Eigennamenerkennung zwischen morphologischer Analyse und Part-of-Speech Tagging: ein automatentheoriebasierter Ansatz

Jörg Didakowski
Berlin-Brandenburgische Akademie der Wissenschaft
Digitales Wörterbuch der deutschen Sprache
Jägerstr. 22/23
10117 Berlin
didakowski@ling.uni-potsdam.de

Alexander Geyken
Berlin-Brandenburgische Akademie der Wissenschaft
Digitales Wörterbuch der deutschen Sprache
Jägerstr. 22/23
10117 Berlin
geyken@bbaw.de

Thomas Hanneforth
Institut für Linguistik, Computerlinguistik
Universität Potsdam
Karl-Liebknecht-Str. 24-25
D-14476 Potsdam
tom@ling.uni-potsdam.de

Abstract Previous rule-based approaches for Named Entity Recognition (NER) in German base NER on Part-of-Speech tagged texts. We present a new approach where NER is situated between morphological analysis and Part-of-Speech Tagging and model the NER-grammar entirely with weighted finite state transducers (WFST). We show that NER strategies like the resolution of proper noun/common noun or company-name/family-name ambiguities can be formulated as a Best-Path-function of a WFST. The frequently used second pass resolution of coreferential Named Entities can be formulated as a re-assignment of appropriate weights. A prototypical NE recognition system built on the basis of WSFT and large lexical resources was tested on a manually annotated corpus of 65,000 tokens. The results show that our system compares in recall and precision to existing rule-based approaches.

Keywords: *Named Entity Recognition, weighted finite-state transducers, large lexical resources*

1 Einleitung

Nicht zuletzt durch die Förderung im Rahmen der MUC-Konferenzen (*Message Understanding Competition*, MUC-7, 1998) stellt die Eigennamenerkennung Gegenstand zahlreicher Arbeiten dar. In den MUC-Konferenzen wurden Eigennamen in folgende Kategorien eingeteilt: Personen, Unternehmen, geographische Ausdrücke, Datumsangaben und Maßangaben. Mit einer Quote von bis zu 97% Vollständigkeit (Recall) bzw. 95% Korrektheit (Precision) (z.B. Mikheev et al., 1998, 1999; Stevenson and Gaizauskas, 2000) gilt das Problem der Eigennamenerkennung (im Sinne einer Markierung von Eigennamen) für das Englische als zufriedenstellend gelöst.

Im Deutschen ist die Eigennamenerkennung gegenüber dem Englischen vor allem dadurch erschwert, dass Eigennamen und Nomen nicht aufgrund der Groß- und Kleinschreibung unterschieden werden können. Somit können eine Reihe von Regeln zur Erkennung von Eigennamen, die im Englischen zur erfolgreichen Erkennung entscheidend beitragen, im Deutschen nicht angewendet werden. Ein Beispiel hierfür ist die Regel VORNAME GEFOLGT VON GROSSGESCHRIEBENEM WORT = PERSONENNAME, die im Deutschen für Kontexte wie *schreibt Klaus Software* oder *Ist Luise Linguistin* die Appellativa *Software* und *Linguistin* fälschlicherweise als Eigennamen identifizieren.

Für das Deutsche liegen auf der einen Seite ressourcenarme Systeme vor, bei denen Eigennamenkontexte mit maschinellen Lernverfahren gelernt werden (Quasthoff and Biemann, 2002, Rössler, 2004), andererseits regel- und lexikonbasierte Systeme, bei denen die Eigennamen aufgrund ihrer Kontexte und lexikalischer Bedingungen identifiziert werden (z.B. Volk and Clematide, 2001, Neumann and Piskorski, 2002). Abgesehen von Rössler (2004), der seinen Ansatz anhand des manuell annotierten, 250.000 Tokens umfassenden CONLL-03 Korpus evaluiert,

wurden die Systeme mit Testsätzen bzw. kleineren Testkorpora ausgewertet. Die Vollständigkeit und Korrektheit der verschiedenen Systeme sind somit nur schwer vergleichbar. Keines der genannten Systeme weist jedoch eine vergleichbar hohe Erkennungsrate auf wie die oben aufgeführten Systeme für das Englische. So erreichen Quasthoff and Biemann (2002) bei ihrem System auf der Basis von 1000 Testsätzen eine Korrektheit von 97,5% bei der Erkennung von Personennamen, die Vollständigkeit liegt jedoch nur bei 71,5%. Bei dem Verfahren von Rössler (2004) werden 78% aller Personennamen erkannt, die Korrektheit hingegen liegt nur bei 71%. Besser sieht dies bei den regelbasierten Systemen aus. Volk and Clematide (2001) geben bei der Evaluation von 990 Sätzen aus der *Computer-Zeitung* eine Erkennungsrate der Personennamen von 86% und eine Korrektheit in 92% aller Fälle an. Ähnlich verhält sich das System von Neumann and Piskorski (2002), welches auf einer Grundlage von 20.000 Tokens der Wirtschaftswoche evaluiert wurde. Hier lagen Vollständigkeit und Korrektheit der Personennamenerkennung bei 81% bzw. 96%. Bei allen genannten Systemen liegt die Erkennung von Organisationsnamen und geographischen Namen, sofern sie diese durchführen, schlechter.

Das hier vorgestellte System ist regel- und lexikonbasiert. Im Unterschied zu den beiden oben genannten Systemen, die ‘gemischte‘ Methoden verwenden (Volk and Clematide, 2001) bzw. POS-Tagger vor die Eigennamenerkennung setzen (Neumann and Piskorski, 2002), siedelt der hier vorliegende Ansatz die Eigennamenerkennung zwischen morphologischer Analyse und dem Part-of-Speech-Tagging an. Somit können die Regeln zur Eigennamenerkennung sowohl auf die morphologischen Analysen als auch auf die Ambiguitäten, insbesondere die Homographie, zurückgreifen. In der Folge werden die Grundideen des hier vorgestellten Systems skizziert (Abschnitt 2). In Abschnitt 3 werden die formalen Grundlagen zur automatenbasierten Reformulierung des Eigennamenerkennungsproblems erläutert. Die lexikalischen Ressourcen sowie das auf den hier beschriebenen theoretischen Grundlagen implementierte Eigennamenerkennungssystem SynCoP werden in den Abschnitten 4 und 5 beschrieben. Schließlich erfolgt in Abschnitt 6 eine Evaluation des Systems anhand eines 65.000 Tokens umfassenden Zeitungskorpus.

2 Ziele und Grundideen der Eigennamenerkennung

Mit dem hier vorgestellten Ansatz zeigen wir, wie sich die wesentlichen Komponenten der regel- und ressourcenbasierten Eigennamenerkennung ausschließlich mittels gewichteter endlicher Transduktoren formulieren lassen. Wir stellen dar, wie es mit automatenbasierten Mitteln möglich ist, die Homographie als Teil des Verfahrens zu behandeln, Nicht-Monotonie zu realisieren, regelbasiert Eigennamenkontexte zu ermitteln, und wie Koreferenzbeziehungen zwischen Eigennamen bestimmt werden können.

Eine Grundidee des Ansatzes ist es, die Homographie von Nomen als Teil des Verfahrens zu

begreifen, d.h. die Disambiguierung innerhalb der Eigennamengrammatik durchzuführen und nicht, wie beispielsweise in den Ansätzen von Stevenson and Gaizauskas (2000) oder Neumann and Piskorski (2002), durch die Vorschaltung eines POS-Taggers. Für die Homographieauflösung werden drei Homographietypen unterschieden: die Homographie zwischen (lexikalischen) Kategorien, also beispielsweise zwischen Eigenname und Appellativum, die 'Binnenhomographie', also die Homographie zwischen Vor- und Nachnamen, geographischen Namen und Organisationsnamen, und schließlich systematische Metonymien, wie beispielsweise diejenige zwischen Institutionen und Gebäuden, die für die Disambiguierung von Organisations- und geographischen Namen von Bedeutung ist. Die Aussagekraft der Homographie ist entscheidend von der Vollständigkeit der zugrundeliegenden lexikalischen Ressourcen abhängig. Insbesondere sind dafür eine möglichst vollständige, die produktive Derivation und die Komposition des Deutschen berücksichtigende Morphologie, sehr umfangreiche Eigennamenlisten sowie ein semantisches Klassensystem für Nomen notwendig (s. Abschnitt 4).

Die Erstellung von absolut vollständigen Ressourcen ist in der Praxis nicht möglich, da in Zeitungstexten grundsätzlich neue Eigennamen auftauchen und die Ressourcen somit immer nur näherungsweise vollständig sein können. Das Regelsystem muss daher in der Lage sein, unbekannte Namen in geeigneten Kontexten als Eigenname zu klassifizieren. Darüber hinaus wird durch neu hinzutretende Namen eine Nicht-Monotonie eingeführt, d.h. bei neu hinzukommenden Namen muss gegebenenfalls lexikalisches Wissen überschrieben werden, wenn es Kontexte gibt, die eine ausreichende Evidenz für die Änderung liefern. Beispielsweise ist dies der Fall, wenn eine Wortform aufgrund des lexikalischen Wissens des System als eindeutiges Appellativum klassifiziert wird, aber der Kontext der Wortform eindeutig auf einen Nachnamen schließen läßt.

In der Folge bezeichnen wir unter Bezugnahme auf McDonald (1996) *interne Evidenz* als das aus den lexikalischen Ressourcen gewonnene Wissen, also beispielsweise die Tatsache, dass es sich bei *Sachsen* sowohl um einen Staat als auch um die Einwohner handelt, oder dass *Beiersdorf* sowohl Familienname als auch Unternehmensname sein kann. *Externe Evidenz* sind die aus dem Kontext des Namens gewonnenen Informationen. Dies sind namensinterne Kontexte wie Titel, Namenszusätze oder Initialen, und appositive Kontexte, wie beispielsweise Funktions- oder Organisationsbezeichner.

Die im vorgestellten System verwendeten Regeln zur Eigennamenerkennung nutzen eine Kombination aus Informationen zur Homographie und zur internen bzw. externen Evidenz. So können homographe Nomen wie *Fischer*, *Braun* oder *Ordnung* zu Personennamen werden, wenn sie mit eindeutigen Kontexten wie *Außenminister*, *Dr.* oder *Carl* verknüpft sind. Dies geschieht durch Formulierung einer Regel, derzufolge zwei adjazente großgeschriebene Wörter `A B` zu einem Personennamen werden, wenn `A` u.a. homograph zu einem Vornamen und `B` u.a. homo-

graph zu einem Nachnamen ist. Falls `B` nur Appellativum ist, trifft diese Regel nicht zu. Damit wird verhindert, dass bei Kontexten wie *weil Karl Software entwickelt* das Nomen *Software* als Familienname erkannt wird – unter der Annahme, dass das System weiß, dass *Software* nur Appellativum, nicht aber Familienname ist. Voraussetzung hierfür sind natürlich sehr große Ressourcen, auf die in Abschnitt 4 näher eingegangen wird. Volk and Clematide (2001) lösen dieses Problem, indem sie `A B` als *priming-Kontext* ansetzen: Nur wenn in einer näheren Umgebung von 15 Sätzen (diese Zahl wurde empirisch ermittelt) ein einzelnes `B` auftaucht, wird `A B` ein gültiger Eigenname. Obwohl durchaus vorstellbar ist, dass das Nomen *Software* in der Umgebung von 15 Sätzen noch einmal isoliert auftaucht, scheint dieser Fall in der Praxis äußerst selten aufzutreten (Volk and Clematide, 2001). Umgekehrt scheint auch die andere mögliche Fehlerquelle, nämlich dass ein Eigenname nicht mehr isoliert in der näheren Umgebung auftaucht, in der Praxis selten vorzukommen (Volk and Clematide, 2001).

