt vom Kehlkopf bis zu den Lippen wird betrachtet. Allerdings können Bewegungen einzelner Punkte nicht verfolgt werden. Bei der optischen Motion-Capture-Methode werden Bewegungen von Markierungen erfasst. Spezialkameras nehmen das von den Markierungen im Gesicht der Versuchsperson gesendete oder reflektierte Licht auf. Aus den gewonnenen Daten werden die Bewegungen der Marker errechnet. Das Verfahren schränkt die Bewegung der Versuchsperson nicht ein,

bietet eine hohe Messgenauigkeit, ist aber aufwändig wegen der notwendigen Nachbearbeitung der Daten. Innere oder häufig verdeckte Artikulatoren können nicht erfasst werden. Bei einem Bilderfassungssystem wird zunächst eine Videoaufnahme gemacht. Die Videoaufzeichnung wird digitalisiert, und ein Computerprogramm extrahiert die Bewegungen bestimmter Körperpartien. Während der Aufnahme muss der Kopf unbeweglich verharren, damit Abstand und Winkel zur Kamera unverändert bleiben.

Gewählt wurde das Verfahren der Elektromagnetischen Artikulographie (EMA[3]). Bei dieser Motion-Capture-Methode wird die Bewegung von Punkten auf der Oberfläche der Artikulatoren gemessen (zur EMA siehe Schönle et al. 1987; Gröne et al. 1992 und Perkell et al. 1992). Dazu trägt die Testperson einen Helm mit drei Sendespulen (jede mit spezifischer Frequenz zwischen 10–30kHz), die ein elektromagnetisches Wechselfeld erzeugen. Empfängerspulen (= Sensoren) sind auf die Artikulatoren der Testperson geklebt und befinden sich beim Sprechen im elektromagnetischen Wechselfeld. In den Empfängerspulen werden dabei elektrische Spannungen induziert. Die Größe der induzierten Spannung ist abhängig von den Abständen zu den drei Sendespulen, da die Wechselfelder inhomogen sind: die Feldstärke nimmt mit dem Quadrat des Abstands zur Sendespule ab. 200-mal pro Sekunde, also in Zeitabständen von 5 Millisekunden messen die Sensoren die induzierte Spannung. Die Sensoren sind durch dünne Drähte, die aus dem Mund der Testperson herausführen, mit dem Computer verbunden. Aus den gemessenen Spannungswerten werden die Abstände der Sensoren von den Sendespulen errechnet. Die erreichte räumliche Genauigkeit bei der Bestimmung der Spulenpositionen zu den einzelnen Zeitpunkten beträgt $\pm$0,12mm. Ein Abbild der Artikulatorbewegungen erhält man, wenn die einzelnen Positionen nacheinander als Funktion der Zeit aufgetragen werden. Die EMA ist ein wichtiges Werkzeug der experimentellen Phonetik im Bereich der Sprachproduktion geworden, da sie präzise, aussagekräftige Messwerte liefert. Da es ohne gesundheitsgefährdende Methoden auskommt, ist das Verfahren sicher (Hasegawa-Johnson, 1998) und wird auch zunehmend bei Kindern angewendet.

4 Messungen

Die Wahl fiel auf die Elektromagnetische Artikulographie. Die Sprechbewegungen eines Menschen sollten gemessen und die Messwerte dann zur Synthese der Sprechbewegungen benutzt werden. Das ZAS (Zentrum für Allgemeine Sprachwissenschaft, Typologie und Universalienforschung – außeruniversitäres Forschungsinstitut des Landes Berlin) stellte dafür seinen Elektromagnetischen Artikulographen bereit. Die Mitarbeiter des ZAS unterstützten die Planung, Durchführung und Nachbearbeitung der Messungen.

Der 2D-Artikulograph AG 100 im Phonetiklabor des ZAS kann Artikulationsbewegungen in zwei Dimensionen erfassen (inzwischen gibt es auch 3D-Artikulographen, die jedoch wenig verbreitet sind). Mit den EMA-Messungen werden deshalb keine Informationen über Lippenrundung und -spreizung gewonnen, da nur Bewegungen in der mediosagittalen Ebene erfasst werden. Deshalb wurde zusätzlich eine Videoaufnahme des Gesichts durchgeführt. Die Auswertung der Videoaufnahmen erfolgte mit einem Computerprogramm, das automatisch innere und äußere Lippenhöhe und -breite ermittelt. So können Lippenspreizung und Lippenrundung untersucht werden. Die Versuchssituation und die Apparatur des EMA können die Natürlichkeit der Sprechweise mindern.

Die Artikulatorpositionen eines Lautes sind abhängig von denen der Nachbarlaute. Diese Nachbarlaute lassen sich wiederum durch deren Nachbarlaute beeinflussen usw. Es galt, durch die Messungen festzustellen, wie die Artikulatorpositionen zweier benachbarter Laute sich durch Koartikulation beeinflussen. Damit der Einfluss weiterer Laute in der Umgebung ausgeschlossen wurde, wurden bei den Aufnahmen nur Kombinationen zweier Laute verwendet. Es wurden solche Laute eingesprochen, die visuell unterscheidbar sind, da ausschließlich sichtbare Artikulationsbewegungen zu synthetisieren sind. Laute ohne nennenswerten sichtbaren Unterschied wurden zu einem Visem zusammengefasst.

[3]Die Abkürzung EMA steht sowohl für den Namen des Verfahrens, die Elektromagnetische Artikulographie, als auch für das verwendete Gerät, den Elektromagnetischen Artikulographen.

Abbildung 7: Messaufbau. Zu sehen sind das Mikrofon, die Videokamera, der Bildschirm, auf dem die zu sprechenden Pattern erscheinen und die drei Sendespulen am Helm des EMA.

Der Kopf der Testperson wurde im Helm des EMA fixiert (Abbildung 7 und 8). Eine Aufhängevorrichtung an der Zimmerdecke hielt das Gewicht der Apparatur und reduzierte die Kopfbewegungen.

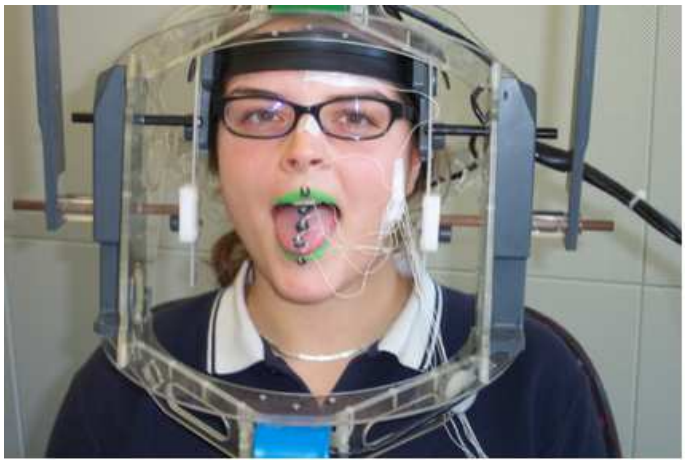

Abbildung 8: EMA-Apparatur und Sprecherin mit den aufgeklebten Messspulen (weiß) und den darunter liegenden Seidenstoffstücke (schwarz). Am unteren Bildrand ist eine der drei Sendespulen erkennbar.

Acht Empfängerspulen wurden am Kopf bzw. im Mund der Testperson angebracht. Ihre Positionen sind in den Abbildungen 7 und 8 zu sehen und werden in Tabelle 1 beschrieben. In Abbildung 7 sind eine Spule an den unteren Schneidezähnen sowie eine an den oberen Schneidezähnen nicht sichtbar, da sie von den Lippen verdeckt sind. Die Spulen 3 bis 8 erfassten Bewegungen der artikulierenden Organe, während die Empfängerspulen 1 und 2 als Referenzpunkte dienten. Diese waren an der Nasenwurzel und an den oberen Schneidezähnen befestigt, um ungewünschte Bewegungen des Kopfes zu messen, die nach dem Experiment aus den Bewegungsdaten der artikulierenden Organe herausgerechnet werden mussten.

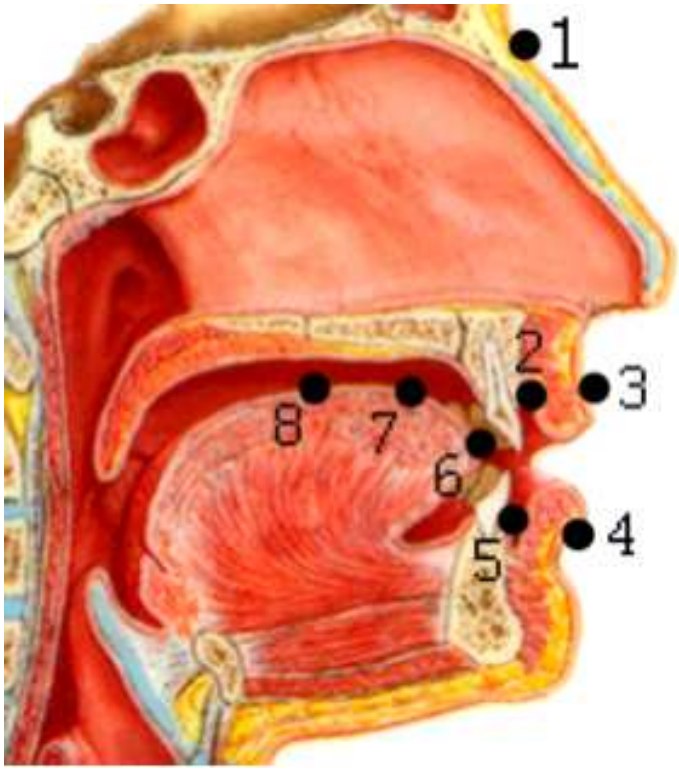

Abbildung 9: Positionen der Empfängerspulen (Grafik des Mediosagittalschnitts aus Wirth 1994, ergänzt durch die nummerierten Punkte). Die Spulen 1 und 2 sind Referenzpunkte, die Spulen 3 bis 8 sind auf beweglichen Artikulatoren befestigt.

