Eigennamenerkennung mit großen lexikalischen Ressourcen

Jörg Didakowski
BBAW
Jägerstr. 22/23
10117 Berlin
didakowski@bbaw.de

Alexander Geyken
BBAW
Jägerstr. 22/23
10117 Berlin
geyken@bbaw.de

Thomas Hanneforth
Universität Potsdam
Am Neuen Palais 10
14415 Potsdam
tom@ling.uni-potsdam.de

1 Einleitung

Nicht zuletzt durch die Förderung im Rahmen der MUC-Konferenzen[1] (MUC, 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 bzw. 95% Korrektheit (z.B. (Mikheev et al., 1998), (Mikheev et al., 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 dadurch erschwert, dass das Deutsche über eine freiere Wortstellung verfügt und 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 Sicherer Vorname gefolgt von unbekanntem grossgeschriebenen Wort = Personenname. Unter der Annahme, dass Fremdwörter wie *Mountains* oder Komposita wie *Hundesalon* dem Lexikon unbekannt wären, würde man somit Sequenzen wie *Rocky Mountains* (hier ist *Rocky* Vorname) oder *Haralds Hundesalon* fälschlicherweise als Eigennamen identifizieren.

Für das Deutsche liegen einerseits ressourcenarme Systeme vor, bei denen Eigennamenkontexte mit maschinellen Lernverfahren gelernt werden ((Quasthoff and Biemann, 2002), (Rössler, 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)). In Ermangelung eines annotierten Testkorpus werden die Systeme anhand verschiedener Testsätze und kleinerer Korpora ausgewertet. Keines der genannten Systeme weist jedoch eine vergleichbar hohe Erkennungsrate auf wie die oben aufgeführten Systeme für das Englische. So erreicht Quasthoff bei seinem 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 werden 78% erkannt, die Korrektheit hingegen liegt auch nur bei 71%. Besser sieht dies bei den regelbasierten Systemen aus. Volk (Volk and Clematide, 2001) gibt bei der Evaluation von 990 Sätzen aus dem Computer-Zeitung [Konradin-Verlag 1998] eine Erkennung von 86% und eine Korrektheit in 92% aller Fälle an. Ähnlich verhält sich das System von Neumann (Neumann and Piskorski, 2002), welches auf einer Grundlage von 20.000 tokens der Wirtschaftswoche evaluiert wurde. Hier lagen Vollständigkeit und Korrektheit bei 81% bzw. 96%. Bei allen genannten Systemen liegt die Erkennung von Organisationsnamen und geographischen Namen, sofern sie diese durchführen, schlechter.

Bei dem hier vorgestellten Eigennamenerkenner handelt es sich ebenfalls um ein regelbasiertes System. Dieses beruht jedoch auf umfangreicheren Ressourcen als die beiden oben genannten regelbasierten Systeme. Dies sind insbesondere eine vollständige Morphologie des Deutschen, die sowohl Derivations- wie auch Kompositionsregeln integriert und somit unbekannte Wörter analysieren kann, sowie einer einer Kontexterkennung mit Hilfe eines 90.000 nach Nomen umfassenden lexikalischen Ontologie, die mehr als 60.000 Menschenbezeichner (z.B. *Politiker*, *Ruderer*, *Tycoon* etc. umfasst). Realisiert wurde der Eigennamenerkenner mit dem regelbasierten System SynCoP (Syntactic Constraint Parser), einem auf Finite-State-Techniken beruhenden *Shallow parser* ((Didakowski, 2005), (Hanneforth, 2005a)). SynCoP basiert auf der TAGH-Morphologie, einer vollständigen Morphologie des Deutschen ((Geyken and Hanneforth, 2005), sowie für die Eigennamenerkennung auf sehr umfangreichen Listen von Personen- und Organisationsbezeichnern (Geyken and Schrader, 2006)). Im folgenden werden zunächst die Grundideen des Systems skizziert (Abschnitt 2) und die verwendeten Ressourcen beschrieben (Ab-

[1] Message Understanding Competition

schnitt 3); in Abschnitt 4 wird das System SynCoP und die Anwendung des Systems für die Eigennamenerkennung beschrieben. Schließlich erfolgt in Abschnitt 5 eine Kurzevaluation der Ergebnisse.