Die Koreferenzauflösung ist ein weiterer notwendiger Mechanismus zur Eigennamenerkennung. Damit sollen Namen mit unzureichender Evidenz erkannt werden, wenn sie im Text an anderer Stelle mit hinreichender Evidenz stehen. Volk and Clematide (2001) realisieren dies, wie oben erwähnt, über einen *priming-Kontext*, bei Neumann and Piskorski (2002) wird die Koreferenz mittels eines *dynamischen Lexikons* erreicht, in welches die Eigennamenkandidaten zur Laufzeit gespeichert werden, die dann mit den bereits vorher aufgrund von externer Evidenz als sicher erkannten Eigennamen abgeglichen werden. Auch im Rahmen des hier verfolgten Ansatzes werden Kandidatenmengen gebildet. Kandidaten für Koreferenzen sind Folgen von einem oder mehreren großgeschriebenen Wortformen. Diese bezeichnen wir in der Folge als potenzielle Eigennamen bzw. für die Modellierung mit gewichteten Transduktoren als Eigennamen mit internem Potenzial. Diese Kandidaten können zu sicheren Eigennamen werden, wenn sie in Lemma, lexikalischer Kategorie und Homographietyp mit einem als sicher erkannten Eigennamensteil übereinstimmen.

3 Formale Grundlagen

In diesem Abschnitt fassen wir das Eigennamenerkennungsproblem als *Klammerungsproblem* auf. Gegeben ein vorverarbeiteter und morphologisch annotierter Eingabetext x [1] und eine Menge von Eigennamenkontexten α wollen wir die Vorkommen von Elementen aus α in x möglichst ambiguitätsfrei bestimmen, indem wir sie jeweils in ein Klammerpaar [und] einfassen. Eigennamenkontexte enthalten Zeichenketten, die den verschiedenen Eigennamenkategorien (Personen-

[1] Der Text wird tokenisiert und mit (in der Regel ambigen) morphologischen Annotationen versehen. Dabei alternieren die von der Morphologie für die Textwörter bestimmten Lemmata mit Informationen über deren Wortart und deren morphosyntaktische Merkmale, so dass sich eine Eigennamengrammatik gleichzeitig auf alle diese Informationen beziehen kann.

, Geo-, Firmen- und Produktname) zugeordnet sind, und möglicherweise darüber hinaus Wortmaterial und kategorielle Annotationen, welche auf Eigennamen hinweisen können. Eigennamenkontexte sind beispielsweise[2]:

1. (Letter+) {NE Nametype=firstname} (Letter+) {NE Nametype=lastname}
2. (Letter+) {NN SemClass=k_l_h_m_eig_aktm} (Letter+) {NE Nametype=lastname Homograph=yes}

Der erste Ausdruck denotiert Zeichenketten wie Jürgen{NE Nametype=firstname} Trittin{NE Nametype=lastname}, während der zweite Ausdruck Zeichenketten von einem eine Berufsbezeichnung denotierenden Nomen gefolgt von einem homographen Nachnamen charakterisiert (z.B. Bundeskanzler {NN SemClass=k_l_h_m_eig_aktm_taet} Schüssel {NN Nametype=lastname}). Die semantischen Klassen werden dabei durch den in Abschnitt 4.2 beschriebenen Nomen-Thesaurus definiert. Die Vereinigung aller Eigennamenkontextspezifikationen bezeichnen wir als *Eigennamengrammatik*.

Im Folgenden abstrahieren wir völlig von den für die Eigennamenerkennung nötigen regulären Ausdrücke und verwenden stattdessen sehr einfache reguläre Mengen.

3.1 Eigennamenerkennung als Klammerungsproblem

Automatentheoretische Klammerungsverfahren können prinzipiell in optionale und obligatorische Verfahren unterschieden werden (cf. Karttunen, 1995).

Die Übersetzung einer Regel zur optionalen Klammerung von Ausdrücken $\alpha \rightarrow [\ \dots\ ]$, die jedes Element der (möglicherweise unendlichen) regulären Sprache $\alpha \subseteq \Sigma^*$ (für ein gegebenes Alphabet Σ) optional in die Klammersymbole [und] einfasst, ist eine reguläre Relation[3]

$$(ID(\Sigma^*) \cdot (\epsilon \times [) \cdot ID(\alpha) \cdot (\epsilon \times]))^* \cdot ID(\Sigma^*) \tag{1}$$

Abb. 1 zeigt einen Klammerungstransduktor $T_{ab|bc}$ für $\alpha = \{ab, bc\}$ und $\Sigma = \{a, b, c\}$

Da α Teilmenge von Σ^* ist, hat der Transduktor aus Abb. 1 die Wahl, Elemente von α in der Schleife an Zustand 0 zu überlesen und somit ungeklammert zu lassen. Dies führt zu einem nicht-funktionalen Transduktor, der eine Eingabekette auf mehrere Ausgaben abbildet. $T_{ab|bc}$ beispielsweise bildet die Eingabe $abbc$ auf die vier Ausgabeketten $[ab][bc]$, $[ab]bc$, $ab[bc]$ und $abbc$ ab.

[2]Für die Annotierung verwenden wir das Stuttgart-Tübingen Tagset (STTS), vgl. `http://www.ims.uni-stuttgart.de/projekte/CQPDemos/Bundestag/help-tagset.html`.

[3]In unserer Notation machen wir freien Gebrauch von den bekannten Äquivalenzen zwischen regulären Mengen, regulären Ausdrücken und endlichen Automaten. Sind X und Y reguläre Mengen und R_1 und R_2 reguläre Relationen, so bezeichnet X^* die Sternhülle von X, X^+ die Plushülle von X, $\overline{X}$ das Komplement von X (also $\Sigma^* - X$), $X \cdot Y$ die Mengenverkettung, $X \times Y$ das Kreuzprodukt und $X - Y$ die Differenz von X und Y. $ID(X)$ bezeichnet die binäre Identitätsrelation zu X, $R_1 \circ R_2$ die Komposition von R_1 und R_2, $R_1 \cdot R_2$ die Verkettung von R_1 und R_2 und $Proj_2(R_1)$ die zweite Projektion von R_1. Vgl. dazu auch Hopcroft and Ullman (1979).

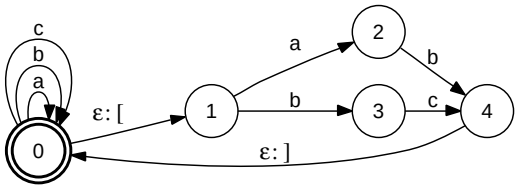

Abbildung 1: *Optionale Klammerung von* $\alpha = \{ab, bc\}$

Demgegenüber ist die obligatorische Interpretation von $\alpha \rightarrow [\ \ldots\ ]$ durch die reguläre Relation

$$(ID(\overline{\Sigma^* \cdot \alpha \cdot \Sigma^*}) \cdot (\varepsilon \times [) \cdot ID(\alpha) \cdot (\varepsilon \times])) ^* \cdot ID(\overline{\Sigma^* \cdot \alpha \cdot \Sigma^*}) \tag{2}$$

gegeben. Abb. 2 zeigt einen Transduktor, der die obligatorische Klammerung einer Eingabe vollführt. Um das Klammern der Elemente in α zu erzwingen, muss verhindert werden,

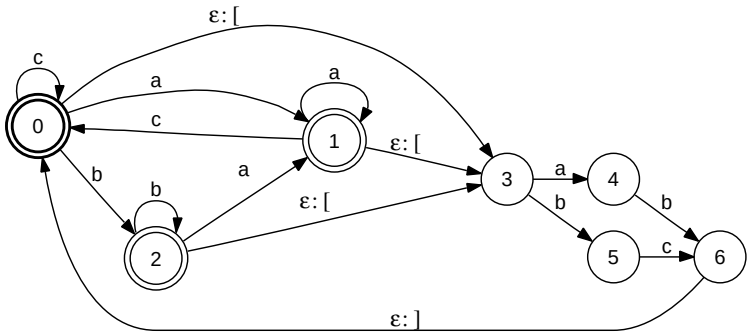

Abbildung 2: *Obligatorische Klammerung von* $\alpha = \{ab, bc\}$

dass der Transduktor α in einer $ID(\Sigma^*)$ entsprechenden Schleife überliest. Dies wird durch eine Komplementierungsoperation erreicht, bei der allerdings nicht nur α, sondern vielmehr $\Sigma^* \cdot \alpha \cdot \Sigma^*$ komplementiert werden muss. Die reguläre Sprache $\Sigma^* \cdot \alpha \cdot \Sigma^*$ enthält alle Zeichenketten, die ein Element aus α enthalten, demzufolge ist $\overline{\Sigma^* \cdot \alpha \cdot \Sigma^*}$ die Sprache, die kein Element aus α enthält. Unter der Voraussetzung, dass α keine Elemente enthält, die in Präfix- oder Suffixbeziehung zueinander stehen, ist das Ergebnis von (2) ein Transduktor, der eine totale Funktion $\Sigma^* \mapsto (\Sigma \cup \{[,]\})^*$ realisiert, d.h. jede Zeichenkette aus Σ^* wird auf genau eine Ausgabezeichenkette abgebildet. Reguläre Sprachen und damit endliche Automaten sind nun unter

Komplementierung abgeschlossen (cf. Hopcroft and Ullman, 1979). Hierbei muss der endliche Automat, der die zu komplementierende Sprache beschreibt, zunächst determinisiert und anschließend vervollständigt werden, so dass seine Übergangsfunktion $\delta : Q \times \Sigma \mapsto Q$ zu einer totalen Funktion wird. Schließlich werden Endzustände und Nichtendzustände vertauscht.

Für Anwendungszwecke ist allerdings problematisch, dass die Determinisierung von endlichen Automaten in $O(2^n)$ ist, sich im schlimmsten Fall also ein exponentielles Anwachsen der Zustände ergeben kann. Ist beispielsweise $\alpha = a \cdot (a|b)^k$, so hat der $\Sigma^* \cdot \alpha$ entsprechende nichtdeterministische Automat $k+2$ Zustände, der äquivalente deterministische Automat jedoch $2^{k+1} + 1$ Zustände[4]. Hinzu kommt die Vervollständigung jedes Zustands bei der Komplementierung, so dass ein Komplementautomat im schlimmsten Fall $O(2^n|\Sigma|)$ Übergänge aufweist, wenn n die Zustandsanzahl des nichtdeterministischen Automat ist.