Da der AG 100 Messungen nur in 2 Dimensionen ermöglicht werden alle Empfängerspulen in der mediosagittalen Ebene befestigt. Die flachen Empfängerspulen haben eine Grundfläche von 1 mm². Abbildung 8 zeigt die Versuchsperson mit den aufgeklebten Spulen. Kleine Stücke schwarzen Seidenstoffs wurden zwischen die weißen Spulen und die Haut gelegt, um die Auflagefläche des medizinischen Klebstoffs zu erhöhen. Der verwendete medizinische Klebstoff ist schnelltrocknend, hält auch auf Weichteilen und Schleimhäuten und läßt sich einfach wieder ablösen[4]. Drähte verbinden die Empfängerspulen mit dem Rechner. Sie sind dünn und biegsam und behindern nicht beim Sprechen. Die Empfängerspulen stören ebenfalls nicht, allerdings kann die Spule auf der Zungenspitze anfangs für die Versuchsperson irritierend sein, da sie bei Enge- und Verschlussbildungen alveolarer Laute zu spüren ist. Um einen großen farblichen Kontrast zu der umgebenden Hautfarbe zu erzielen wurden die Lippen der Testperson grün geschminkt. Dies verbesserte bei der späteren Analyse des Videomaterials die Erkennung der Lippenkonturen.

[4]Dennoch ist ggf. eine Enthaarung der betreffenden Körperstellen (Spule 3!) vor dem Ankleben ratsam, um Schmerzen beim Ablösen der Spulen zu vermeiden, da sich der Klebstoff schlecht von Haaren löst.

Spule	Position	Funktion
1	Nasenwurzel	Referenzspulen
2	obere Schneidezähne	
3	Oberlippe außen	Spulen auf beweglichen Artikulatoren
4	Unterlippe außen	
5	untere Schneidezähne	
6	Zungenspitze	
7	Zungenblatt	
8	Zungenrücken	

Tabelle 1: Positionen und Funktionen der Empfängerspulen

Der Versuchsleiter hält sich im Nebenraum auf. Die Räume sind schallisoliert, damit die Aufnahmen nicht gestört werden. Die Kommunikation zwischen Versuchsleiter und Versuchsperson ist durch ein Sichtfenster sowie Mikrofon und Lautsprecher möglich. Ein Bildschirm zeigt der Versuchsperson die zu sprechenden Lautfolgen an.

Die Versuchsperson konnte den Bildschirm bequem sehen ohne den Kopf bewegen zu müssen[5]. Die Videokamera hat das Gesicht frontal gefilmt und wurde so positioniert, dass sie die Sicht der Versuchsperson auf den Bildschirm nicht behindert hat. Das Mikrofon durfte wegen der Videoaufnahmen das Gesicht der Versuchsperson nicht verdecken, musste sich aber in möglichst geringem Abstand zum Mund befinden. Der Beginn einer neuen Aufnahme wurde der Versuchsperson durch einen Signalton angezeigt. Jede Aufnahme dauerte 2 Sekunden, danach folgen einige Sekunden Pause. Alle Lautfolgen wurden zunächst in normaler Sprechweise aufgenommen. Es folgte ein zweiter Durchgang, in dem hyperartikuliert gesprochen wurde.

Im Folgenden wird das phonetische Transkriptionssystem IPA in der Notation SAMPA[6] verwendet. Für die Aufnahmen wurden 9 Konsonanten (/m/, /n/, /N/, /j/, /R/, /v/, /z/, /Z/, /l/) mit 15 Vokale des Deutschen (/a/, /e/, /i/, /o/, /u/, /E/, /y/, /2/, /I/, /O/, /U/, /9/, /Y/, /6/, /@/) kombiniert. Eine Beschreibung der Laute zeigt Tabelle 2.

[5] Vor dem Beginn von EMA-Aufnahmen sollte der Sitzposition genügend Aufmerksamkeit geschenkt werden, da die Testperson in der gewählten Körperhaltung über einen längeren Zeitraum möglichst bewegungsfrei verweilen muss.

[6] Das maschinenlesbare phonetische Alphabet SAMPA stellt IPA-Lautsymbole mit Zeichen des ASCII Codes dar.

Beschreibung		**SAMPA-Zeichen**	**IPA-Zeichen**
Vokale	tief, ungerundet	a	a
	obermittelhoch, vorne, ungerundet	e	e
	hoch, vorne, ungerundet	i	i
	obermittelhoch, hinten, gerundet	o	o
	hoch, hinten, gerundet	u	u
	untermittelhoch, vorne, ungerundet	E	ɛ
	hoch, vorne, gerundet	y	y
	obermittelhoch, vorne, gerundet	2	ø
	halbhoch, vorne-zentral, ungerundet	I	ɪ
	untermittelhoch, hinten, ungerundet	O	ɔ
	halbhoch, zentral-hinten, gerundet	U	ʊ
	untermittelhoch, vorne, gerundet	9	œ
	halbhoch, vorne-zentral, gerundet	Y	ʏ
	halbtief, zentral	6	ɐ
	mittel, zentral	@	ə
Stimmhafte Konsonanten	bilabialer Nasal	m	m
	alveolarer Nasal	n	n
	velarer Nasal	N	ŋ
	palataler Approximant	j	j
	uvularer Frikativ	R	ʁ
	labiodentaler Frikativ	v	v
	alveolarer Frikativ	z	z
	postalveolarer Frikativ	Z	ʒ
	alveolarer Lateralapproximant	l	l

Tabelle 2: Die eingesprochenen Laute mit Beschreibung, Transkriptionssymbolen des IPA und der SAMPA-Notation.

Die ausgewählten deutschen Konsonanten stehen stellvertretend für ein Visem. Ein Visem fasst visuell nicht oder kaum unterscheidbare Laute zu einer Gruppe zusammen. Bei ausschließlich visueller Perzeption sind Stimmhaftigkeit und Nasalität fast nicht zu erkennen. Daher zählen Laute eines Artikulationsortes, die sich nur durch Stimmhaftigkeit oder Nasalität unterscheiden, zum gleichen Visem. Es ist nicht auszuschließen, dass bei genauer Beobachtung geringe Unterschiede festzustellen sind. Geübte Personen, wie z.B. Menschen mit Hörstörungen, vermögen mitunter Details wahrzunehmen, die anderen Betrachtern entgehen. Eine Übersicht über die zu Visemen zusammengefassten Konsonanten gibt Tabelle 3.

Laute		**Laut, stellvertretend für Visem**
bilabial, Plosive und Nasal	p, b, m	m
alveolar, Plosive und Nasal	t, d, n	n
velar, Plosive und Nasal	k, g, N	N
labiodental, Frikative	f, v	v
alveolar, Frikative	s, z	z
postalveolar, Frikative	S, Z	Z
palatal, Frikativ und Approximant	C, j	j
velar/uvular, Frikative	x, R	R
alveolar, Lateralapproximant	l	l

Tabelle 3: Die Konsonanten des Deutschen wurden zu Visemen zusammengefasst

Das von der Sprecherin üblicherweise produzierte /R/ ist ein uvularer Frikativ und wurde als R-Variante gewählt. Der glottale Frikativ /h/ blieb unberücksichtigt, da er bei der Artikulation nicht sichtbar ist. Eine Beschreibung der Laute ist Tabelle 1 zu entnehmen. Alle 9 Konsonanten wurden mit allen 15 Vokalen kombiniert, sodass sich 135 unterschiedliche Konsonant-Vokal-Kombinationen (Tabelle 4) ergaben. Aus jeder Kombination wurde nach dem Muster /,CVCV'CVCV/ ein viersilbiges „Wort" gebildet. die erste Silbe trug beim Sprechen die Nebenbetonung und die dritte Silbe die Hauptbetonung, damit es wortmedial eine unbetonte (zweite) und eine betonte (dritte) Silbe gab. Die 135 Kombinationen wurden alle einmal normal und einmal hyperartikuliert gesprochen, so dass es 270 Aufnahmen gab.

		Konsonanten								
		m	**n**	**N**	**j**	**R**	**v**	**z**	**Z**	**I**
Vokale	**a**	ma	na	Na	ja	Ra	va	za	Za	Ia
	e	me	ne	Ne	je	Re	ve	ze	Ze	Ie
	i	mi	ni	Ni	ji	Ri	vi	zi	Zi	Ii
	o	mo	no	No	jo	Ro	vo	zo	Zo	Io
	u	mu	nu	Nu	ju	Ru	vu	zu	Zu	Iu
	E	mE	nE	NE	jE	RE	vE	zE	ZE	IE
	2	m2	n2	N2	j2	R2	v2	z2	Z2	I2
	y	my	ny	Ny	jy	Ry	vy	zy	Zy	Iy
	I	mI	nI	NI	jI	RI	vI	zI	ZI	II
	O	mO	nO	NO	jO	RO	vO	zO	ZO	IO
	U	mU	nU	NU	jU	RU	vU	zU	ZU	IU
	9	m9	n9	N9	j9	R9	v9	z9	Z9	I9
	Y	mY	nY	NY	jY	RY	vY	zY	ZY	IY
	@	m@	n@	N@	j@	R@	v@	z@	Z@	I@
	6	m6	n6	N6	j6	R6	v6	z6	Z6	I6

Tabelle 4: Die 135 verwendeten Konsonant-Vokal-Kombinationen (Transkription mit SAMPA). Bei diesen Lautfolgen kann ein Laut nur durch einen einzelnen anderen Laut beeinflusst werden. So können Koartikulationsphänomene durch einen komplexen lautlichen Kontext ausgeschlossen werden.

5 Verwendung der Messdaten

Das Datenmaterial umfasst EMA-Daten und Videodaten. Jede der 270 EMA-Aufnahmen ist 2 Sekunden lang. Pro Sekunde wurden 200-mal Messwerte erhoben. Da die horizontale und die vertikale Komponente einer Spulenposition getrennt vorliegen, gibt es zu jeder der 6 Spulen 2 Datenreihen. Somit ergibt sich eine Anzahl von $270 \cdot 2 \cdot 200 \cdot 6 \cdot 2 = 1.296.000$ Messwerten. Die Videoaufzeichnung wurde für die weitere computergestützte Verarbeitung digitalisiert. Die Videoaufnahme ist nicht in 2 Sekunden lange Abschnitte aufgeteilt wie die EMA-Daten, sondern erfolgte über den gesamten Zeitraum der Messdurchführung. So wurden auch alle Einatembewegungen erfasst. Da die Tonspur der Videokamera die Signaltöne enthält, die den Beginn der AG 100-Aufnahmen markieren, ist die Synchronisierung von EMA- und Videoaufnahmen unproblematisch.

Die Abbildung 10 zeigt als Beispiel Einzelbilder aus den Sequenzen /ama/ und /omo/. Die Realisierungen des Lautes [m] sind optisch sehr unterschiedlich abhängig davon, in welchem vokalischen Kontext sie auftreten. Hier ist ein deutlicher koartikulatorischer Einfluss der angrenzenden Vokale auf die Lippenbreite des medialen Konsonanten zu erkennen.

Abbildung 10: Bildsequenzen der Äußerungen /ama/ (oben) und /omo/ (unten). Die beiden mittleren Bilder zeigen unterschiedliche Mundformen bei der Artikulation von /m/ als Effekt der Koartikulation.