2 Ziele und Grundideen des Eigennamenerkenners

Ziel des hier beschriebenen Systems ist die sichere Erkennung von Eigennamen in neueren nichtfachsprachlichen Zeitungstexten in einem möglichst ausreichenden Kontext auf der Basis sehr umfangreicher lexikalischer Ressourcen. Aufgrund der Homographie von Eigennamen und Appellativa unterscheidet das System sichere und unsichere Eigennamenkontexte. Im folgenden soll dies anhand der Eigennamenkategorie Personenname illustriert werden. Wir unterscheiden drei Fälle von Nachnamen in Texten: Nachnamen in Texten können a) dem System als Nachname bekannt und nicht homograph sein, oder b) dem System bekannt, aber homograph zu einem Appellativum oder einer anderen Eigennamenkategorie sein, oder schließlich c) einem für das System unbekannten token entsprechen. Die Grundidee des Systems beruht darauf, dass die dem System bekannten Namen in Zeitungstexten in der Regel nicht mehr in derselben Weise eingeführt werden wie die dem System unbekannten Namen. Mit anderen Worten sind Personennamenkontexte in Fall a) eher kleiner als in Fall c), bei dem der Name zumindest einmal im Artikel oder zumindest in der Tagesausgabe der Zeitung durch eine Funktions- oder sonstige Menschenbezeichnung eingeführt wird. Das System sollte in Fall c) das token nur dann als Personennamen klassifizieren, wenn es von einem ausreichenden Kontext umgeben ist, der die Erkennung des tokens als Personenname sicher macht. Sichere Personennamenkontexte sind entweder Appositionen, in denen eine Funktionsbezeichnung (z.B. Politikerin, Abteilungsleiter) oder eine anderweitige die Person charakterisierende Menschenbezeichnung (*Schlafmütze*, *Tycoon*, *Blutsbruder*) enthalten ist, oder aber 'namensinterne' Informationen wie Vornamen und Titel. Da das Verfahren (s. Abschnitt 3) eine gewichtete longest-match Strategie nutzt, wird der längste Kontext ausgewählt, der durch die lokale Eigennamengrammatik spezifiziert ist. Im Falle b) eines homographen Nachnamens, d.h. ein für das System bekannter Nachname, welcher graphematisch entweder einem Vornamen, einem geographischen Namen oder einem der Morphologie bekannten Appellativum (Simplizium oder Kompositum) entspricht, müssen die Namenskontexte ebenfalls größer gewählt werden; sind diese nicht gegeben, wird der homographe Name zwar als Eigenname markiert, aufgrund des geringen Kontexts jedoch mit einem niedrigeren Gewicht versehen.

Der Ansatz ähnelt in Teilen den bei (Mikheev et al., 1999) beschriebenen „sure-fire rules“. Die Besonderheit dieses Systems ist dabei, daß die Morphologiekomponente einen hohen Vollständigkeitsgrad aufweisen muß, da es ansonsten im Unterschied zum Englischen zu einer zu hohen Überschneidung von unbekannten großgeschriebenen Wörtern und nicht erkannten Substantiven insbesondere von Komposita kommt. Dies wird durch die TAGH-Morphologiekomponente gewährleistet, welche im folgenden Abschnitt beschrieben wird.

3 Ressourcen

3.1 TAGH-Morphologie

Für die Eigennamenerkennung wurde das TAGH-Morphologiesystem (Geyken and Hanneforth, 2005) sowie ein Nomenthesaurus (LexikoNet, (Geyken and Schrader, 2006)) verwendet.

Das TAGH-Morphologiesystem lemmatisiert und zerlegt Wortformen auf der Grundlage gewichteter endlicher Transduktoren. Bei gewichteten Transduktoren können Endzustände und Übergänge mit Elementen aus einer Menge von Gewichten versehen sein, die bezüglich einer abstrakten algebraischen Struktur, eines Semirings, interpretiert werden. Diese abstrakte Struktur kann mit unterschiedlichen konkreten Operationen instantiiert werden, bei einem *probabilistischen Semiring* erhält man probabilistische Automaten, bei einem sog. *tropischen Semiring* Automaten, die das Auffinden kürzester Pfade effizient unterstützen[2].

Die Transduktoren sind auf der Basis der *Potsdam Finite State Maschine Library* realisiert (Hanneforth, 2005b). Diese in C++ geschriebene Bibliothek implementiert etwa 40 Operationen der Automatenalgebra in effizienter Weise und erlaubt zudem eine kompakte Speicherung in verschiedenen Repräsentationsformaten. Der TAGH-Morphologietransduktor weist ca. 4 Mio Zustände und 7 Mio Übergänge auf und belegt als Datei ca. 32 MB Festplattenspeicher. Die Verarbeitungsgeschwindigkeit liegt - je nach Rechnerleistung - zwischen 10.000 und 30.000 Wörtern pro Sekunde.