Diese Tatsache kann nun die Anwendung der obligatorischen Klammerung bei Mustermengen α mit einer höheren Komplexität des α repräsentierenden endlichen Automaten, wie sie für die Zwecke der Eigennamenerkennung erforderlich sind, inpraktikabel werden lassen, da sie hohe Compilationszeiten und große Automaten nach sich ziehen kann [5].

Auf der anderen Seite weist die optionale Klammerung eine sehr viel geringere Komplexität auf. Für einen gegebenen endlichen Automaten α ist sie in $O(\alpha)$, da hierbei nur Operationen wie Verkettung und Hüllenbildung Verwendung finden, deren Komplexität linear zur Größe ihrer Operanden ist. Das Problem der optionalen Klammerung ist aber ihr inhärenter Nichtdeterminismus, da es für jede gefundene Instanz v aus α, die Teilkette der Eingabekette w ist, zwei Ausgabeketten gibt: einmal $[v]$ und einmal v. Enthält w k Instanzen aus α, so erzeugt die optionale Klammerung 2^k Ausgabeketten.

Abschnitt 3.2 zeigt, wie die Erweiterung des Transduktors um numerische Gewichte dieses Problem löst.

3.2 Klammerung mit gewichteten endlichen Automaten

Ein gewichteter endlicher Transduktor T bzgl. einer Gewichtsstruktur W ist ein 8-Tupel $\langle \Sigma, \Delta, Q, q_0, F, E, \lambda, \rho \rangle$ mit

1. Σ, dem endlichen *Eingabealphabet*
2. Δ, dem endlichen *Ausgabealphabet*
3. Q, einer endlichen Menge von Zuständen

[4] Man beachte, dass das exponentielle Anwachsen der Zustände durch das Σ^*-Präfix in $\Sigma^* \cdot \alpha$ entsteht.

[5] In Hanneforth (2006) wird von einem Experiment berichtet, bei dem ein α repräsentierender endlicher Automat mit 177 Zuständen und 7.449 Übergängen zu einem Klammerungstransduktor mit 4.115 Zuständen und knapp 3 Mio. Übergängen führte.

4. $q_0 \in Q$, dem Startzustand

5. $F \subseteq Q$, der Menge der Endzustände

6. $E \subseteq Q \times \Sigma \cup \{\epsilon\} \times \Delta \cup \{\epsilon\} \times W \times Q$, der Menge der *Übergänge*

7. λ, dem *Initialgewicht* und

8. $\rho : F \mapsto W$ der *Endzustandsgewichtsfunktion*

Gegenüber der Definition eines ungewichteten Transduktors tragen Übergänge sowie Endzustände demnach zusätzlich ein Gewicht aus einer Menge W. Diese Gewichte müssen die Axiome einer algebraischen Struktur, eines sog. *Semirings* erfüllen. Eine Struktur $\langle W, \oplus, \otimes, \overline{0}, \overline{1} \rangle$ ist ein Semiring (Kuich and Salomaa, 1986), wenn sie die folgenden Bedingungen erfüllt:

1. $\langle W, \oplus, \overline{0} \rangle$ ist ein kommutativer Monoid mit $\overline{0}$ als dem neutralen Element bzgl. $\oplus$.
2. $\langle W, \otimes, \overline{1} \rangle$ ist ein Monoid mit $\overline{1}$ als dem neutralen Element bzgl. $\otimes$.
3. $\otimes$ distribuiert über $\oplus$.
4. $\overline{0}$ ist ein Annihilator for $\otimes$: $\forall w \in W, w \otimes \overline{0} = \overline{0} \otimes w = \overline{0}$.

Die Interpretation der Gewichte aus W im gewichteten Transduktor wird durch folgende Formel hergestellt (cf. Mohri, 2002):

$$\Omega(x) = \bigoplus_{\pi \in \prod(\{q_0\}, x, F)} \lambda \otimes \omega(\pi) \otimes \rho(n(\pi)) \tag{3}$$

Das Gewicht $\Omega(x)$, welches einer Eingabekette $x \in \Sigma^*$ zugewiesen wird, berechnet sich wie folgt: es sei π ein Pfad vom Startzustand des Transduktors q_0 zu einem Endzustand $\in F$, so dass die Verkettung der Eingabesymbole entlang dieses Pfades gerade x ergibt. Das Gewicht von π ermittelt sich aus der abstrakten Multiplikation des Initialgewichtes λ, gefolgt vom Gewicht des Pfades $\omega(\pi)$ multipliziert mit dem Gewicht, welches die Endzustandsgewichtsfunktion ρ dem Zustand, mit dem π endet ($n(\pi)$), zuweist. Besteht π aus k Übergängen $\langle q_0, x_1, y_1, w_1, p_1 \rangle \langle p_1, x_1, y_1, w_1, p_2 \rangle \ldots \langle p_{k-1}, x_k, y_k, w_k, p_k \rangle$ mit $p_k \in F$, so ist $\omega(\pi)$ gleich $w_1 \otimes w_2 \otimes \cdots \otimes w_k$. In kommutativen Semiringen — d.h. solchen, in denen auch die $\otimes$-Operation kommutativ ist — ist die Reihenfolge der Gewichte w_i irrelevant, zudem können Gewichte wegen der Monoid-Eigenschaft auch zusammengefasst werden. Dies ermöglicht Freiheitsgrade bei der Realisierung der Gewichte an den Übergängen eines Pfads.

Gibt es im Automaten mehrere Pfade $\pi_1 \ldots \pi_m$ für x (dies kann beispielsweise in einem nichtdeterministischen Transduktor der Fall sein), so werden die Gewichte von $\pi_1 \ldots \pi_m$ mit

der abstrakten Additionsoperation $\oplus$ kombiniert. Da $\oplus$ kommutativ ist, spielt dabei die Kombinationsreihenfolge der Pfade keine Rolle.

Für die Zwecke der gewichteten Klammerung verwenden wir einen sog. *tropischen Semiring*, bei dem $\otimes$ mit der arithmetischen Addition und $\oplus$ mit der Minimumsoperation instantiiert ist. Genauer ist ein tropischer Semiring ein Quintupel $\langle \mathbb{R} \cup \{\infty\}, min, +, \infty, 0\rangle$ mit der Menge der reellen Zahlen $\mathbb{R}$ als Trägermenge, ∞ als neutralem Element der Minimumfunktion ($\forall x \in \mathbb{R}\ , x\ min\ \infty = \infty\ min\ x = x$) und 0 als neutralem Element der Addition. Eingesetzt in (3) bedeutet das, dass Gewichte entlang eines Pfades *addiert* werden und bei alternativen akzeptierenden Pfaden das *Minimum* der aufkumulierten Gewichte genommen wird. Abb. 3 zeigt einen mit dem tropischen Semiring gewichteten Transduktor mit $\Sigma = \Delta = \{a, b, c, x\}$, der ab obligatorisch durch x mit Gewicht 1 ersetzt[6]. Beispielsweise wird eine Eingabekette *cabbabb*

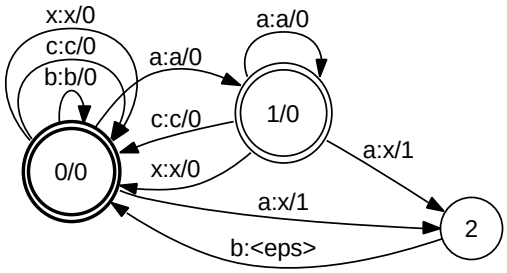

Abbildung 3: *Gewichteter Transduktor für* $ab \rightarrow x\langle 1\rangle$

zu *cxbxb* mit $\Omega(cabbabb) = 2$ (da 2 Vorkommen von *ab* ersetzt wurden und jede einzelne Ersetzung mit 1 gewichtet wurde. Da der Transduktor in Abb. 3 zwar nichtdeterministisch, jedoch funktional ist, gibt es nur einen akzeptierenden Pfad je Eingabekette x, so dass die Minimumoperation aus (3) keine Rolle spielt.

Ihre Nützlichkeit entfalten tropisch gewichtete Automaten jedoch im Kontext optionaler Ersetzungen bzw. Klammerungen. Hierbei versehen wir die Klammerung selbst mit einem *negativen* numerischen Gewicht, d.h. die Klammerungsregeln haben nun die Form[7]

$$\alpha \rightarrow [\ \ldots\ ]\ \langle w\rangle \tag{4}$$

[6]Gewichte (an Übergängen und Endzuständen) werden nach einem Schrägstrich dargestellt
[7]Semiring-Gewichte werden in $\langle\ \rangle$ notiert.

wobei $w \in \mathbb{R}$ und $w < 0$. Formal ist dies zulässig, da gewichtete endliche Automaten abgeschlossen sind unter Verkettung, Vereinigung und Hüllenbildung. Sie sind nicht abgeschlossen unter Komplementierung, komplementiert wird unter der obligatorischen Interpretation der Klammerungsregel jedoch nur der ungewichtete endliche Automat, der α entspricht. Abb. 4 zeigt einen gewichteten Transduktor, der äquivalent ist zur gewichteten Regel $ab|bc \rightarrow [\ \ldots\ ]\langle -1\rangle$ (mit $\Sigma = \{a, b, c, x\}$). Die (vorläufige Version der) Anwendung eines gewichteten Transduktors

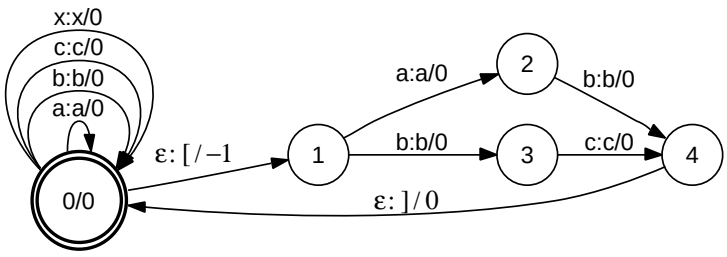

Abbildung 4: *Gewichteter Transduktor* $T_{ab|bc}$ *für* $ab|bc \rightarrow [\ \ldots\ ]\langle -1\rangle$

T auf eine Eingabekette x entspricht formal der Komposition des Identitätstransduktors[8] für x mit T_{rule} und anschließender Projektion des Ausgabebandes:

$$Apply(T_{rule}, x) =_{def} Proj_2(ID(x) \circ T_{rule}) \tag{5}$$

Abb. 5 zeigt das Ergebnis von $Apply(T_{ab|bc}, abbc)$. Die gewichtete Sprache des Automaten aus

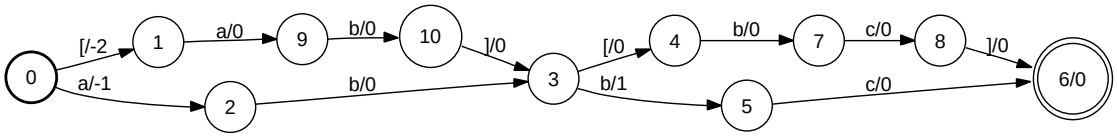

Abbildung 5: $Apply(T_{ab|bc}\ , abbc)$

Abb. 5 ist $\{\langle abbc, 0\rangle, \langle [ab]bc, -1\rangle, \langle ab[bc], -1\rangle, \langle [ab][bc], -2\rangle\}$. Wenden wir nun auf diesen Automaten die abstrakte Additionsoperation aus (3) an — also *min* im tropischen Semiring — so präferieren wir den Pfad im Automaten mit dem geringsten Gewicht, d.h. denjenigen mit den meisten Anwendungen der ursprünglichen Klammerungsregel. Die endgültige Version von

[8]Der Identitätstransduktor für eine Eingabekette $x = a_1 a_2 \ldots a_k$ ist ein linearer Transduktor mit $k+1$ Zuständen, Startzustand q_0, Endzustand q_k und einer Übergangsmenge $\{\langle q_i, a_i, a_i, \overline{1}, q_{i+1}\rangle \mid 0 \leq i < k)\}$.