Abbildung 11 zeigt eine Darstellung der Äußerung /nananana/. Die Werte auf der X-Achse entsprechen den Abtastungen (Abtastintervall = 5 ms). Die Werte auf der Y-Achse sind in cm angegeben, wobei die Bissebene mit 0 festgelegt ist. Die erste Abwärtsbewegung des Unterkiefers und der Unterlippe wird durch Einatmen verursacht, die übrigen durch die Mundöffnung der Vokale /a/. Die größte Unterkieferöffnung in der dritten Silbe entsteht durch die Hauptbetonung dieser Silbe. Die Kurven von Unterkiefer und Unterlippe verlaufen nahezu deckungsgleich. Die linke der beiden vertikalen Linien markiert das ungefähre Zentrum der Verschlussphase des [n], die rechte die maximale Öffnung des [a].

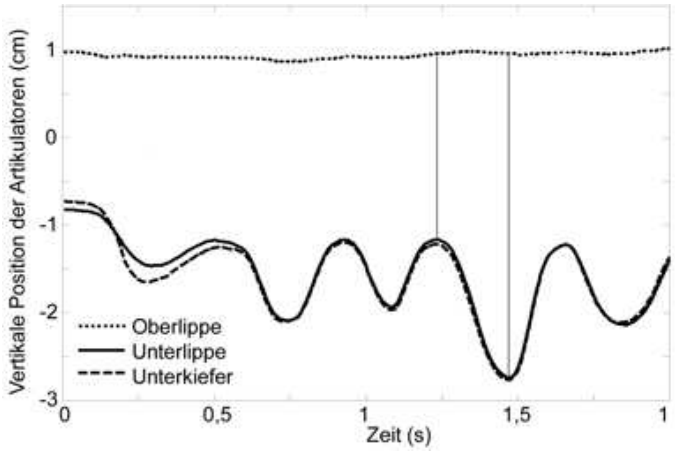

Abbildung 11: EMA-Daten der vertikalen Position der Oberlippe, der Unterlippe und des Unterkiefers während der Äußerung /nananana/.

In Abbildung 12 sind die Spulenkurven der Äußerung /mamamama/ zu sehen. Unterlippe und Unterkiefer heben und senken sich synchron. Die Bewegung der Unterlippe hat jedoch für die Bildung des bilabialen Verschlusses des [m] eine größere Amplitude, was die (partielle) Unabhängigkeit dieser beiden Artikulatoren verdeutlicht. Die Oberlippe beschreibt eine Gegenbewegung. Sie zeigt außerdem eine Plateaubildung in der Verschlussphase. Die linke der beiden vertikalen Linien markiert das ungefähre Zentrum der Verschlussphase des [m], die rechte die maximale Öffnung des [a].

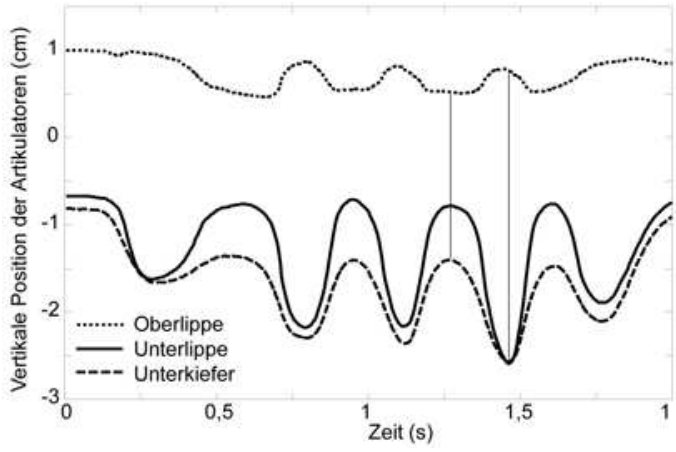

Abbildung 12: EMA-Daten der vertikalen Position der Oberlippe, der Unterlippe und des Unterkiefers während der Äußerung /mamamama/.

Aus den Messungen lässt sich mittels Linearkombinationen der Messwerte der Empfängerspulen ein Di-Visem-Modell (siehe nächstes Kapitel) ableiten. Der dritte Vokal jeder gemessenen CVCVCVCV-Sequenz weist das deutlichste Extremum der Spulenpositionen auf und wurde verwendet, um die Artikulatorkonstellation dieses Vokals im Kontext des Konsonanten zu bestimmen. Der Zeitpunkt der geringsten Veränderung der Artikulationsparameter wurde über die erste Ableitung der Artikulationsparameterfunktionen (siehe unten: Gleichungen 1 bis 6) ermittelt und für die Bestimmung der Artikulatorpositionen

verwendet. So ergaben sich jeweils die Extrempositionen der Artikulatoren nahe der zeitlichen Mitte des jeweiligen Lautes. Entsprechend wurde der dritte Konsonant für die Bestimmung der Artikulatorpositionen herangezogen. Die Spulennamen gefolgt von X bzw. Y bezeichnen deren Position auf der X- bzw. Y-Achse. Vereinfachend wurde von linearen Bewegungen der Artikulatoren ausgegangen und ihre jeweils stärkere Komponente parallel zur X- oder Y-Achse verwendet. Alle Artikulationsparameter werden auf einen Bereich zwischen 0 und 1 bzw. zwischen -1 und 1 normiert. Einige der Artikulationsparameter beschreiben keine absolute, sondern eine relative Position eines Artikulators bezüglich eines anderen. So bewegen sich Zunge und Unterlippe mit dem Unterkiefer mit, dessen Anteil herausgerechnet werden muss. Zunächst wurden nach geometrischen Eigenschaften der Sprecherin entsprechende Koeffizienten gewählt (Abbildung 13). Dann wurden mithilfe von Sequenzen, in denen ein Artikulationsparameter als statisch angenommen werden kann, die Koeffizienten optimiert: So wurde z.B. die Welligkeit der Lippenhöhe in Sequenzen mit relativ zum Unterkiefer konstanter Lippenhöhe wie /gagagaga/ minimiert. Für die Unterkieferöffnung wurde die vertikale Position der Empfängerspule auf dem Unterkiefer herangezogen (Gleichung 1). Die radiale Bewegung des Unterkiefers wurde somit vernachlässigt. Die Lippenhöhe wurde aus der Distanz zwischen Ober- und Unterlippe, reduziert um die Unterkieferöffnung, berechnet (Gleichung 2). Ähnlich der Bestimmung der Lippenbreite/-rundung wurden Zungenspitzen- und Zungenrückenhöhe aus der vertikalen Höhe der Empfängerspulen berechnet, jeweils reduziert um den Anteil der Unterkieferöffnung. Für die Optimierung der Koeffizienten der Gleichungen 4 und 5 wurden Sequenzen mit angenommener neutraler Lage der Zunge verwendet: /m@m@m@m@/ und /v@v@v@v@/. Einer der Artikulationsparameter ist die Rückverlagerung der Unterlippe für labiodentale Frikative. Die Sprecherin realisierte diese jedoch durch Anhebung der Oberlippe, während gleichzeitig die Unterlippe annähernd entspannt und der Unterkiefer fast geschlossen war. Gleichung 3 liefert in diesen Fällen große Werte, sonst Werte gleich oder kleiner 0. Vor der Normierung wurden negative Werte (keine labiodentale Konstriktion) auf 0 gesetzt.

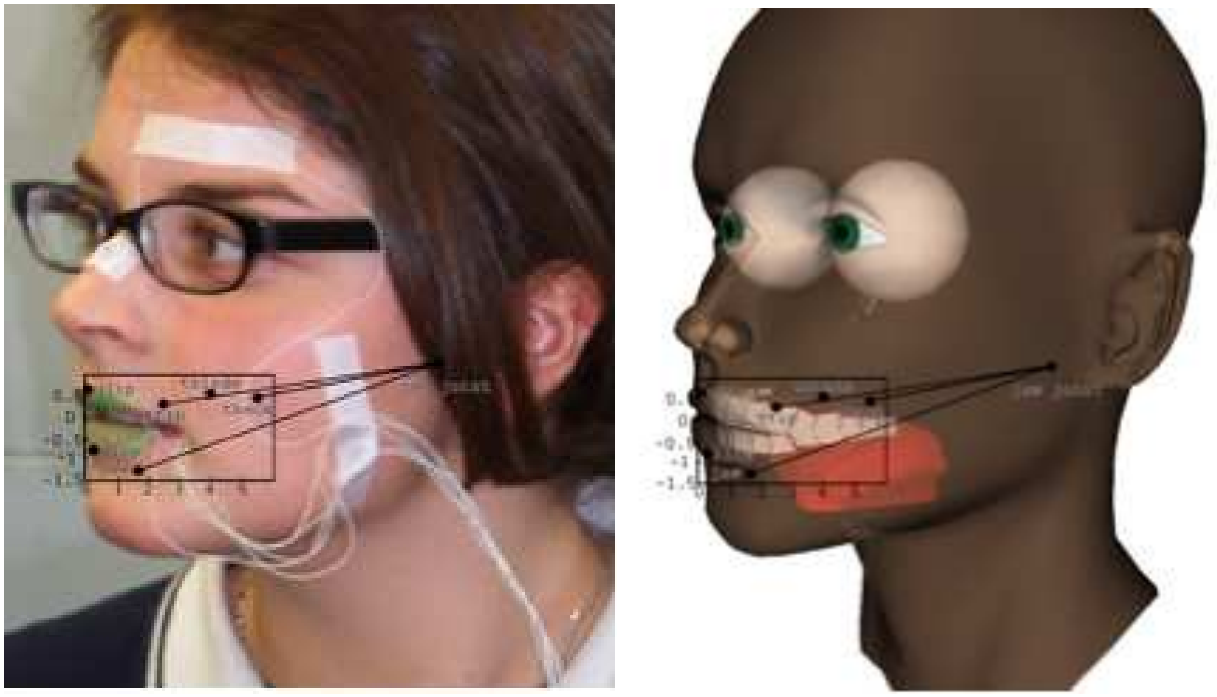

Abbildung 13: Übertragung der Messpunkte auf den virtuellen Kopf von MASSY.