Die Erkennungsrate des TAGH-Systems bei neueren Zeitungstexten (z.B. Die ZEIT, Spiegel) liegt zwischen 98,5% und 99,5%.

Ausgangspunkt der TAGH-Morphologie sind eine Reihe von Morphem- und Wortformenlexika, die mittels verschiedener Compiler in endliche gewichtete Transduktoren übersetzt und dann durch einige hundert algebraische Operationen in den endgültigen Morphologietransduktor überführt werden. Die wichtigsten Teillexika sind die folgenden:

[2]Im tropischen Semiring werden Gewichte entlang eines Pfades addiert, Gewichte verschiedener Pfade, die die gleiche Zeichenkette akzeptieren, werden per Minimumsoperation verknüpft.

```
<token id="tid78" normalized="false">
  <text>Gewerkschaftsboss</text>
  <analysis id="aid78.1" pos="NN">
    <NN SemClass="k_l_h_m_eig_aktm_taet"
        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="12">Gewerkschaft/N\s#Boss</lemma>
  </analysis>
  <analysis id="aid78.3" pos="NN">
    <NN SemClass="k_l_h_m_eig_aktm_taet"
        Gender="masc" Number="sg" Case="nom_acc_dat"/>
    <lemma weight="22">Gewerk/N#Schaft/N\s#Boss</lemma>
  </analysis>
  <analysis id="aid78.4" 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 1: Analysen für *Gewerkschaftsboss* im XML-Format

Nomenlexikon: 88.000 einfache und komplexe Stämme mit Informationen zur Flexions- und Wortbildung.

Eigennamen: 160.000 geographische Eigennamen, 65.000 Vornamen, 240.000 Familiennamen

Verblexikon: 33.000 Lemmata

Adjektive: 18.000 Lemmata

Adverbien: 2.000 Wortformen

Geschlossene Formen: ca. 1.500 Präpositionen, Determinativa, Konjunktionen, Zahlwörter, Interjektionen.

Konfixe: 105 Konfixe

Abkürzungen und Akronyme: 9.000 (11.500) Einträge.

Nomenthesaurus: 60.000 klassifizierte Nomen in einer Nomenhierarchie.

Die Ausgabe der TAGH-Morphologie ist pro Wort ein gewichteter endlicher Automat, der die dem Wort zugeordneten Analysen in kompakter Form repräsentiert. Durch einen eigenen Formalismus können diese Analysen in beliebige Ausgabeformate gebracht werden. Abbildung 1 zeigt die XML-Ausgabe für das Wort *Gewerkschaftsboss*.

Wie in Abbildung 1 beispielhaft ersichtlich, kann ein Wort auch in linguistisch nicht motivierter Weise segmentiert werden. Das jeder Analyse zugeordnete Gewicht erlaubt es jedoch, diejenige/n mit dem/den geringsten Gewicht/en zu selegieren. Im Beispiel *Gewerkschaftsboss* sind das die Analysen aid78.1 und aid78.2. Nomen (mit dem STTS-Tag NN markiert) wird daneben noch eine semantische Klasse zugeordnet; *SemClass=k_l_h_m_eig_aktm_taet* bedeutet beispielsweise etwa *aktiver Mensch nach Tätigkeit*. Die im nächsten Abschnitt beschriebenen Grammatiken zur Eigennamenerkennung nehmen auf diese Merkmale Bezug.

3.2 Nomenhierarchie

Wichtige Personen, - außer wenn sie täglich in den Medien sind - werden in Zeitungsartikeln zumindest einmal im Artikel in einer Funktion, relationalen Zuordnung zu anderen Personen oder einer sozialen Stellung erwähnt wird. 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 Menschenbezeichnern zur Verfügung. Es ist somit möglich, Menschen mit politischen Berufen (z.B. *Bundesfinanzminister*) von künstlerischen Tätigkeiten (*Orchestermusiker*) zu unterscheiden, Menschen nach ihrer relationalen Zuordnung (*Nachkomme*, *Freund*) oder in ihrer Stellung (*Gewerkschaftsboss* zu erkennen. Diese Funktionen stehen in aller Regel in einem lokalen Kontext der Person. Hinzu kommen Listen von Institutionen, Firmen, geographische Nomen bzw. Adjektivableitungen. Aufgrund der Verknüpfung dieser Nomen mit der TAGH-Morphologie können auch Komposita mit Menschenbezeichnern erkannt werden.