$Apply(T_{rule}, x)$ lautet dann:[9]

$$Apply(T_{rule}, x) =_{def} Bestpath(Proj_2(ID(x) \circ T_{rule})) \tag{6}$$

Unter der Annahme, dass die zu klammernde reguläre Menge nicht ϵ enthält, ist das Ergebnis der Komposition des azyklischen Transduktors $ID(x)$ mit dem zyklischen Klammerungstransduktor T_{rule} selbst auch azyklisch. Die Komplexität der Komposition zweier Transduktoren mit n bzw. m Zuständen ist in $O(nm)$. Da der Klammerungstransduktor selbst eine Konstante ist, hängt die Kompositionskomplexität nur von der Zustandsanzahl des Identitätstransduktors für x ab. Die Anzahl der Zustände und Übergänge in $ID(x) \circ T_{rule}$ wächst demnach linear mit der Länge von x, so dass die Komplexität der Komposition in $O(|x|)$ ist.

Für azyklische Automaten kann man sehr effizient beste Pfade bestimmen: Man bedient sich dazu des Algorithmus der sog. *Relaxierung in topologischer Ordnung* (cf. Lawler, 1976). Dieses Verfahren ist für einen endlichen Automaten mit $|Q|$ Zuständen und $|E|$ Übergängen in $O(|Q| + |E|)$ und hat auch mit negativen Gewichten keine Probleme. Zusammengefasst ist $Apply(T_{rule}, x)$ demzufolge in $O(|x|)$, d.h. linear zur Länge der Eingabekette x.

3.3 Klammerung bei überlappenden Mustern

Stehen die Elemente der zu klammernden regulären Sprache α in Präfix- und/oder Suffixbeziehung zueinander, so kann die Anwendung einer Klammerungsregel $a \rightarrow [\ \ldots\ ]\langle -1 \rangle$ auf eine Eingabekette zu mehreren Ausgabeketten führen. Ist α beispielsweise ab^+ und die Eingabekette $x = ab^k$, so ist $|Apply(T_{ab^+}, x)| = k$, wobei alle Zeichenketten in $Apply(T, x)$ das gleiche Gewicht -1 tragen. Zur Disambiguierung wird nun üblicherweise (cf. Abney, 1996) eine *Longest-Match*-Bedingung verwendet: Es wird diejenige Ausgabekette bevorzugt, bei der das längste Element aus α, welches Teilkette von x ist, geklammert wird. Interessanterweise kann diese Bedingung, die automatentheoretisch gedeutet den Längenvergleich verschiedener akzeptierender Pfade notwendig macht, als reguläre Sprache repräsentiert werden (cf. Karttunen, 1996). Problematisch ist bei diesem Ansatz allerdings, dass er auf einer Komplementierung von $\Sigma^* \cdot \alpha$ beruht, was, wie bereits oben dargestellt, zu einem exponentiell größeren Automaten führen kann. Auch hier können gewichtete Automaten die algorithmische Komplexität in den traktablen Bereich zurückbringen. Die Grundidee hierbei ist einfach die, dass man das Material innerhalb der Klammern [und] gewichtet, z.B. einfach zählt mit einer Funktion $C : \Sigma^* \mapsto W$: $C(x) = |x|$. Dies leistet der folgende gewichtete Akzeptor über dem tropischen Semiring (mit $\Sigma = \{a, b, c\}$): Der nur vom jeweiligen Alphabet abhängige tropische Bewertungsautomat $T^0_{eval}(\Sigma)$ kann nun mit dem Klammerungsregeltransduktor T_{rule} komponiert und anstelle von T_{rule} in (6) verwendet

[9] *Bestpath()* ist der traditionelle Name für $\oplus$ im tropischen Semiring.

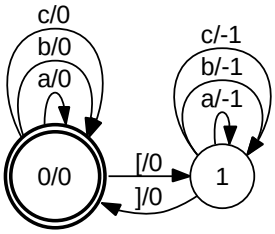

Abbildung 6: *Bewertungsautomat* $T^0_{eval}(\{a, b, c\})$

werden:

$$T'_{rule} =_{def} T_{rule} \circ ID(T^0_{eval}(\Sigma)) \tag{7}$$

Beispielsweise ist $Apply(T'_{ab^+}, cabbcab) = \{\langle c[abb]c[ab], -7\rangle\}$. Hierbei trägt das Material innerhalb der Klammern das Gewicht -5 bei, wozu noch -2 für die beiden Regelanwendungen hinzuaddiert wird.

Allerdings kann es zu Interaktionen zwischen der Gewichtung der optionalen Klammerung und dem Bewerten des Materials innerhalb der Klammerungen kommen. Ist die zu klammernde Menge $\alpha = \{a, aa\}$ und die Eingabekette $x = caac$, so ist $Apply(T'_{a|aa}, caac) = \{\langle c[a][a]c, -4\rangle\}$. Die erwartete Ausgabekette $c[aa]c$ wird nur mit -3 und damit schlechter bewertet. Die Anzahl der Klammerungen und die Summe der Längen der geklammerten Zeichenketten müssen demnach entgegengesetzt bewertet werden. Wird beispielsweise ein Zeichen innerhalb eines Klammerpaars mit $-c_0$ und eine Anwendung der Klammerungsregel mit einer positiven Konstanten c_1 gewichtet und gilt $|c_0| > c_1$[10], so wird bei gleichem Gewicht des bei ambigen Klammerungen jeweils geklammerten Materials diejenige mit weniger Klammern präferiert.

Trotz Verwendung eines Bewertungstransduktors wie in Abb. 6 sind noch nicht alle Ambiguitäten beseitigt. Der gewichtete Transduktor $T'_{a|aa}$ bildet aaa auf zwei gleich gewichtete Ausgabeketten $[aa][a]$ und $[a][aa]$ ab. Eine davon zu präferieren, erfordert eine Stipulation – beispielsweise Karttunens *Longest-Match*-Beschränkung (Karttunen, 1996), die, falls zwei Elemente der regulären Menge α an der gleichen Position in der Eingabekette beginnen, diejenige mit der größeren Ausdehnung nach rechts bevorzugt. Der Preis hierfür ist allerdings die bereits erwähnte Komplementierung von $\Sigma^* \cdot \alpha$. Aus Effizienzgründen lösen wir diese Art von Ambigui-

[10]Der Betrag des Materialgewichts einer Klammerung muss das Gewicht einer einzelnen Anwendung einer Klammerungsregel übersteigen, sonst könnte bei $\alpha = x = a$ nicht zwischen den Ausgabeketten a und $[a]$ unterschieden werden.

tät nicht auf und wählen einfach eine Ausgabekette mit minimalen Gewicht aus. In Didakowski (2005) wird jedoch gezeigt, wie ein spezieller Semiring konstruiert werden kann, der erlaubt, $[aa][a]$ und $[a][aa]$ unterschiedlich zu gewichten.

Der Bewertungsautomat T_{eval} muss nicht so einfach sein wie der in Abb. 6 dargestellte, der nur die Zeichen innerhalb der Klammersymbole zählt. Vielmehr kann er eine beliebige Gewichtsfunktion $\Sigma^* \mapsto W$ repräsentieren, die auch an die zu klammernde Sprache angepasst sein kann. Hinsichtlich der Eigennamenerkennung könnte diese Funktion beispielsweise bestimmte kategorielle Abfolgen besser bewerten als andere. Da dann das Wechselspiel zwischen der Maximierung des Materialgewichts einerseits und der Minimierung der Anzahl der Klammern andererseits nicht mehr in der oben dargestellten, einfachen und vorhersehbaren Weise funktioniert, ist es konzeptuell klarer, beide Bewertungsebenen zu trennen. Dies leistet das sog. *Semiring-Produkt*. Seien $A = \langle W_A, \oplus_A, \otimes_A, \overline{0_A}, \overline{1_A}\rangle$ und $B = \langle W_B, \oplus_B, \otimes_B, \overline{0_B}, \overline{1_B}\rangle$ zwei Semiringe. Das Produkt $A \times B$, gegeben durch

$$A \times B = \langle W_A \times W_B, \oplus_{A\times B}, \otimes_{A\times B}, \langle \overline{0_A}, \overline{0_B}\rangle, \langle \overline{1_A}, \overline{1_B}\rangle\rangle \tag{8}$$

$$\langle x_1, y_1\rangle \oplus_{A\times B} \langle x_2, y_2\rangle =_{def} \langle x_1 \oplus_A x_2, y_1 \oplus_B y_2\rangle, \quad \forall\langle x_1, y_1\rangle, \langle x_2, y_2\rangle \in W_A \times W_B \tag{9}$$

$$\langle x_1, y_1\rangle \otimes_{A\times B} \langle x_2, y_2\rangle =_{def} \langle x_1 \otimes_A x_2, y_1 \otimes_B y_2\rangle, \quad \forall\langle x_1, y_1\rangle, \langle x_2, y_2\rangle \in W_A \times W_B \tag{10}$$

bildet ebenfalls einen Semiring (cf. Kuich and Salomaa, 1986). Für die Aufgabe der Eigennamenerkennung sind A und B tropische Semiringe: A maximiert das Gewicht innerhalb der Klammern, B minimiert die Klammeranzahl. Da die über $A \times B$ definierte *BestPath*-Operation in (6) eine partielle Ordnung der Paare in $W_A \times W_B$ verlangt, definieren wir $\langle x_1, y_1\rangle \leq \langle x_2, y_2\rangle$ wie folgt:

$$\langle x_1, y_1\rangle \leq \langle x_2, y_2\rangle \ =_{def} \ (\langle x_1, y_1\rangle = \langle x_2, y_2\rangle) \vee (x_1 < x_2) \vee (x_1 = x_2 \wedge y_1 < y_2) \tag{11}$$

Auf diese Weise geben wir den Gewichten des Semirings A Priorität vor den Gewichten des Semirings B.