Aus den EMA-Daten:

1. Unterkieferhöhe = UnterkieferY
2. Lippenhöhe = UnterlippeY − OberlippeY − 1,06 · UnterkieferY
3. Unterlippenrücklage = (UnterlippeY + 2) · OberlippeY − 1,4
4. Zungenspitzenhöhe = ZungenspitzeY − 0,925 · UnterkieferY
5. Zungenrückenhöhe = ZungenrückenY − 0,581 · UnterkieferY

Aus den Video-Daten (P bedeutet Pixel):

6. Lippenbreite = LippenbreiteP − Mittelwert(LippenbreiteP)

6 Steuerung der Sprachvisualisierung

Für die Steuerung der Sprachvisualisierung werden für jeden Laut einer darzustellenden Lautkette entsprechende Artikulatorpositionen benötigt. Um auch solche Lautketten darstellen zu können, bei denen die Artikulatorpositionen nicht gemessen wurden, werden die Artikulatorpositionen modelliert. Dieses Modell wählt aus einer Datenbank zwei Parameterbelegungen eines Lautes aus, die dem linksseitigen und rechtsseitigen lautlichen Umfeld des Lautes in der zu generierenden Äußerung am ehesten entsprechen. Der für MASSY entwickelte Pattern-Selection-Algorithmus verwendet Viseme im Kontext genau eines anderen Visems (Di-Viseme). Die Datenbank enthält eine Parameterbelegung der Zielposition jedes Konsonanten in jedem vokalischen Kontext P(C|V) und jedes Vokals in jedem konsonantischen Kontext P(V|C), z.B. [a] im Kontext [m] /mam/ oder [m] im Kontext [a] /ama/. Ausgewählt werden im Falle eines Konsonanten die Daten für diesen Konsonanten im Kontext des nächsten linken (vorhergehenden) und des nächsten rechten (nachfolgenden) Vokals, im Falle eines Vokals diejenigen Daten des Vokals im Kontext des nächsten linken sowie rechten Konsonanten (z.B. werden für das [m] in /oma/ die Daten der [m]s aus /omo/ und /ama/ ausgewählt). Zwischen diesen beiden Parameterbelegungen wird dann gewichtet mit dem Kehrwert der Distanz des Lautes zum jeweiligen Kontext-Laut (Anzahl der Laute von einem zum anderen) interpoliert. So hat z.B. das [t] in /atmo/ die Distanz dlinks=1 vom [a] und somit das Gewicht 1/1 und drechts=2 vom [o] und das Gewicht 1/2. Nach Normierung (Division durch die Summe der Gewichte) wird die Parameterbelegung P([t]|[a],[o]) des [t] in /atmo/ kombiniert zu 1/3 aus P([t]|[o]) und 2/3 aus P([t]|[a]). Analog werden die Parameterbelegungen für Vokale berechnet. Nicht immer existieren in einer Äußerung je ein nächster rechter und nächster linker Kontextlaut. In diesen Fällen muss der fehlende Kontextlaut geschätzt werden. Liegt auf einer der beiden Seiten eines Konsonanten kein Vokal, wird die Grenze der Äußerung mit dem vokalischen Visem [@] substituiert unter der Annahme, dass dieses einen möglichst geringen Einfluss ausübt. Für Vokale wurde dasselbe Vorgehen implementiert mit dem möglichst neutralen konsonantischen Visem {[X], [R]} als Substitut. Es ergaben sich nach subjektiver Beurteilung plausiblere Parameterbelegungen als bei Verwendung einer Parameterbelegung mit allen Artikulationsparameterwerten gleich 0 oder bei Verwendung nur eines Kontextlauts.

Dass die Verständlichkeit des Systems MASSY durch die Nutzung der experimentalphonetischen Messdaten verbessert wurde, konnte experimentell nachgewiesen werden (Fagel, 2004).

Die Autoren

Caroline Clemens
Zentrum Mensch-Maschine-Systeme
Technische Universität Berlin
Jebensstraße 1, Sekretariat J 2-2
D-10623 Berlin
Telefon (0049 30) 314 29634
Fax (0049 30) 314 72581
E-Mail caroline.clemens@zmms.tu-berlin.de
URL http://www.zmms.tu-berlin.de/prometei/

Caroline Clemens studierte germanistische Linguistik an der Humboldt-Universität zu Berlin und Kommunikationswissenschaft an der Technischen Universität Berlin. Ihre Magisterarbeit schrieb sie über *Phonetische Merkmale hyperartikulierter Sprechweise*. Seit 2004 arbeitet sie freiberuflich und schreibt ihre Dissertation über *Benutzerklassifikation für automatisierte Sprachdialogsysteme*. Seit 2004 ist sie Doktorandin der Siemens AG in der Abteilung User Interface Design. Sie ist Kollegiatin im Graduiertenkolleg prometei (Prospektive Gestaltung von Mensch-Technik-Interaktion) am Zentrum Mensch-Maschine-Systeme.

Dr. Sascha Fagel
Technische Universität Berlin
Ernst-Reuter-Platz 7, Sekretariat TEL10-1
10587 Berlin
Telefon (0049 30) 314 245 27
Fax (0049 30) 314 798 83
E-Mail sascha.fagel@tu-berlin.de
URL http://avspeech.info/fagel/index.html
MASSY: http://avspeech.info

Nach siebenjähriger Tätigkeit als freiberuflicher Tontechniker studierte Sascha Fagel Kommunikationswissenschaft und Informatik an der Technischen Universität Berlin. Er schloss sein Studium im Frühjahr 2001 ab und arbeitete in der Folge als Wissenschaftlicher Mitarbeiter an der Freien Universität Berlin auf den Gebieten Content Management Systems und eLearning sowie beim Fraunhofer Institut FOKUS in den Bereichen Ubiquitous Computing und Sprachtechnologie. Im Sommer 2002 wechselte er an die Technische Universität Berlin, wo er 2004 über die Entwicklung des audiovisuellen Sprachsynthesizers MASSY mit Auszeichnung promovierte.

Literatur

Fagel, S., 2004. *Audiovisuelle Sprachsynthese – Systementwicklung und -bewertung*. Logos Verlag, Berlin.

Fagel, S., 2005. Merging methods of speech visualization. *ZAS Papers in Linguistics* **40**, 19–32.

Fagel, S., Clemens, C., 2004. *An Articulation Model for Audiovisual Speech Synthesis – Determination, Adjustment, Evaluation*. In: Speech Communication (special issue on auditory-visual speech processing). Vol. 44. 141–154.

Fagel, S., Sendlmeier, W. F., 2003. *An Expandable Web-based Audiovisual Text-to-Speech Synthesis System*. In: Proceedings of the 8th EUROSPEECH European Conference on Speech Communication and Technology. Genf, 2449–2452.

Gröne, B. F., Hoch, G., Schoenle, P. W., 1992. *Die Elektromagnetische Artikulographie (EMA) – dynamische Analyse und Wiedergabe von Sprechbewegungen auf dem Computerschirm*. In: F. Roth (ed.), *Computer in der Sprachtherapie – Neue Wege*. Vol. 7 of *Sprachtherapie*. Narr, Tübingen, 113–124.

Hasegawa-Johnson, M., 1998. Electromagnetic exposure safety of the Carstens Articulograph AG100. *Journal of the Acoustical Society of America* **104**, 2529–2532.

IPA, 2006. *IPA – International Phonetic Alphabet, International Phonetic Association.* `http://www.arts.gla.ac.uk/IPA/ipa.html` (Stand: 2006).

Karlsson, I., Faulkner, A., Salvi, G., 2003. *SYNFACE – A Talking Face Telephone*. In: Proceedings of the 8th EUROSPEECH European Conference on Speech Communication and Technology. Genf.

Kröger, B., 2006. *Speech Trainer*. `http://www.ukaachen.de/content/page/3105627` Stand: 2006.

Perkell, J. S., Cohen, M., Svirsky, M., Matthies, M., Garabieta, I., Jackson, M., 1992. Electromagnetic midsagittal articulometer (EMMA) systems for transducing speech articulatory movements. *Journal of the Acoustical Society of America* **92**, 3078–3096.

SAMPA, 2006. *SAMPA, Speech Assessment Methods Phonetic Alphabet*. `http://www.phon.ucl.ac.uk/home/sampa/home.htm` Stand: 2006.

Schönle, P. W., Grabe, K., Wenig, P., Hohne, J., Schrader, J., Conrad, B., 1987. Electromagnetic articulography: Use of alternating magnetic fields for tracking movements of multiple points inside and outside the vocal tract. *Brain and Language* **31**, 26–35.

Wirth, G., 1994. *Sprachstörungen – Sprechstörungen – kindliche Hörstörungen.* Deutscher Ärzte-Verlag, Köln-Kövenich.

ZAS, 2006. *ZAS – Zentrum für Allgemeine Sprachwissenschaft, Typologie und Universalienforschung, Jägerstrasse 10/11, 10117 Berlin.* `http://www.zas.gwz-berlin.de` Stand: 2006.

Recognising Textual Entailment using semantic structure extraction and conceptual distance

Marthe Catharina Dekker[1]
Utrecht University

Abstract

We describe the design we developed as a solution to the 2005 PASCAL Recognizing Textual Entailment Challenge. In order to recognise the textual entailment relation between two texts, we performed semantic structure extraction to a wide-coverage parser output and matched these structures using the notion of conceptual distance. Preliminary results show no significant performance above baseline, but our method seems promising as there is room for improvement and optimisation in several places.

1 Introduction

Natural language is loaded with pragmatic information; humans constantly refer to the discourse, can play with language, vary their vocabulary dependent on the context, try to express themselves as an individual and to distinguish themselves from other speakers. The extraordinary capability of natural language of being able to express the "same" information in a virtually infinite number of ways is, however, a big challenge for various natural language processing applications.

In common and popular tasks like Information Retrieval and Question Answering, many different language interpretation strategies are being used; e.g. deep semantic analysis, use of non-linguistic statistical methods, parse tree matching or keyword matching. The methods used are so diverse and tuned to the specific task that it is difficult to compare them to one another and evaluate their ability to "understand" language. There was a need for a new generic and well-defined task that could be used to evaluate interpretation systems.

In 2004, the PASCAL-network of excellence[2] organised a first challenge in Recognizing Textual Entailment (hereafter referred to as *RTE1*) as a new task and evaluation framework for natural language understanding. PASCAL is a world-wide network of researchers in natural language processing and statistical modelling, investigating and developing methods to improve human-machine communication. They organise application- as well as theoretical challenges in order to raise interest in and stimulate literature on these topics. The *RTE1* task was to automatically recognise if, for two short texts, the meaning of the first was entailing the meaning of the second, and to give a confidence value to this judgement. Example (1) is an example of a pair annotated as TRUE.

(1) a. *T*: An Iraqi official reported today, Saturday, that 68 Iraqi civilians were killed today as a result of the American and British bombing on Iraq and that their funerals were held today in Baghdad.

b. *H*: Reports from an Iraqi official today, Saturday, said that the 68 civilians killed as a result of the Anglo-American bombing of Iraq, were buried today in Baghdad.