4 Ressourcen: Morphologie und Nomen-Thesaurus

Dieser Abschnitt befasst sich mit der Frage, woher die morphologischen Annotierungen kommen, die durch das im vorhergehenden Abschnitt beschriebenen Klammerungsverfahren bewertet werden. Wir benutzen dazu zwei Ressourcen: das morphologische Analysesystem TAGH und LexikoNet, ein Begriffsnetz deutscher Nomen.

```
<token id="tid78" normalized="false">
  <text>Gewerkschaftsboss</text>
  <analysis id="aid78.1" pos="NN">
    <NN SemClass="k_l_h_m_eig_sozk_stat"
    Gender="masc" Number="sg" Case="nom_acc_dat"/>
    <lemma weight="12">Gewerkschaft/N\s#Boss</lemma>
  </analysis>
  <analysis id="aid78.2" pos="NN">
    <NN SemClass="k_l_h_m_eig_sozk_stat"
    Gender="masc" Number="sg" Case="nom_acc_dat"/>
    <lemma weight="22">Gewerk/N#Schaft/N\s#Boss</lemma>
  </analysis>
</token>
```

Abbildung 7: Analysen für *Gewerkschaftsboss* im XML-Format

4.1 TAGH-Morphologie

Bei dem TAGH-Morphologiesystem (Geyken and Hanneforth, 2006) handelt es sich um eine vollständige, die produktive Derivation und Komposition berücksichtigende Morphologie des Deutschen. Das System basiert auf umfangreichen Stammlexika (s.u.) und erkennt in korrekter Weise über 99% aller Tokens bei neueren Zeitungstexten.

Ausgangspunkt der TAGH-Morphologie sind umfangreiche Morphem- und Wortformenlexika, die mittels verschiedener Compiler in endliche gewichtete Transduktoren übersetzt und dann durch algebraische Operationen in den endgültigen Morphologietransduktor überführt werden. Eigennamenerkennung und Morphologie verwenden somit denselben theoretischen Rahmen.

Die Teillexika der TAGH-Morphologie bestehen aus einem Nomenlexikon (88.000 einfache und komplexe Stämme mit Informationen zur Flexions- und Wortbildung), einem Verblexikon (23.000 Einträge), Lexika zu Adjektiven (18.000) und Adverbien (2.200) sowie einem Lexikon der geschlossenen Formen (2.000 Einträge). Hinzu kommen Lexika von Abkürzungen und Akronymen (20.000) sowie verschiedene Eigennamenlexika: 160.000 geographische Namen, 65.000 Vornamen, 4.500 Organisationsnamen und 240.000 Familiennamen.

Die Ausgabe der TAGH-Morphologie ist pro Wort ein gewichteter endlicher Automat, der die dem Wort zugeordneten Analysen in kompakter Form repräsentiert. Diese Analysen können in beliebige Ausgabeformate umgewandelt werden. Die Abbildung 7 zeigt die XML-Ausgabe für das Wort *Gewerkschaftsboss*. Wie ersichtlich, kann ein Wort auch in linguistisch nicht motivierter Weise segmentiert werden. Über das jeder Analyse zugeordnete Gewicht ist es jedoch möglich, diejenige(n) mit dem geringsten Gewicht auszuwählen. Im Beispiel der Abbildung 7 ist das die Analyse `aid78.1`. Dem Nomen (mit dem STTS-Tag NN markiert) wird daneben noch eine semantische Klasse zugeordnet: *SemClass=k_l_h_m_eig_sozk_stat* bedeutet beispielswei-

se etwa *Mensch mit sozialem Status*. Die im nächsten Abschnitt beschriebenen Grammatiken zur Eigennamenerkennung nehmen auf diese Merkmale Bezug.

4.2 LexikoNet

Die externe Evidenz für Eigennamen besteht zu einem erheblichen Teil aus appositiven Kontexten, in denen Menschen- oder Organisationsbezeichnungen den Eigennamen spezifizieren. Es ist daher von großem Nutzen, entsprechende Substantive erkennen und semantisch zuordnen zu können. Hierfür steht dem System mit LexikoNet (Geyken and Schrader, 2006) eine Liste von etwa 60.000 Menschen-, Organisations- und Ortsbezeichnern zur Verfügung, die gemäß einem Inventar von etwa 1.000 semantischen Kategorien klassifiziert sind. LexikoNet ermöglicht die Klassifikation von Personenbezeichnungen, wie z.B. Berufe (*Bundesfinanzminister*), künstlerische Tätigkeiten (*Orchestermusiker*), Menschen nach ihrer relationalen Zuordnung (*Nachkomme*, *Freund*) oder in ihrer sozialen Stellung (*Gewerkschaftsboss*). Ebenso können Organisationsbezeichnungen wie *Regierung*, *Kommission* klassifiziert werden. Schließlich erfasst LexikoNet auch systematische Metonymien, wie die Institutions-Gebäude Metonymie, wie z.B. *Kirche* oder *Ministerium*.

Aufgrund der Verknüpfung dieser Nomen mit der TAGH-Morphologie können auch Komposita mit Menschenbezeichnern erkannt werden (s.o. das Beispiel *Gewerkschaftsboss*).

5 Das Analysesystem SynCoP

Der Eigennamenerkenner basiert auf dem regelbasierten Analysesystem SynCoP, welches – ebenso wie die im vorigen Abschnitt erwähnte TAGH-Morphologie – auf der *Potsdam Finite State Library* (FSM<2.0>) beruht. Das System wurde für das Chunking, das syntaktische Tagging (cf. Didakowski, 2005) und für die Analyse von Konstituentensatzstrukturen entwickelt. Das SynCoP-System basiert auf den in Abschnitt 3 entwickelten Klammerungsverfahren mit endlichen Transduktoren.

Die Architektur von SynCoP ist in Abbildung 8 skizziert. Im Zentrum des linken Blocks ist der Grammatik-Compiler, der eine Grammatik-Spezifikation in einen Eigennamenklammerer als gewichteten Transduktor überführt. Der mittlere Block zeigt das Anwendungssystem, das den Eigennamenklammerer nach einer Textvorverarbeitungsphase anwendet und so Eigennamen markiert und klassifiziert. Der rechte Block verweist auf weitere Module, die vom Anwendungssystem eingebunden werden: die TAGH-Morphologie und die Koreferenzauflösungsfunktion, mit der Koreferenzen zwischen Eigennamen berechnet werden.

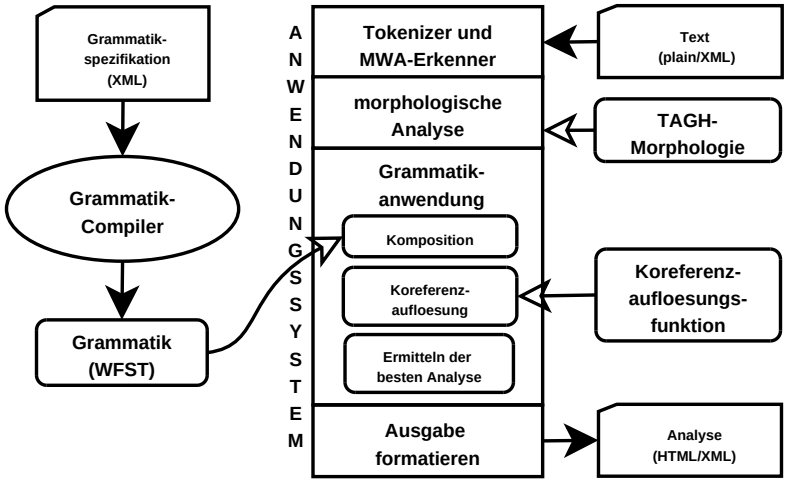

Abbildung 8: Architektur des Analysesystems SynCoP

Im Folgenden werden die Grammatikspezifikation, der Grammatik-Compiler, die Koreferenzauflösungsfunktion und das Anwendungssystem beschrieben.

5.1 Die Grammatikspezifikation

SynCoP unterscheidet streng zwischen Grammatik und Parser und macht es möglich, verschiedene Grammatiken für unterschiedliche Teilaufgaben zu formulieren und diese zu kombinieren.

In der Grammatikspezifikation können interne und externe Evidenz sowie das interne Potenzial als Menge von Eigennamenkontexten unter Verwendung gewichteter regulärer Ausdrücke formuliert werden. Zur Erstellung der Grammatik steht ein Formalismus mit relativ großer Ausdruckskraft zur Verfügung: es können Muster in einer XML-Notation formuliert und gewichtet und Informationen innerhalb von Mustern umgeschrieben werden. Aus den Regeln wird dann durch das in Abschnitt 3 beschriebene Klammerungsverfahren ein Eigennamenmarkierer erzeugt.

5.1.1 Aufbau des vorverarbeiteten Eingabetexts

Nach der Vorverarbeitungsphase und der morphologischen Analyse (s. 4) kann der Eingabetext durch einen regulären Ausdruck

$$(Vorspann \cdot Lemma \cdot (Kategorie_1|Kategorie_2|\ldots|Kategorie_n) \cdot Etikett)^+ \qquad (12)$$

beschrieben werden. Dieses Format stellt auch die Basis für die Grammatikentwicklung dar. Abb. 9 demonstriert dieses (als Automat) am Beispiel der Analyse des Worts *Fischer*.

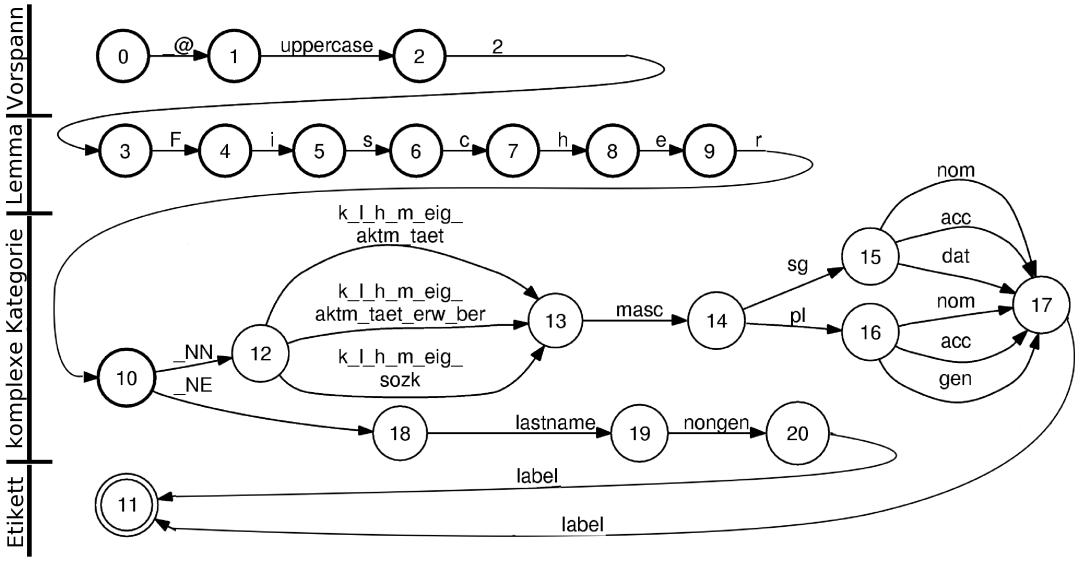

Abbildung 9: Wortanalyse für das Token *Fischer*

Hierbei gibt der Vorspann Auskunft über die Oberflächenform des entsprechenden Wortes (Zustand 1-2) und über die Homographieeigenschaften (Zustand 2-3). Die morphologische Analyse liefert das Lemma (Zustand 3-10) und die komplexen morphologischen Lesarten. Durch das Etikett wird einer Wortanalyse gegebenenfalls interne oder externe Evidenz oder internes Potenzial zugeschrieben. In der Eingabe sind die Wortanalysen mit dem Symbol [LABEL] zunächst als unspezifiziert markiert.

5.1.2 Regeln für Eigennamen

Bei der Eigennamenerkennung werden getrennte Grammatikspezifikationen für Personennamen, Organisationsnamen und geographische Namen (fortan kurz Geoname) definiert. Jede dieser Teilgrammatiken enthält Spezifikationen für interne und externe Evidenzen und für interne Potenziale. Da über semantische Merkmale und Homographieeigenschaften abstrahiert werden kann, sind für die Modellierung der Eigennamenerkennung nur wenige Regeln notwendig.

Innerhalb von Regeln für Eigennamen können unspezifizierte Etiketten mit dem Symbol [LABEL] umgeschrieben werden. Auf diese Weise wird bestimmten Wortanalysen Evidenz oder Potenzial für die verschiedenen Eigennamentypen (s. o.) mit Hilfe von *Evidenz-Symbolen* bzw. *Potenzial-Symbolen* zugeschrieben. Für den Eigennamentyp Geoname wäre dies beispielsweise [EVIDENCE_GEONAME] bzw. [POTENTIAL_GEONAME].

Folgende Beispielregel illustriert, wie die interne Evidenz Geonamen zugeschrieben wird, wenn sie nicht homograph sind:

```
<ne_rule name="internal_evidence">
  <category>[NE_Phrase Evidence=intern NE_Type=geoname]</category>
  <pattern>[@ Form=Uppercase Homographtype=1][MorphLetter]*
           [NE Nametype=geoname]([label]:[evidence_geoname])</pattern>
<\ne_rule>
```

Der reguläre Ausdruck innerhalb der Regel besteht aus den Teilen Vorspann ([@ FORM=UPPERCASE HOMOGRAPHTYPE=1]), Lemma ([MORPHLETTER]*), komplexe Kategorie ([NE NAMETYPE=GEONAME]) und Etikett ([LABEL]:[EVIDENCE_GEONAME]).

Im Beispiel werden somit Muster für Geonamen (NAMETYPE=GEONAME) beschrieben, die mit einem Großbuchstaben beginnen und weder zu anderen Kategorien noch zu einem anderen Eigennamentyp homograph sind. Dies wird durch den Wert 1 des Homographietyps ausgedrückt. Das Etikett [LABEL] wird in [EVIDENCE_GEONAME] umgeschrieben, was den entsprechenden Wortanalysen eine Evidenz für einen Geonamen zuschreibt. [NE_PHRASE EVIDENZ=INTERN NE_TYPE=GEONAME] kennzeichnet das Ergebnis dieser Regel als phrasale komplexe Kategorie. Aus ihr geht hervor, welche Evidenz vorliegt (EVIDENZ=INTERN) und von welchem Typ der entsprechende Eigenname ist (NE_TYPE=GEONAME). Die oben definierte Grammatikspezifikation trifft nicht für die Eingabe *Fischer* in Abb. 9 zu, da der Homographietyp 2 (dies steht für die Homographie zu einem Appellativ) nicht mit dem geforderten Homographietyp 1 (dies steht für ein nicht homographes Nomen) übereinstimmt.

Es ist auch möglich, Kategorien innerhalb der Eigennamenmuster umzuschreiben. So kann einem unbekannten Wort eine Eigennamenkategorie zugewiesen werden. Gleiches gilt für Wörter, die zwar in der Morphologie keine Analyse als Eigenname haben, jedoch durch genügend Evidenz zu einem Eigennamen umgeschrieben werden.

Im Fall von externer Evidenz können bei den Eigennamenregeln auch rechte und linke adjazente Kontexte angegeben werden:

```
<leftcontext>[@ Form=Uppercase][MorphLetter]*[NN SemClass=k_l_h_m_eig][label]
                          ([epsilon]:[MB_right])</leftcontext>
```

Im Beispiel wird ein Eigennamenkontext definiert, der den Menschenbezeichner [SEMCLASS=K_L_H_M_EIG] enthält - die Kodierung steht dabei für MENSCH NACH EIGENSCHAFTEN wie z.B. *Bürgermeister*, *Dichter* oder *Geizkragen*. Zusätzlich wird [EPSILON] durch [MB_RIGHT] ersetzt. Dies bedeutet, dass sich der Menschenbezeichner auf einen rechts von ihm stehenden Personennamen bezieht.

5.2 Der Grammatik-Compiler

Die einzelnen Grammatikspezifikationen für Personen-, Organisations- und Geonamen werden mit Hilfe des in Abschnitt 3.2 beschriebenen Verfahrens in die Klammertransduktoren $T_{personname}$, $T_{orgname}$ und $T_{geoname}$ überführt. Als Klammersymbole werden hierbei die komplexen phrasalen Kategoriesymbole der Eigennamenregeln verwendet. Innerhalb von Klammerungen werden durch einen Bewertungsautomaten ähnlich dem in Abb. 6 lediglich Etiketten gewichtet. Andernfalls würden Kategorien mit höherer morphologischer Zerlegungskomplexität zu längeren Pfaden im Automaten führen und somit allein durch ihr Materialgewicht 'gewinnen'. Bei der Bewertung werden die Etiketten innerhalb von Mustern für internes Potenzial statt mit dem in Abschnitt 3.2 definierten Gewicht mit einem *Strafgewicht* versehen. Dieses ist das inverse Gewicht zu dem, das 'normalerweise' innerhalb von Klammerungen zugewiesen wird. Das Klammern dieser Muster wird somit letztlich unterdrückt. So wird ausgedrückt, dass internes Potenzial nicht eindeutig auf einen Eigennamentyp schließen läßt.

Die einzelnen Klammertransduktoren werden zu einem Eigennamenklammerer T_{main} vereinigt:

$$T_{main} = T_{geoname} \cup T_{personname} \cup T_{orgname} \tag{13}$$

Hierdurch konkurrieren die einzelnen Eigennamentypen miteinander. Um bei identischen Klammerungsgewichten Präferenzen herzustellen, werden einerseits Klammerungen mit externer Evidenz mit Hilfe von Gewichtungen bevorzugt und andererseits werden den einzelnen Eigennamentypen durch Gewichte Präferenzen zugeordnet. Diese Gewichte sind so gewählt, dass sie die *Longest-Match*-Bedingung nicht beeinflussen. Hierzu ist ein Semiring-Produkt mit entsprechender Ordnung definiert, bei der diese Gewichtungen eine niedrigere Präferenz haben als die *Longest-Match*-Bedingung (siehe Abschnitt 3.3). In unserem System wird folgende Ordnung von Eigennamenklassen verwendet:

$$Personennamen \succ Geonamen \succ Organisationsnamen \tag{14}$$

5.3 Die Eigennamengrammatik

In der Folge wird eine einfache Eigennamengrammatik skizziert, die die Grundlage für die Evaluation des Systems in Abschnitt 6 bildet. Die Grammatik beschreibt zulässige Eigennamenskontexte auf der Basis von inneren Namenskontexten und Appositionskontexten. Innere Namenskontexte bei Personen sind Vornamen, Nachnamen und Namenszusätze (wie Anrede oder Titel); bei Geonamen sind dies diejenigen Wortformen des Lexikon, denen die semantischen Kategorien Land, Ort oder Gewässer zugeordnet wird, sowie Zusätze wie Platz oder Straße. Bei Organisationsnamen sind es die im Lexikon verzeichneten Firmen und Organisationsnamen

sowie Namenszusätze (wie z.B. GmbH oder Co.). Grundsätzlich kann zwischen einer engen, pränominalen Apposition und einer weiten, postnominalen Apposition unterschieden werden. In der jetzigen Grammatik wird nur die enge Apposition verwendet, da die weite Apposition in den untersuchten Beispieltexten die Vollständigkeit der Eigennamenerkennung nur geringfügig erhöht, dafür jedoch die Korrektheit der Grammatik deutlich verringert. Enge Appositionen sind bei Personen Funktions- oder Menschenbezeichner, wie *der Bürgermeister X* oder der *X-Freund*, bei Geonamen Bezeichner wie *die Stadt* oder *der See* und bei Organisationsnamen schließlich Nominalphrasen wie *das Unternehmen* oder *das Institut*. Als Eigenname wird eine Wortform bzw. eine Folge von Wortformen von der Grammatik nur dann bewertet, wenn sie über ausreichenden namensinternen bzw. appositiven Kontext verfügt. Ausreichend sind beispielsweise nicht-homographe Familiennamen oder Geonamen oder Organisationsnamen; die Homographie basiert hierbei auf den im Lexikon verzeichneten Homographieinformationen (vgl. hierzu Abschnitt 4). Ausreichend sind ferner Mehrwortkontexte, wenn sie beispielsweise die Sequenz *Vorname Nachname* enthalten und entweder *Vorname* oder *Nachname* im Lexikon als nicht homograph zu einem Nicht-Personennamen klassifiziert wird. Enthält der Namenskontext nur einen homographen Namen, ist zusätzlich ein appositiver Kontext notwendig, um von der Grammatik als Eigenname geklammert zu werden. Ausnahmen hiervon bilden nur koreferente Kontexte, auf die im folgenden Abschnitt eingegangen wird.

5.4 Auflösung der Koreferenz

Internen Potenzialen wird durch den Grammatik-Compiler innerhalb von Klammerungen ein Strafgewicht zugewiesen, so dass diese letztlich nicht markiert würden (vgl. Abschnitt 5.2). Von internem Potenzial kann aber auf eine Eigennamenklasse geschlossen werden, wenn die enthaltenen Wörter in einer Koreferenzbeziehung zu anderen Wörtern im Text stehen. In dem hier beschriebenen Verfahren ist Koreferenz durch gleiche Lemmaform zusammen mit gleichem Kategoriesymbol und gleicher Eigennamenklasse definiert. Interne Potenziale, die in solch einer Koreferenzbeziehung stehen, werden in interne Evidenz überführt.

Wir definieren hierzu eine Koreferenzauflösungsfunktion, die internes Potenzial auf interne Evidenz abbildet. Die Funktion bezieht sich dazu auf eine gewichtete reguläre Sprache L, die das Resultat der Komposition eines Eigennamenklammerers mit einer Eingabe darstellt (vgl. (6)). Nur die Wortanalysen aus L werden herangezogen, die Evidenz für eine Eigennamenklasse ausweisen. Für diese werden dann koreferente Potenziale innerhalb der Sprache L durch Gewichtungen gestützt.

Die Vorgehensweise soll anhand der folgenden Textpassage erläutert werden: *Der Verein trennt sich von Trainer Wolfgang Wolf(1). Wolf(2) ist bereits der elfte Trainer.* Nach der Analy-

se dieses Textes durch den Eigennamenklammerer enthält die Ergebnissprache L Wortanalysen mit Evidenz für einen Personennamen für *Wolfgang Wolf(1)*. Weiter enthält L für *Wolf(2)* eine Wortanalyse mit internem Potential für einen Personennamen. Im ersten Schritt der Koreferenzauflösung werden alle möglichen Wortanalysen, die zu in L enthaltenen Evidenzen koreferent sind, als reguläre Sprache abgelegt, wobei ihnen internes Potential zugeschrieben wird. Diese Sprache enthält in unserem Beispiel eine Wortanalyse mit internem Potential für *Wolf(1)* und wird im zweiten Schritt dazu verwendet, enthaltene interne Potentiale in L durch Gewichtungen zu stützen. Auf diese Weise kann der Wortanalyse bezüglich *Wolf(2)* durch eine Gewichtung nachträglich interne Evidenz für einen Personennamen zugeschrieben werden.