The data set consisted of hundreds of pairs of English texts from the news domain, manually annotated with a boolean telling if entailment holds or not and with a marker indicating of which practical text processing task (Information Retrieval, Question Answering, etc.) they were a typical success or failure case.

[1] Questions are welcome at: `m.c.dekker1@students.uu.nl`

[2] "PASCAL" stands for "Pattern Analysis, Statistical Modelling and Computational Learning". See `http://www.pascal-network.org/` for more information about the network and `http://www.pascal-network.org/Challenges/RTE/` for more information about this challenge.

The outcome of the challenge (see Dagan et al. 2005) shows a variety of methods and frameworks being used. An important conclusion that can be drawn from the outcome is that *RTE* is a very difficult task and is still very much in its infancy. Accuracy, precision and recall most of the time scarcely outperform the baseline of simple guessing. It is striking that methods as different as naive lexical based systems and sophisticated logical inference systems do not show very different results. However, some methods seem more promising than others. Bayer et al. (2005) cites Hubert Dreyfus' well-known quote on that "one should be aware of the temptation to climb a tree in order to get to the moon" to point out that short-term solutions can be initially superior, but are frequently dead ends when the goal is human-like performance. It is best to invest in solutions combining the best of both.

In this paper, we will present our own approach to this task, which uses a deep syntactic analysis of the texts using a wide coverage CCG parser, extraction of semantic structure and matching of these structures with the use of the notion of conceptual equivalence.

2 The Task

Participants to *RTE1* had to develop their system using a development corpus of about 570 pairs of short English texts, typically one or two longer sentences as "Text" (*T*, the first text) and one shorter sentence as "Hypothesis" (*H*, the second), xml-coded and manually annotated for existence of the entailment relation (TRUE/FALSE) and the kind of task the pair would be a typical example of. TRUE and FALSE examples were distributed evenly over the data. A large evaluation set of 800 pairs, annotated for task type and TRUE/FALSE was present for a final evaluation. As to the use of generic software, not tuned to the development set, there were no restrictions.

Textual Entailment is the relation between two texts for which holds that the truth of the former entails the truth of the latter. Recognizing textual entailment is the task of deciding if two unseen texts are in the entailment relation and giving a confidence score to this judgement.

2.1 The Data

The dataset of Text-Hypothesis pairs consisted of seven subsets, corresponding to seven different applications; *IR* (Information Retrieval), *CD* (Comparable Documents), *RC* (Reading Comprehension), *QA* (Question Answering), *IE* (Information Extraction), *MT* (Machine Translation) and *PP* (Paraphrase Acquisition).

The data originated from various corpora of English newspaper text (see the *RTE1* website for more details) but were manually adapted in order to create useful *RTE* pairs and were grammatically corrected if necessary. It did include elliptical structures typical for newspaper headlines like: "No Weapons of Mass Destruction Found in Iraq Yet".

Within the pairs annotated for a specific task, there were examples of various inference types (e.g. syntactic, semantic and pragmatic) and difficulty levels. An example of type *MT* can be found in Example (1). It is annotated as TRUE. *T* and *H* are an automatic translation and a gold-standard human translation of the same original sentence. The main characteristic of *MT* pairs is their parallelism on the word order level, whereas structurally they are rather different. The example illustrates some of the problems we are facing when trying to recognise entailment.

3 Design

Figure 1 shows the design of our system and Figure 2 is a pseudo-code overview of our implementation.

We divided our development set in two equal sized parts that both contained 50% TRUE and 50% FALSE pairs and trained our algorithm on the first half (hereafter called *training set*) and evaluated it on

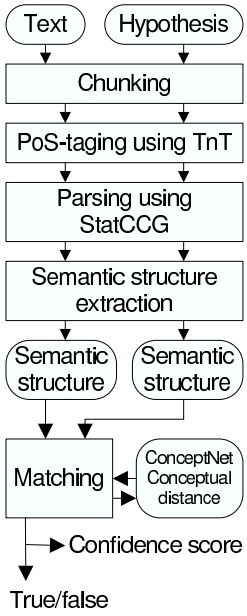

Figure 1: System architecture.

the second half (the *test set*). We performed a detailed syntactic analysis of each and every sentence in the development set using a PoS-tagger and a wide-coverage parser. After this, for each pair, we extracted the semantic structure of both *T* and *H*, interpreted these as a set of assertions and calculated for each assertion in *H* the conceptual similarity between this assertion and the most (conceptually) similar assertion in *T*. The total entailment score for each pair was similar to the sum of all similarity scores of the pair, divided by the number of assertions in *H*.

3.1 Training

We used the training set to optimise the threshold that should divide the TRUE ones (entailment score *above* the threshold) from the FALSE ones (entailment score *below* the threshold). This threshold was optimised by calculating the accuracy of the algorithm for a series of threshold values (in fact, the set containing the entailment values of all pairs) and taking the threshold that gave the most accurate result. Accuracy was calculated by summing up, for all right judged pairs (true positives and true negatives), the confidence score and to substract from that the confidence score for every wrongly judged pair.

The confidence score (the relative distance to the threshold, expressing the confidence of the algorithm in its judgement) was calculated directly out of the entailment value by simply taking the distance between entailment value a and threshold relative to the difference between the maximum entailment value b (for pairs judged right) or minimal entailment value c (for pairs judged wrongly) and the threshold (also see Figure 3 for an imaginary situation):

$$\textit{confidence score} = \begin{cases} \frac{|(threshold-a)|}{|(threshold-b|} & if\ a > threshold \\ \\ \frac{|(threshold-a)|}{|(threshold-c|} & if\ a < threshold \end{cases}$$

1. ***Pre-processing***
2. for first half of .xml development set (= training set)
3. Read xml and extract T-H pairs
4. for each pair
5. for each sentence
6. Apply chunking
7. Run TnT and transform output into StatCCG-input
8. Run StatCCG
9. ***Information extraction***
10. for every pair of a list of tasks we want to train on/test on, for every sentence
11. Extract "meaningful" verbs and their arguments (AGENT, PATIENT, THEME) and negation information
12. ***Semantics matching***
13. for every pair, for every *H*
14. for every assertion in *H*, compute the conceptual distance to the closest assertion in *T* using ConceptNet conceptual distance
15. ***Threshold optimisation***
16. Search for threshold that gives best accuracy
17. ***Evaluation***
18. for second half of .xml development set (= test set)
19. Perform $[1-14]$
20. for every pair
21. if score > threshold *t* return **true**
22. else return **false**
23. if **true** return confidence score
24. Calculate accuracy, precision and recall

Figure 2: Pseudo-code for our method of recognising entailment.

3.2 Testing

After having optimised the threshold on the training set, the algorithm was run over the semantically analysed test set and its accuracy was calculated as above. In general, the more confident the algorithm is about its right judgements, the more accurate it is.

We now describe the semantic analysis of our data in more detail.

3.3 Tagging

We run our own chunking algorithm over every sentence and used Thorsten Brants' TnT tagger (Brants, 2000) to PoS-tag each word. After that we run Julia Hockenmaier's wide coverage StatCCG parser (Hockenmaier, 2003) over TnT's output. TnT (Trigrams'n'Tags) is a fast tagger, trainable on corpora in almost any language with almost every tagset. It has been pretrained on the German NEGRA and the English Susanne corpora and the (Wall Street Journal (*WSJ*) part of the) Penn Treebank. I used the *WSJ* part because of its performance on unseen text (accuracy of about 97,7%) and the fact that its tagset is the same as the one StatCCG has been trained on.

TnT's interpretation of text (1a) can be found in Figure 4.

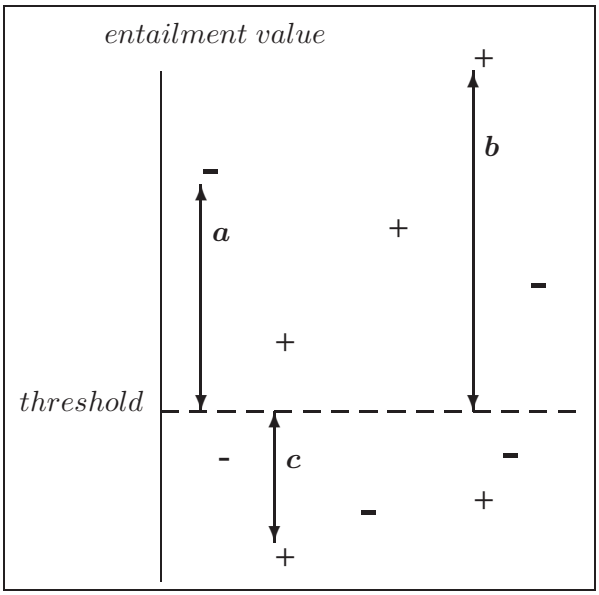

Figure 3: $+$ is a pair judged as TRUE, $-$ a pair judged as FALSE, a = highest value, b = difference between entailment value and threshold, c = lowest value. The horizontal dimension has no meaning here.

```
An_DT Iraqi_JJ official_NN reported_VBD today_NN ,_, Saturday_NNP ,_,
that_IN 68_CD Iraqi_JJ civilians_NNS were_VBD killed_VBN as_IN a_DT
result_NN of_IN the_DT American_JJ and_CC British_JJ bombing_NN on_IN
Iraq_NNP and_CC that_IN their_PRP funerals_NNS were_VBD held_VBN
today_NN in_IN Baghdad_NNP
```

Figure 4: TnT's PoS-tagging of Example (1a).

3.4 Parsing

StatCCG is a wide-coverage parser based on Combinatory Categorial Grammar (CCG, Steedman 2000) and had been trained on a translation of the Penn Treebank tree structures into CCG derivations (see Hockenmaier 2003). It is based on a statistical model and returns the most plausible of possible parses. CCG is a lexicon-based surface-structure grammar formalism based on the Categorial Calculus and Combinatory Logic. The syntactic behaviour of a word is defined in the lexicon by its lexical category, which is either an atomic category like N or a complex functor category build out of atomic categories and backward and foreward slashes or other combinators, as in NP/N. Rules provide for the combination of words to phrases, including interesting phenomena such as long-range dependencies being brought about by raising, coordination and extraction.

We are interested in using derivations as a source of semantic information. A reason for choosing a CCG parser is that CCG's semantics are well defined and the interpretation is very much parallel to the syntactic derivation. Another reason is the handling of long-range dependencies. The reason we chose to use StatCCG is its excellent performance and the fact that is was trained on texts of a similar kind as the ones we use. See Hockenmaier's thesis (Hockenmaier, 2003) for an evaluation of StatCCG.

Figure 6 shows a fragment of StatCCG's parse of text (1a), "killed as a result of the American and British bombing". "killed" has category `S[pss]\NP`, indicating a passive sentence (`S[pss]`) lacking a noun phrase `NP` to the left side. If there is a slash, the part before it is always the head of the phrase this word or phrase 'would like to be', the right side is the category of the satellite it has to have to the right (`/`) or to the left (`\`) in order to build this phrase.

```
<s> 34
2     0     NP[nb]/N                1     official_NN    An_DT
2     1     N/N                     1     official_NN    Iraqi_JJ
2     3     (S[dcl]\NP)/S[em]       1     official_NN    reported_VBD
3     4     (S\NP)\(S\NP)           2     reported_VBD   today_NN
3     6     (S\NP)\(S\NP)           2     reported_VBD   Saturday_NNP
8     3     (S[dcl]\NP)/S[em]       2     that_IN        reported_VBD
11    9     N/N                     1     civilians_NNS  68_CD
11    10    N/N                     1     civilians_NNS  Iraqi_JJ
11    12    (S[dcl]\NP)/(S[pss]\NP) 1     civilians_NNS  were_VBD
11    13    S[pss]\NP               1     civilians_NNS  killed_VBN
12    8     S[em]/S[dcl]            1     were_VBD       that_IN
13    12    (S[dcl]\NP)/(S[pss]\NP) 2     killed_VBN     were_VBD
13    14    ((S\NP)\(S\NP))/NP      2     killed_VBN     as_IN
13    23    ((S\NP)\(S\NP))/NP      2     killed_VBN     on_IN
16    14    ((S\NP)\(S\NP))/NP      3     result_NN      as_IN
16    15    NP[nb]/N                1     result_NN      a_DT
16    17    (NP\NP)/NP              1     result_NN      of_IN
22    17    (NP\NP)/NP              2     bombing_NN     of_IN
22    18    NP[nb]/N                1     bombing_NN     the_DT
22    19    N/N                     1     bombing_NN     American_JJ
22    21    N/N                     1     bombing_NN     British_JJ
24    23    ((S\NP)\(S\NP))/NP      3     Iraq_NNP       on_IN
26    3     (S[dcl]\NP)/S[em]       2     that_IN        reported_VBD
28    27    NP[nb]/N                1     funerals_NNS   their_PRP
28    29    (S[dcl]\NP)/(S[pss]\NP) 1     funerals_NNS   were_VBD
28    30    S[pss]\NP               1     funerals_NNS   held_VBN
29    26    S[em]/S[dcl]            1     were_VBD       that_IN
30    29    (S[dcl]\NP)/(S[pss]\NP) 2     held_VBN       were_VBD
30    31    (S\NP)\(S\NP)           2     held_VBN       today_NN
30    32    ((S\NP)\(S\NP))/NP      2     held_VBN       in_IN
33    32    ((S\NP)\(S\NP))/NP      3     Baghdad_NNP    in_IN
<\s>
```

Figure 5: StatCCG's listing of all head-satellite pairs in Example (1a).

So the satellite of "killed" must be to the left of it and the NP in its category does not refer to one of the NP's in the parse of Figure 6. Other heads in this fragment are "as", "a", "a result", "of", "the", "American" and "American and British". As to the phrase "and British", it is impossible to say which is the head and which is the satellite, "and" or "British", because "British" has its satellite to the right ("bombing") and "and" does not contain a slash.

A category like $((S\backslash NP_1)\backslash(S\backslash NP_2))/NP_3$ ("as") contains three 'slots' that, in this case, can be matched with NP's. We indexed them here for clarity. In Table 5, which is part of the StatCCG output for Example (1a), every row represents the relation between a head and its argument, which is represented by the leaf with the deepest embedding. The fifth and sixth column from the left represent argument and head, the first and second column their place in the sentence, the third column the category of the head and the fourth column the index of the argument.

When extracting meaningful assertions from a structure like this, we are interested in only part of the word-word relations. Grammatical time plays no role in this challenge, therefore modals can be ignored, and also the distinction of present and past verbs.

```
{S[pss]\NP {S[pss]\NP killed_VBN}
           {(S\NP)\(S\NP) {((S\NP)\(S\NP))/NP as_IN}
                          {NP {NP {NP[nb]/N a_DT}
                                  {N result_NN}}
                              {NP\NP {(NP\NP)/NP of_IN}
                                     {NP {NP[nb]/N the_DT}
                                         {N {N/N {N/N American_JJ}
                                                 {N/N[conj] {conj and_CC}
                                                            {N/N British_JJ}}}
                                            {N bombing_NN}}}}}}}
```

Figure 6: StatCCG's parse of "killed as a result of the American and British bombing" in Example (1a).

3.5 Extracting Semantic Structures

Gildea & Hockenmaier (2003) proposed a system to automatically identify semantic roles by mapping CCG parse trees to Penn Treebank trees labeled with PropBank semantic roles. Head-satellite relations as in Figure 5 could in this way be mapped to semantic roles.

Semantic structure extraction is, in our design, done in two steps. For each pair, for each sentence in *T* and *H*, we extract all verbs (except modals), the noun phrases that are their primary arguments and their thematic roles, and negation information. All verbs and their arguments are stored in an attribute-value-matrix like in Figure 7 (representing the meaning of Example (1a)). The numbers 3, 12 etcetera represent the place of the predicate in the original sentence and are used to identify the predicate in case of multiple instances of the same verb in a sentence. In this example, each of the five AVM's represent an assertion and the conjunction of the five assertions represents the meaning of sentence (1a).

$$
\begin{bmatrix}
3 & \begin{bmatrix} \text{PREDICATE} & \textit{reported\_VBD} \\ \text{AGENT} & \begin{bmatrix} \text{CORE} & \textit{official} \\ \text{NP} & \textit{Iraqi official} \end{bmatrix} \end{bmatrix} \\
12 & \begin{bmatrix} \text{PREDICATE} & \textit{were\_VBD} \\ \text{THEME} & \begin{bmatrix} \text{CORE} & \textit{civilians} \\ \text{NP} & \textit{68 Iraqi civilians} \end{bmatrix} \\ \text{VERBCLUSTER} & \boxed{1} \end{bmatrix} \\
13 & \begin{bmatrix} \text{PREDICATE} & \boxed{1}\,\textit{killed\_VBN} \\ \text{THEME} & \begin{bmatrix} \text{CORE} & \textit{civilians} \\ \text{NP} & \textit{68 Iraqi civilians} \end{bmatrix} \end{bmatrix} \\
29 & \begin{bmatrix} \text{PREDICATE} & \textit{were\_VBD} \\ \text{THEME} & \begin{bmatrix} \text{CORE} & \textit{funerals} \\ \text{NP} & \textit{funerals} \end{bmatrix} \\ \text{VERBCLUSTER} & \boxed{2} \end{bmatrix} \\
30 & \begin{bmatrix} \text{PREDICATE} & \boxed{2}\,\textit{held\_VBN} \\ \text{THEME} & \begin{bmatrix} \text{CORE} & \textit{funerals} \\ \text{NP} & \textit{funerals} \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

Figure 7: Extracted semantic structure after step 1.

Extraction of verbs and their arguments was performed in the following way:

1. For each *T* or *H*, take its head-satellite structure like in Figure 5 and its parse tree. For every non-modal head:
2. Store all verbal heads with a singular of plural noun phrase, personal pronoun or comparative adverb as satellites. The largest noun phrase in which the noun is present and the verb is not, will be the argument of the head, its thematic role will be AGENT if its index is 1, PATIENT if its index is 2 and THEME if the verb is passive (if the category of the head includes the *voice* feature $[pss]$). The satellite itself is saved as the 'core' of the argument.
3. If the satellite is a verb (and no modal) itself, the verbal head and satellite form a verb cluster. This information is added as an index to the semantic structure information of the head verb.
4. If the head is a negating element and the satellite a verb, the negation of the verb is added to the semantic structure of this verb.

At this stage, we go through each attribute-value-matrix for a second time to delete auxiliaries:

1. For every assertion:
2. Remove the assertion if it contains a verb cluster and a negation. Add the negation information to the semantic structure of the predicate.
3. If the VERBCLUSTER-attribute points to an auxiliary and the predicate is a head verb, remove the assertion.
4. If an assertion contains a PATIENT and no AGENT: change thematic role of the argument to AGENT. This is probably due to a parsing error.
5. Remove all remaining assertions with verb clusters

For Example (1a), we come to the semantic structure in Figure 8.

$$\left[\begin{array}{ll} 3 & \left[\begin{array}{ll} \text{PREDICATE} & \textit{reported} \\ \text{AGENT} & \left[\begin{array}{ll} \text{CORE} & \textit{official} \\ \text{NP} & \textit{Iraqi official} \end{array}\right] \end{array}\right] \\ 13 & \left[\begin{array}{ll} \text{PREDICATE} & \textit{killed} \\ \text{THEME} & \left[\begin{array}{ll} \text{CORE} & \textit{civilians} \\ \text{NP} & \textit{68 Iraqi civilians} \end{array}\right] \end{array}\right] \\ 30 & \left[\begin{array}{ll} \text{PREDICATE} & \textit{held} \\ \text{THEME} & \left[\begin{array}{ll} \text{CORE} & \textit{funerals} \\ \text{NP} & \textit{funerals} \end{array}\right] \end{array}\right] \end{array}\right]$$

Figure 8: Extracted semantic structure after step 2.

3.6 Conceptual Distance

For the matching of the extracted semantic structures, we used a common sense knowledgebase.

We wanted to have access to linguistic knowledge ("he's held captive" means that "he's detained"), encyclopedic knowledge (an "Academy Award" is also called an "Oscar") as well as common sense knowledge. Common sense knowledge is knowledge that every child knows although it may not be in any dictionary or lexicon. An example could be that "if you sleep you don't stand upright", or "a fork is not used for eating soup". An often-used source of linguistic knowledge in language technology, as in *RTE1*, is WordNet[3], a semantic ontology containing lexical information about words like synonymy. An example of a similar project is FrameNet[4], in which concepts or words are grouped together in situation types having their own specific semantic behaviour, the so-called semantic frames. An interesting phenomenon however, is the development of more associative conceptual networks like the common sense database and reasoning system Cyc[5] and also Mindpixel[6] and ThoughtTreasure[7]. Many of these projects have integrated WordNet's ontology.

ConceptNet, the system we used in our design, was developed at MIT Media Lab, has been inspired by both WordNet and Cyc and contains 1.6 million assertions in natural language notation. The ConceptNet database was automatically generated out of the Open Mind Common Sense Repository[8], which is a collection of nearly 700,000 English sentences of common sense facts brought together by thousands of volunteers. Similar knowledgebases have often been hand-crafted by experts. ConceptNet contains (often multi-word) concepts which are all connected by 19 binary relation types. The semantics of these relations is informal, syntactics and semantics of their arguments are not restrained in any way, making it possible to calculate the distance along one or more path types between concepts of different kinds.