Für die weiteren Definitionen sind folgende reguläre Sprachen definiert, die Bestandteile von Wortanalysen aus L sind (vgl. Abschnitt 5.1.1): $L_{introduction}$ ist die Sprache aller möglichen Vorspänne; $L_{category} = L_{cat\_sym} \cdot L_{features}$ ist die Sprache der möglichen komplexen Kategorien, die wiederum in die Sprache der Kategoriesymbole und deren Merkmale unterteilt ist; L_{lemma} ist die Sprache aller möglichen Lemmaformen; $L_{label} = L_{potential} \cup L_{evidence}$ ist die Sprache der möglichen Etiketten, die aus der Sprache der möglichen Potentiale und der möglichen Evidenzen besteht.

Die Sprachen der Wortanalysen mit Potenzial oder mit Evidenz für eine Eigennamenklasse wie folgt definiert:

$$L_{w\_evidence} = L_{introduction} \cdot L_{lemma} \cdot L_{cat\_sym} \cdot L_{features} \cdot L_{evidence} \tag{15}$$

$$L_{w\_potential} = L_{introduction} \cdot L_{lemma} \cdot L_{cat\_sym} \cdot L_{features} \cdot L_{potential} \tag{16}$$

Die reguläre Sprache $L_w = L_{w\_evidence} \cup L_{w\_potential}$ enthält hierbei Zeichenketten der Form $(s_i \cdot s_l \cdot s_c \cdot s_f \cdot s_e)$ mit $s_i \in L_{introduction}, s_l \in L_{lemma}, s_c \in L_{cat\_sym}, s_f \in L_{features}$ und $s_e \in L_{label}$.

Evidenzen für eine Eigennamenklasse sollen nur Potenziale bezüglich der gleichen Eigennamenklasse stützen. Um dies zu formulieren, denotiert $\widetilde{s_e}$ die Eigennamenklasse von $s_e \in L_{label}$, also Personen-, Organisations- oder Geoname. Beim Koreferenzabgleich zweier Wortanalysen werden das Lemma, das Kategoriesymbol und die Eigennamenklasse auf Gleichheit geprüft. Koreferenz, denotiert durch $\bowtie$, ist demnach folgendermaßen definiert ($s_1, s_2 \in L_w$):

$$s_1 \bowtie s_2 = (s_{i1} \cdot s_{l1} \cdot s_{c1} \cdot s_{f1} \cdot s_{e1}) \bowtie (s_{i2} \cdot s_{l2} \cdot s_{c2} \cdot s_{f2} \cdot s_{e2}) \iff \begin{array}{r} s_{l1} = s_{l2}, \\ s_{c1} = s_{c2}, \\ \widetilde{s_{e1}} = \widetilde{s_{e2}} \end{array} \tag{17}$$

Anhand der Koreferenzdefinition kann die reguläre Sprache $coreference(L)$ definiert werden:

$$coreference(L) = \{s_{w\_potential} \in L_{w\_potential} | \text{es gibt ein } s_{w\_evidence} \in L_{w\_evidence} \text{ mit } s_{w\_potential} \bowtie s_{w\_evidence} \text{ , für das gilt } \Sigma^* \cdot \{s_{w\_evidence}\} \cdot \Sigma^* \subseteq L\} \tag{18}$$

Diese Sprache $coreference(L)$ enthält alle möglichen Wortanalysen, die mit internem Potential ausgezeichnet sind ($s_{w\_potential} \in L_{w\_potential}$) und koreferent zu Wortanalysen sind ($s_{w\_potential} \bowtie s_{w\_evidence}$), die Evidenz für eine Eigennamenklasse aufweisen ($s_{w\_evidence} \in L_{w\_evidence}$) und in L vorkommen ($\Sigma^* \cdot \{s_{w\_evidence}\} \cdot \Sigma^* \subseteq L$).

Um anhand der regulären Sprache $coreference(L)$ Potenziale in L zu gewichten und somit zu stützen, definieren wir die gewichtete reguläre Sprache $support\_potential(L)$:

$$support\_potential(L) = (\Sigma^* \cdot coreference(L) \cdot \langle \omega_{support} \rangle)^* \cdot \Sigma^* \tag{19}$$

$\omega_{support} \in W$ denotiert darin das Gewicht, mit dem die Potenziale gewichtet werden sollen; W ist der verwendete Gewichtungssemiring von L. Die Gewichtung erfolgt wieder optional und somit komplementierungsfrei.

Anhand der gewichteten regulären Sprache $support\_potential(L)$ können innerhalb von L zu interner oder externer Evidenz koreferente interne Potenziale gewichtet werden. Die Koreferenzauflösungsfunktion ϕ von einer gewichteten regulären Sprache in eine gewichtete reguläre Sprache, die dies realisiert, ist folgendermaßen definiert:

$$\phi(L) = L \cap support\_potential(L) \tag{20}$$

Das Gewicht $\omega_{support}$ ist hierbei so gewählt, dass es dem doppelten des Inversen des Strafgewichts aus Abschnitt 5.2 entspricht. Wenn demnach innerhalb des Grammatik-Compilers das Strafgewicht $\omega_{punish} \in W$ und 'normalerweise' innerhalb von Klammerungen das Gewicht $\omega_{normal} \in W$ vergeben wird, wobei $\omega_{normal}^{-1} = \omega_{punish}$, ist $\omega_{support} = \omega_{punish}^{-1} \otimes \omega_{punish}^{-1}$ und somit $\omega_{punish} \otimes \omega_{support} = \omega_{normal}$ und es gilt die *Longest-Match*-Bedingung, wie sie in Abschnitt 3.3 definiert werde.

Die $Apply(T_{rule}, x)$-Funktion aus (6) hat, erweitert um die Koreferenzauflösungsfunktion, nun die folgende Form:

$$Apply(T_{rule}, x) =_{def} Bestpath(\phi(Proj_2(ID(x) \circ T'_{rule}))) \tag{21}$$

5.5 Das Anwendungssystem

Im Anwendungssystem wird die Grammatik auf einen Eingabetext angewendet. Die Grundoperation ist dabei die in (21) definierte, modifizierte $Apply(T, x)$-Funktion. Der aufgrund der Vorverarbeitung des Eingabetextes entstehende Textautomat (vgl. Abbildung 9) wird mit dem gewichteten Transduktor, der die Grammatik repräsentiert, komponiert.[11] Einzelne Analysen werden dann durch die Koreferenzfunktion durch Gewichte gestützt. Unter allen dadurch entstehenden konkurrierenden Analysen, d.h. Klammerungen, wird dann die beste (oder eine beste) isoliert. Die Kriterien für eine beste Analyse sind hierbei die *Longest-Match*-Bedingung und die Klammerungspräferenzen. Obwohl dies scheinbar nach einem Algorithmus mit hoher Komplexität verlangt – immerhin können sich im ungünstigsten Fall exponentiell viele Analysen zur Länge des Eingabetexts ergeben – ist die Komplexität, wie oben gezeigt, doch in $O(|x|)$.

Das Anwendungssystem besteht aus folgenden Einzelschritten: In der Vorverarbeitung wird die Eingabe in die einzelnen Tokens zerlegt und nach Groß- und Kleinschreibung bzw. Satzzeichen unterschieden. In der zweiten Phase werden mehrere Tokens auf der Grundlage eines Mehrwortlexikons zu Mehrwortausdrücken zusammengefasst. Diese Phase, in der auch Abkürzungen und Satzenden erkannt werden, basiert auf einer zweistufigen Vorverarbeitungsarchitektur, deren untere Ebene ein effizienter, in C implementierter lexikalischer Scanner bildet. Diejenigen Tokens, die als potenzielle Wörter vorklassifiziert wurden, werden mit Hilfe der TAGH-Morphologie analysiert. Falls ein Token nicht erkannt werden konnte, wird ihm die Kategorie [UNKNOWN] zugeordnet. Anschließend werden aus den morphologischen Analysen die Homographietypen dynamisch berechnet. Das Ergebnis ist ein endlicher Automat, wie er in Abbildung (9) dargestellt ist. Die einzelnen Wortanalysen werden inkrementell mit dem Eigennamenmarkierer komponiert. Aus dieser Komposition entsteht ein Automat, der alle möglichen Analysen bezüglich der Markierung und Klassifizierung von Eigennamen enthält. Diese Analysen verursachen hierbei verschiedene Kosten. Anhand der Koreferenzauflösungsfunktion werden nun Gewichtungen vorgenommen. So wird das Umwandeln von internem Potenzial zu interner Evidenz realisiert. Danach wird mit Hilfe des *Bestpath*-Algorithmus für azyklische Automaten die beste Analyse ermittelt; diese bildet das Ergebnis der Eigennamenerkennung. Abschließend erfolgt die Formatierung des Ergebnisses (z.B. im XML-Format).

[11] Tatsächlich wird dieser Textautomat aus Effizienzgründen niemals konstruiert, sondern stattdessen ein inkrementeller Kompositionsalgorithmus verwendet.

6 Evaluation

Die Qualität des Eigennamenerkenners SynCoP haben wir anhand eines eigenen Korpus von 100 Zeitungsartikeln deutscher Tages- und Wochenzeitungen (Berliner Zeitung, Bild, FAZ, Leipziger Volkszeitung, Stern, Super-Illu, SZ, Tagesspiegel, TAZ, Welt, NEWS) im Bereich Politik evaluiert. In diesem 64.372 Tokens umfassenden Korpus haben wir per Hand 1214 Personennamenkontexte, 845 Organisationsnamenkontexte und 1009 Ortsnamenkontexte annotiert.

	Personennamen	Organisationsnamen	Ortsnamen
Vollständigkeit	92,83%	82,49%	80,87%
Genauigkeit	93,45%	94,96%	95,44%

Tabelle 1: Vollständigkeit und Genauigkeit

Vollständigkeit und Genauigkeit des SynCoP-Systems sind in Tabelle 1 aufgeführt. Aus der Tabelle 1 geht hervor, dass SynCoP alle Eigennamentypen mit etwa der gleichen Genauigkeit erkennt. Bei der Vollständigkeit fallen die Unterschiede größer aus: sie liegen bei den Personennamen mit knapp 93% etwa 10 Prozentpunkte über den Organisations- bzw. Ortsnamen. In Tabelle 2 ist aufgeführt, wie stark namensinterne, appositive Evidenz und die Koreferenzauflösung zur Vollständigkeit beitragen.