A possible problem with ConceptNet is its anarchistic nature, making it difficult to evaluate. See Liu & Singh (to appear) for more information about the quality of ConceptNet.

For the calculation of the conceptual distance between two assertions, we used ConceptNet's `get_context` method. `get_context` walks the conceptual network like a spider: "Technically speaking, the contextual neighbourhood around a node is found by performing spreading activation radiating outward from that source node. The relatedness of any particular node is not simply a function of its link distance from the source, but also considers the number and strengths of all paths which connect the two nodes." (Liu & Singh, to appear). `get_context` contains a vector of relation types to weights varying from 0.0 to 1.0, which can be biased towards, for example, spatial or causal context. When calculating the conceptual distance, we decided to emphasise the so-called "K-Line" relations "ConceptuallyRelatedTo", "ThematicKLine" and "SuperThematicKLine" and the Event relations "FirstSubeventOf", "SubEventOf", "LastSubeventOf" and "PrerequisiteEventOf", the causal relations "EffectOf", "DesirousEffectOf" and the affective relation "MotivationOf" for predicates and the relations "IsA", "DefinedAs", "PartOf", "MadeOf", the K-Line relations and so-called Thing relations for noun phrases. We tried out different weight distributions.

When calculating the conceptual distance between two assertions, we first calculated the distance between word groups having the same semantic function (and assigned this distance a heavy weight). When this distance was low, we calculated the distance between word groups with possibly different roles. There was a penalty for negation disagreement between two assertions.

Figure 9 shows a possible context for the verb "shoot". Conceptual distance between "shoot" and "kill person" is approximately 0.12.

[3] `http://wordnet.princeton.edu/`
[4] `http://framenet.icsi.berkeley.edu/`
[5] `http://www.cyc.com`
[6] `http://www.mindpixel.com`
[7] `http://www.signiform.com/tt/htm/tt.htm`
[8] `http://openmind.media.mit.edu/`

```
[[shoot, 1.0], [win, 0.26], [backdrop, 0.16], [run after ball, 0.16],
[hang out at bar, 0.16], [commit suicide, 0.20], [fight war, 0.19], [gun,
0.15], [play basketball, 0.19], [attend school, 0.17], [firearm, 0.20],
[pistol, 0.20], [cannon, 0.20], [cannonball, 0.20], [fight enemy, 0.20],
[rifle, 0.20], [spit, 0.15], [shoot animal, 0.15], [dangerous weapon, 0.14],
[revolver, 0.13], [bullet, 0.09], [eat pizza, 0.11], [shoot gun, 0.14],
[shoot person, 0.16], [shoot bullet, 0.16], [handgun, 0.17], [projectile,
0.16], [rubber band, 0.17], [advance into battle, 0.11], [shoot water, 0.12],
[kill person, 0.12], [skate, 0.15], [shooting, 0.15], [shoot wild animal,
0.06], [hunt, 0.07], [victim, 0.09], [shoot person with gun, 0.09] (\ldots)]
```

Figure 9: Example of a context list; fragment of (a possible) context of "shoot". We rounded the values to three decimals here for clarity.

4 Conclusions and Discussion

Our design was test-wise evaluated as described above. Preliminary results on an unseen test set show a performance of around baseline (= 50%). Although, at the moment, we scarcely outperform simple guessing, we hope to improve the system's performance by implementing the following functionality:

1. Dealing with tagging and parsing errors, which is not done at the moment. Also adding rules that recognise and rewrite specific syntactic constructions that the parser may not be able to deal with (and parse again).

2. Extracting more information; apart from only primary arguments and negation also TIME, PLACE and maybe also CAUSE.

3. Performing (naive) anaphora resolution.

4. (Automatically) optimising the weighting inside of the conceptual distance calculation.

5. Optimising the algorithm for each task type separately, or, more appealing if dealing with real-world data, recognise entailment types using syntactic of semantic features and adapt the algorithm or the weighting to them separately.

Some participants to *RTE1* have complained about the development data's quality and representativity. Real-world data would be more appropriate. Also, the quality of taggers, parsers and the like should not play a role in the competition as it is now. For *RTE2*, which is held in 2006, some of these wishes have been met.

References

Bayer, S., Burger, J., et al., 2005. *Mitre's submissions*. In: Proceedings of the PASCAL Challenges Workshop.
URL `http://www.cs.biu.ac.il/~glikmao/rte05/bayer_et_al.pdf`

Brants, T., April 29 – May 3 2000. *TnT – a statistical part-of-speech tagger*. In: Proceedings of the 6th Applied NLP Conference, ANLP-2000. Seattle, WA.
URL `http://www.coli.uni-saarland.de/~thorsten/tnt/`

Dagan, I., Glickman, O., Magnini, B., 2005. *The PASCAL recognizing textual entailment challenge*. In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment.
URL `http://www.cs.biu.ac.il/~glikmao/rte05/dagan_et_al.pdf`

Gildea, D., Hockenmaier, J., 2003. *Identifying semantic roles using Combinatory Categorial Grammar*. In: 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP). Sapporo, Japan, 57–64.

Hockenmaier, J., 2003. *Data and models for statistical parsing with Combinatory Categorial Grammar*. Ph.D. dissertation, School of Informatics, University of Edinburgh.
URL `http://www.ircs.upenn.edu/~juliahr/Parser/`

Liu, H., Singh, P., to appear. Conceptnet: A practical commonsense reasoning toolkit. *BT Technology Journal* **22**, (forthcoming issue) 2004.

Steedman, M., 2000. *The syntactic process*. The MIT Press, Cambridge, Massachusetts.

Ein Named-Entity-Recognition-System fürs Deutsche

Julia Ritz
Institut für Maschinelle Sprachverarbeitung (IMS)
Universität Stuttgart
Julia.Ritz@ims.uni-stuttgart.de

Zusammenfassung

Die Named-Entity-Recognition (NER) beschäftigt sich mit der Erkennung und Klassifikation von Eigennamen, sowie temporalen Angaben und Maßangaben. Im Rahmen einer Studienarbeit wurde ein regelbasiertes NER-System entwickelt. Dieses wird im Folgenden vorgestellt.

1 Aufgabe

In den Jahren 1987 bis 1998 wurde im Rahmen einer Reihe von Konferenzen, den Message-Understanding-Conferences, eine Aufgabendefinition zur Named-Entity-Recognition (Chinchor, 1998) ausgearbeitet. Diese Definition liegt meiner Arbeit zugrunde. Chinchor (1998) unterscheidet zwischen folgenden Typen von Named-Entities (NEs):

- ENAMEX
- NUMEX
- TIMEX

Die Kategorie ENAMEX enthält Namen von Personen, Organisationen und Orten; NUMEX enthält prozentuale und monetäre Angaben sowie Maßangaben; TIMEX Datums- und Zeitangaben (Beispiele: siehe Tabelle 1). Die Aufgabe eines NER-Systems ist es, Entitäten der genannten Kategorien in beliebigem Text aufzufinden und ihre Grenzen sowie ihre Kategorie in einem XML-Format zu annotieren.

Zu den Schwierigkeiten, die sich bei dieser Aufgabe ergeben, gehören u.a. Homonymie (siehe Bsp. (1)) und Vagheit (siehe Bsp. (2)). Weitere Probleme sind die Findung der Phrasengrenzen im Mittelfeld (siehe Bsp. (3)), sowie eine relativ vielfältige Flexion des Deutschen im Vergleich zum Englischen (Bsp. (4)).

(1) Mark/NE/NN
Essen/NE/NN

(2) *Ankara* spricht von einem historischen Schritt.

(3) Unklar bleibt, ob die [Firma] [Günter Wagner] weiter beschäftigt.
Unklar bleibt, ob die [Firma Günter] [Wagner] weiter beschäftigt.
Unklar bleibt, ob die [Firma Günter Wagner] weiter beschäftigt.

(4) das Rote Kreuz vs. *the Red Cross*
des Roten Kreuzes
dem Roten Kreuz

ENAMEX	
person:	George W. Bush; Dr. Kohl; Franz Josef Strauß
organization:	Robert Bosch GmbH; Universität Stuttgart; Bundesministerium für Arbeit
location:	New York; Österreich; Neckar; Rocky Mountains
NUMEX	
percentage:	30%; 2,9 Prozent
measure:	300 Meter; 40 000 Tonnen
monetary expression:	1,20 Euro; 20 Millionen Australische Dollar
TIMEX	
date:	14.09.04; 14. September; Ende 2004
time:	12.30 Uhr; 12 Uhr 30

Tabelle 1: Beispiele.

2 Motivation

Zum einen ist Named-Entity-Recognition ein Teilschritt der Informationsextraktion. Diese hat eine kompakte, übersichtliche Darstellung von Information in einem ‚maschinenlesbaren' Format (d.h. in Tabellenform) zum Ziel.

Zum anderen kann Named-Entity-Recognition einen wichtigen Beitrag zu verschiedenen computerlinguistischen Disziplinen leisten, u.a. zum Information Retrieval (NEs als Indexterme), zu Tagging (Desambiguierung von Konflikten zwischen NN und NE), Chunking/Parsing (als Hilfe bei der Bestimmung von Phrasengrenzen), sowie in der komputationellen Lexikographie (z.B. zur Extraktion von Selektionsrestriktionen).

Die meisten bestehenden NER-Systeme sind speziell fürs Englische entwickelt. Darüberhinaus existieren einige multilinguale Systeme (z.B. von SailLabs), meist auf statistischen Methoden basierend.

3 Ansatz und Implementierung

Basierend auf bestehenden Ressourcen sollte ein regelbasiertes NER-System entwickelt werden. Abb. 1 zeigt den Systemaufbau und die verwendeten Ressourcen.

Die Tokenisierung, das Nachschlagen im Lexikon und die Zwischenspeicherung in einer Chart wird von PEP übernommen (unter Verwendung der Deutschen Morphologie (DMOR, Schiller 1995)). PEP steht für *pattern matching easy-first planning* und bezeichnet ein System, das neben dem hier verwendeten Tokenizer und Tagger auch noch einen Parser beinhaltet. Es arbeitet mit Suchmustern, die zu komplexen Bedingungen zusammengefügt werden können. Dargestellt werden diese komplexen Bedingungen als Netzwerke, wobei die einzelnen Suchmuster die Übergänge zwischen den Zuständen bilden. Die für die NER definierten Regeln werden durch den PEP Netzwerkinterpreter abgearbeitet. Den Anstoß für die Abarbeitung eines Netzwerks gibt ein sogenannter *Trigger*. Dem easy-first Ansatz von PEP folgend, werden zunächst NEs behandelt, die in einem Schritt erkannt und klassifiziert werden können. Trigger dieser NEs fungieren sowohl als Indikator, als auch als Akzeptor für eine bestimmte Kategorie. Beispiele für derartige Trigger sind *AG* und *GmbH* für die Kategorie `ENAMEX organization` (siehe Beispiele (5) und (6)).