	Personennamen	Organisationsnamen	Ortsnamen
a) namensinterne Evidenz	49,60%	91,82%	58,82%
b) appositive Evidenz	19,67%	4,89%	26,68%
a) oder b)	6,32%	0,85%	3,59%
Koreferenz	24,40%	2,44%	10,91%

Tabelle 2: Beitrag der verschiedenen Faktoren zur Vollständigkeit

Tabelle 2 enthält den Prozentsatz, zu dem der jeweilige Evidenztyp (1. Spalte) zur Eigennamenerkennung beiträgt. Die Beitrag der internen Evidenz wird demnach durch die Addition der ersten und dritten Zeile ermittelt – die Zeile ”a oder b” steht hierbei für Kontexte, in denen sowohl namensinterne als auch appositive Evidenz für eine Erkennung ausreichen. Der Beitrag der appositiven Evidenz berechnet sich durch die Addition der zweiten und dritten Zeile. Dementsprechend erhält man, dass in 55,92% (62,41%) aller vom System erkannten Personennamen (Ortsnamen) die namensinterne Evidenz für eine Erkennung ausreichend ist. Bei Organisationsnamen lassen sich mit der namensinternen Evidenz sogar 92,67% erkennen. Die Hinzunahme von appositiver Evidenz bringt bei den Personennamen (Ortsnamen) bei 19,67% (26,68%) aller Fälle eine höhere Vollständigkeit. Bei Organisationsnamen sind dies lediglich 4,89%. Diese deutlich unterschiedlichen Anteile bei der appositiven und namensinternen Evidenz sind darauf

zurückzuführen, dass Personennamen und Geonamen deutlich häufiger durch Namensbezeichner eingeführt werden als Organisationsnamen.

Darüber hinaus geht aus der Tabelle hervor, dass die Anzahl der Fälle relativ klein ist, in denen sowohl namensinterne als auch appositive Evidenz für die Erkennung vorliegt (Zeile "a oder b"). Eigennamen werden somit überwiegend nur dann mit externer Evidenz versehen, wenn der Eigenname als solcher homograph ist oder keine namensinterne Evidenz vorliegt. Schließlich zeigt die Tabelle, dass nahezu ein Viertel aller Personennamen (24,4%) nur durch die Koreferenz zu einem durch ausreichende namensinterne bzw. appositive Evidenz im gleichen Text belegten Personennamen erkannt werden können. Dabei muss der als sicher erkannte Personenname nicht notwendig vorher im Text stehen; es reicht, wenn dieser im selben Artikel verzeichnet ist. Der Beitrag der Koreferenz ist bei der Erkennung von Geonamen mit knapp 11% bzw. bei Organisationsnamen mit gerade einmal 2,44% wesentlich schwächer. Letzteres ist darauf zurückzuführen, dass die Variation der Benennung von Organisationsnamen geringer ist als bei Personennamen, die in der Regel mit Vor- und Nachnamen eingeführt, im Text dann aber nur noch mit dem Nachnamen referenziert werden.

Ein Teil der hohen Erkennungsrate lässt sich durch die hohe Abdeckung der Namen der TAGH-Morphologie im Bereich Politik erklären. Darüber hinaus werden aber auch etliche dem System unbekannte Namen dadurch erkannt, dass Funktions- und Berufsbezeichnungen in deren Kontext richtig identifiziert werden können. Eine wichtige Rolle spielt hierbei das Zusammenspiel der semantischen Lexika und die Kompositazerlegung der TAGH-Morphologie. Hierdurch können beispielsweise nicht im Lexikon als Ganzes enthaltene Komposita auf bekannte Bezeichnungen zurückgeführt werden. Beispiele hierfür sind die im Korpus enthaltenen *CSU-Landesgruppenvorsitzende*, das *Polizeipräsidium* oder die *Baptistenkirche*, die als Determinativkomposita behandelt werden und damit semantisch auf die im Lexikon klassifizierten Nomen *Vorsitzende*, *Präsidium* und *Kirche* zurückgeführt werden.

Die Behandlung der Homographie ist Teil des Verfahrens. Beispielsweise können damit Kontexte wie *Der SODI-Vorsitzende Carl Ordnung* vollständig erkannt werden, da *Ordnung* im TAGH-Morphologiesystem als Nomen und als potenzieller Nachname kodiert ist. Desweiteren spielt die Ausnutzung verschiedener Homographietypen eine Rolle. Im Kontext *Wedgwood-Baptistenkirche* ist *Wedgwood* ein unbekanntes Nomen, das Kompositum *Baptistenkirche* erbt die semantischen Klassen von Kirche und entspricht somit entweder einem `Gebäude`, einer `sozialen Gruppe` oder der Bedeutung `Ort`. Somit führt ein Pfad zu der einzigen gültigen Sequenz UNBEKANNTES NOMEN BINDESTRICH ORT. In diesem Fall wird somit korrekterweise die Ortslesart gewählt.

Die Auflösung der Binnenhomographie, also der Homographie zwischen verschiedenen Eigennamentypen wird über den *Longest match* und die in Abschnitt 5.2 beschriebene Abfol-

gepräferenz realisiert. Dies geschieht in einem konkurrierenden Verfahren der verschiedenen Eigennamengrammatiken. Beispielsweise wird *Unternehmen Beiersdorf* aufgrund der externen Evidenz *Unternehmen* dem Organisationsnamen zugeordnet und erhält ein höheres Gewicht als der Familienname *Beiersdorf*. Darüber hinaus erhöht die konkurrierende Anwendung der Teilgrammatiken die Korrektheit des Verfahrens: im vorliegenden Beispiel wird *Beiersdorf* nur als Organisations-, nicht jedoch als Personenname klassifiziert.

Nicht vollständig erkannt werden Eigennamen vornehmlich aufgrund von ambigen Namen ohne hinreichenden Kontext. Beispielsweise ist im Zeitungskorpus bei Geonamen wie *Polen* oder *Sachsen* keine externe Evidenz vorhanden, die es ermöglicht, die Lesart *Land* von der Einwohnerlesart zu trennen. Ebenso wird der Geoname *Autobahn A1* nicht erkannt, da Sonderformen von der Grammatik noch nicht behandelt werden. Bei den nicht erkannten Organisationsnamen fehlt meist die externe Evidenz. Beispielsweise ist dies in Texten der Fall, in denen Unternehmensnamen wie *Siemens* oder *Beiersdorf* nicht eingeführt werden und somit die Ambiguitäten nicht aufgelöst werden können. In einigen Fällen ist der Kontext für eine Disambiguierung mit der Grammatik nicht ausreichend. Dies ist beispielsweise bei homographen Personennamen der Fall, die im Text nicht mit externer Evidenz eingeführt werden. In diesem Fall können kontextlose homographe Eigennamen nicht durch die Koreferenzauflösung markiert werden. Da der Grammatik nur die morphologische Analyse, aber kein POS-Tagger vorgeschaltet ist, spielen auch kategorieübergreifende Ambiguitäten eine Rolle.

Der Einsatz sehr großer Listen führt teilweise zu einer Übergenerierung. Ein Beispiel hierfür ist die Sequenz *Hand Ordnung*, welche fälschlicherweise als Personenname erkannt werden, da *Hand* bzw. *Ordnung* neben ihrer Interpretation als Appellativum von der TAGH-Morphologie auch als – englischer – Vorname bzw. potenzieller *Ordnung* Nachname klassifiziert werden. Hier bräuchte man zusätzlich syntaktisches Wissen, um zu erkennen, dass der Kontext *mit eiserner Hand Ordnung schaffen* aus zwei Phrasen besteht und somit *Hand Ordnung* kein Eigenname sein kann. In dem hier verwendeten Formalismus mit gewichteten Transduktoren könnte man könnte dies durch eine "Strafgewicht" realisiert werden, welches potenzielle Eigennamenkontexte dann erhalten, wenn sie phrasenübergreifend sind.

Weitere Ursachen für falsch erkannte Namen liegen in der Unvollständigkeit der Morphologie. So wird beispielsweise in der Passage *das A und O. Hermann Löhr* das Nomen *A und O* nicht erkannt und somit *O.* fälschlicherweise dem Personennamen zugeordnet.

7 Zusammenfassung

Vorgestellt wurde ein Verfahren, welches die Eigennamenerkennung zwischen morphologischer Analyse und Part-of-Speech Tagging ansiedelt und alle wesentlichen Komponenten der regelba-

sierten Eigennamenerkennung mit gewichteten endlichen Transduktoren formuliert. Es wurde gezeigt, dass dieses Verfahren sehr effizient ist und mit sehr großen lexikalischen Ressourcen umgehen kann. Vollständigkeit und Genauigkeit des vorgestellten SynCop-Anwendungssystems liegen deutlich über den maschinellen Lernverfahren und sind mit denen anderer regelbasierter Ansätze vergleichbar (Neumann and Piskorski (2002); Volk and Clematide (2001)). Die künftige Arbeit wird darin bestehen, die Genauigkeit des Systems weiter zu erhöhen. Dies lässt sich durch die Einbeziehung der syntaktischen Komponente von SynCop in die Eigennamenerkennung erreichen, durch die es beispielsweise möglich wäre, Eigennamen nicht zu markieren, wenn sie phrasenübergreifend sind.

Literatur

S. Abney. Partial parsing via finite-state cascades. *In Proceedings of the ESSLLI '96 Robust Parsing Workshop*, 1996.

J. Didakowski. *Robustes Parsing und Disambiguierung mit gewichteten Transduktoren*. Linguistik in Potsdam, Bd. 23, 2005.

A. Geyken and T. Hanneforth. TAGH: A Complete Morphology for German based on Weighted Finite State Automata. In A. Yli-Jyrä, L. Karttunen, and J. Karhumäki, editors, *Finite State Methods and Natural Language Processing*, pages 55–66. Springer, Berlin, Heidelberg, 2006.

A. Geyken and N. Schrader. Lexikonet - a lexical database based on type and role hierarchies. *In Proceedings of LREC-2006*, 2006.

T. Hanneforth. Longest-match recognition with weighted automata. *In Proceedings of FSMNLP 2005, Lecture Notes in Artificial Intelligence*, 2006.

J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

L. Karttunen. Directed replacement. *In Proc. of the 34rd Annual Meeting of the ACL*, 1996.

L. Karttunen. The replace operator. In *Meeting of the Association for Computational Linguistics*, pages 16–23, 1995.

W. Kuich and A. Salomaa. *Semirings, Automata, Languages*. Springer, 1986.

E.L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart & Winston, 1976.

D. McDonald. Internal and external evidence in the identification and semantic categorization of proper names. In *Corpus Processing for lexical Acquisition*, pages 21–39. MIT-Press, 1996.

A. Mikheev, C. Grover, and M. Moens. Description of the LTG system used for MUC-7. *Seventh Message Understanding Conference (MUC-7)*, 1998.

A. Mikheev, M. Moens, and C. Grover. Named entity recognition without gazetteers. *In Proceedings of EACL*, 1999.

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

G. Neumann and J. Piskorski. A shallow text processing core engine. Technical Report DFKI, 2002.

U. Quasthoff and C. Biemann. Named entity learning and verification: EM in large corpora. *In Proceedings of CoNLL-2002*, pages 8–14, 2002.

M. Rössler. Corpus-based learning of lexical resources for german neamed entity recognition. *In Proceedings of LREC-2004*, 2004.

M. Stevenson and R. Gaizauskas. Using corpus-derived name lists for named entity recognition. *In Proceedings of ANLP*, 2000.

M. Volk and S. Clematide. Learn - filter - apply - forget. mixed approaches to named entity recognition. *In Proceedings of 6th International Workshop on Applications of Natural Language for Information Systems.*, 2001.