(5) die Thyssen *AG*

(6) der japanische Weltkonzern Nikon Precision *GmbH*

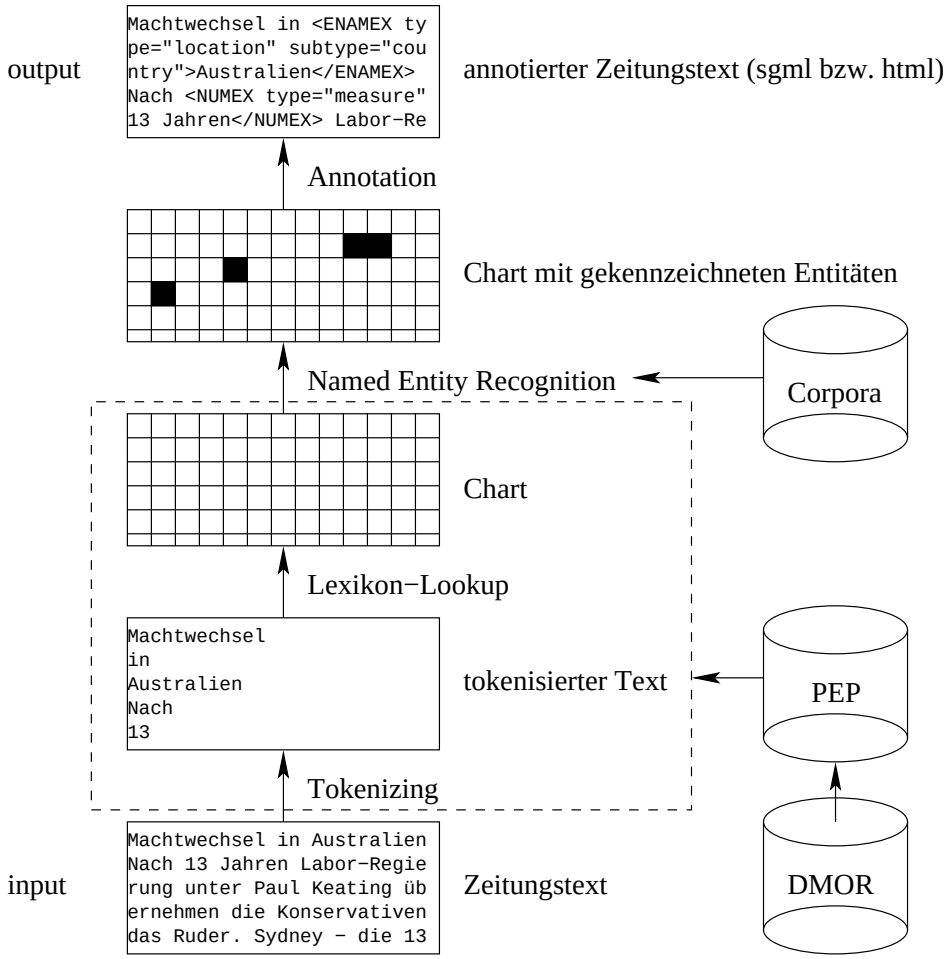

Abbildung 1: Systemaufbau.

Darüberhinaus existieren Trigger, die keinen Hinweis auf eine Klassifikation zulassen. Dazu gehört v.a. die Wortart. Wörter, die beim PoS-Tagging schon als (Teil einer) NE erkannt wurden (oder eine NE-Lesart haben) oder Wörter, deren Wortart vom Tagger nicht erkannt wurde (*u*, d.h. *unbekannt*) – siehe Beispiele (7) bis (10) – werden als NEs akzeptiert, falls sie (und ggf. Nachbartoken(s)) klassifiziert werden können.

(7) Donau$_{NE}$

(8) Teufel$_{NN|NE}$

(9) Bam$_{u}$

(10) Infineon$_{u}$

Die Klassifikation kann entweder duch Akzeptoren im lokalen Kontext geschehen (siehe Beispiel (11)) oder durch Zurückgreifen auf ein Korpus: eine Suche nach den häufigsten Wörtern im linken Kontext der zu klassifizierenden Entität wird angestoßen. Enthalten diese Wörter Hinweise auf eine bestimmte Klasse (z.B. das Suffix *-in* als Hinweis auf `ENAMEX person`), wird die Entität dieser Klasse zugeordnet.

(11) der baden-württembergische *Regierungschef* Teufel

(12) Aung San Suu Kyi

Friedensnobelpreisträgerin	*11*
Oppositionsführerin	*4*
Politikerin	*2*

Eine konsistente Annotation wird dadurch erreicht, dass sämtliche Vorkommen einer einmal aufgefundenen Entität innerhalb des Eingabetexts identifiziert und derselben Klasse zugeordnet werden. Damit wird das ‚*one sense per discourse*'-Prinzip (Yarowsky, 1995) umgesetzt.

(13) <article> Der Aussiedlerbeauftragte der Bundesregierung, Jochen Welt/~~NN~~/NE, ...
Welt/~~NN~~/NE wies darauf hin, ...
Nach Welts/~~NN~~/NE Meinung ...
</article>

Bei komplexen NEs spielt natürlich die Erkennungsreihenfolge eine Rolle. In den Beispielen (14) bis (16) wird deutlich, warum Orte und Personen vor Organisationen erkannt werden müssen und numerische Ausdrücke vor Datumsangaben.

(14) die Universität Stuttgart

(15) die Robert Bosch AG

(16) So wurden allein im *Januar 2000* Liter Öl verbraucht.

4 Ergebnisse

Zur Evaluation wurden 28 Zeitungsartikel unterschiedlicher Sparten (Politik, Sport, Reise, etc.) mit dem NER-System analysiert. Zuvor wurde manuell, den Guidelines (Chinchor, 1998) entsprechend, ein *gold standard* zum Vergleich annotiert. Die Ergebnisse sind in den Tabellen 2 und 3 dargestellt[1].

	cor	mis	spu	Prec	Rec
ENAMEX	755	253	113	87,0	74,9
TIMEX	107	42	2	98,2	71,8
NUMEX	129	18	15	89,6	87,8

Tabelle 2: Ergebnisse.

	cor	mis	spu	Prec	Rec
location	382	65	26	93,6	85,5
person	196	109	12	94,2	64,3
organization	177	79	75	70,2	69,1

Tabelle 3: Ergebnisse für ENAMEX im Detail.

[1] Abkürzungen in den Tabellen 2 und 3:
cor – correct
mis – missing (fehlt)
spu – spurious (überflüssig)
Prec – Precision ($=\frac{cor*100\%}{cor+spu}$)
Rec – Recall ($=\frac{cor*100\%}{cor+mis}$)

Ein Beispiel für die Systemausgabe findet sich in Abb. 2.

```
<article>
<s>
<ENAMEX type=person trigger=vorname cat=np>
Helmut
Herzfeld
</ENAMEX>
ist
in
<ENAMEX type=location subtype=city trigger=ne cat=np>
Berlin
</ENAMEX>
geboren
,
vor
genau
<NUMEX type=num subtype=measure trigger=me cat=np>
66
Jahren
</NUMEX>
</s>
(. . .)
</article>
```

Abbildung 2: Systemausgabe (Beispiel).

5 Ausblick

Eventuelle Erweiterungen des Systems könnten z.B. gezielt Mehrwortausdrücke (siehe Beispiele (17) bis (19)) untersuchen. In einigen Fällen haben die Tokens einzeln andere Lesarten als im Mehrwortausdruck. Darüberhinaus ist denkbar, das Web als Korpus einzusetzen. Ein Vorteil wäre die höhere Aktualität: Beispiele (20) bis (23) sind in den verwendeten Korpora nicht enthalten und können daher nicht akzeptiert (Beispiele (20) bis (22)) bzw. verworfen (Beispiel (23)) werden.

(17) Toll Collect

(18) Elf Aquitaine

(19) Paris Hilton

(20) eBay

(21) e.on

(22) 50 Cent

(23) Ich AG

Um die Erkennungsrate zu erhöhen, kann zusätzlich zu den oben genannten Kategorien die Kategorie `misc` (miscellaneous) an nicht klassifizierbare Entitäten vergeben werden.

Eine weitere Verbesserung für die NER würde sich aus einer vorgeschalteten topologischen Felderanalyse ergeben: im Vorfeld (VF) darf nur eine Konstituente stehen, d.h. aneinandergrenzende NEs müssen einen Mehrwortausdruck bilden (vgl. Beispiele (24) und (25)).

(24) [Der neue Ford Fokus]$_{VF}$ wird in Saarlouis produziert.

(25) [Der Ayers Rock]$_{VF}$ ist ein aus Arkosesandstein bestehender Monolith.

Schließlich könnte der Begriff der Named-Entity erweitert werden: in Anbetracht der Tatsache, dass Datums- und Maßangaben als relevante Entitäten erachtet werden, erscheinen Produkt- und Typbezeichnungen (Beispiele (26) bis (32)) ebenfalls relevant. Dadurch könnte ein Beitrag zur Textklassifikation (z.B. Klassifikation in Werbetexte, Nachrichtentexte, etc.) geleistet werden.

(26) Aspirin

(27) Tesa

(28) Tempo

(29) Persil

(30) BMW 318Ci

(31) Audi A-8

(32) Piper Seneca II

Literatur

Chinchor, N., 1998. *MUC-7 Named Entity Task Definition (Version 3.5)*. In: Proceedings of the Seventh Message Understanding Conference. Fairfax, Virginia, URL `http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_proceedings/overview.html`.

Klatt, S., 2004. *Kombinierbare Textanalyseverfahren für die Korpusannotation und Informationsextraktion*. Dissertation, Universität Stuttgart.

SailLabs, 2004. *Online Named Entity Recognition System für Englisch, Deutsch, Französisch, Spanisch*. URL `http://extftp.sail-technology.com/mms/ned_demo.html`

Schiller, A., 1995. *DMOR Benutzerhandbuch*. IMS, Universität Stuttgart.

Yarowsky, D., 1995. *Unsupervised word sense disambiguation rivaling supervised methods*. In: Meeting of the Association for Computational Linguistics. 189–196.