4 Einbettung der Eigennamenerkennung in SynCoP

4.1 Systemüberblick

Der Eigennamenerkenner basiert auf dem regelbasierten Parser SynCoP, der eine schnelle und robuste Verarbeitung von Texten ermöglicht (Didakowski, 2005). SynCop basiert - ebenso wie die morphologische Analyse TAGH - auf der *Potsdam Finite State Library* (Hanneforth, 2005b). SynCoP, das für das Chunking, das syntaktische Tagging und die Analyse von Konstituentensatzstrukturen entwickelt wurde, verwendet hauptsächlich Finite-State-Techniken und besteht aus zwei Hauptkomponenten: dem Grammatikcompiler und dem eigentlichen Analysesystem. Für die Eigennamenerkennung wurde SynCoP so adaptiert, dass Eigennamen wie Chunks behandelt werden.

Eingabe von SynCop ist Fließtext, Ausgabe ist ein HTML-Text, in dem Eigennamenkontexte und -typen markiert sind. Zudem werden in den HTML-Text Verweise auf die verschiedenen morphologischen Analysen, zu den Regeln, die zu der jeweiligen Markierung führten, sowie zu bestimmten Eigenschaften der Markierungen selbst angelegt.

Bei der Analyse wird zwischen sicheren und unsicheren Eigennamen unterschieden. Als sicher gelten Eigennamen dann, wenn sie nicht homograph zu anderen Wortarten sind oder durch einen genügend großen Kontext eindeutig sind (s nächster Abschnitt). Unsichere Eigennamen werden mit einem

schwachen Gewicht markiert; dieses kann jedoch erhöht werden wenn die unsicheren Eigennamen von einem sicheren Eigennamen gestützt werden. Die Regeln für sichere und unsichere Eigennamen können in einem Grammatikformalismus angegeben werden. Dieser stellt eine Erweiterung der ursprünglichen Funktionalität von SynCop dar.

4.2 Grammatikcompiler

Durch den Grammatikcompiler innerhalb von SynCoP können durch entsprechende rationale und kombinatorische Operationen sowie Äquivalenztransformationen gewichtete Transduktoren kompiliert werden, mit deren Hilfe Eigennamen und Eigennamenkontexte *optional* markiert und gewichtet werden können. Dieses Markieren wird durch das Einfügen von Klammerungen realisiert. Anders als beim obligatorischen Einfügen von Klammern ist bei der optionalen Variante keine Komplementierungsoperation nötig. Die Komplementierung einer regulären Sprache (die beispielsweise die gesuchten Muster oder Kontexte beschreibt) wird bei robusten Verfahren dazu verwendet, diejenigen Teile der Eingabe, die von der Mustermenge *nicht* beschrieben werden, zu überlesen. Die Komplementierung ist allerdings eine sehr aufwändige Operation von exponentieller Komplexität, da die zu komplementierenden Automaten zuvor determinisiert werden müssen. Da Komplementautomaten definitionsgemäß eine totale Übergangsfunktion δ besitzen, kommt eine Sensitivität gegenüber großen Alphabeten hinzu, so dass die Erzeugung robuster Markierer für gegebene Suchmuster (vgl. z.B. das in (Karttunen, 1996) beschriebene Verfahren) schon bei geringer Grammatikkomplexität intraktabel sein kann (Hanneforth, 2005a).

Klammerungen können im allgemeinen Fall verschiedene Extensionen besitzen, da sich die Elemente der Suchmustermenge überlappen können, d.h. in Suffix- bzw. Präfix-Beziehungen zueinander stehen. Hinzu kommt, dass die morphologische Analyse im Fall von Segmentierungsalternativen weitere Ambiguitäten hinzufügt.

Aus den verschiedenen Analysen wird die präferierte Analyse mittels einer *Besten-Pfad*-Strategie ermittelt, die über einem tropischen Semiring formuliert ist. Hierzu werden die verschiedenen Analysen über eine Bewertungsfunktion mit reellwertigen Gewichten versehen, die eine longest-match Präferierung ausdrückt. Die Bewertungsfunktion ist als gewichteter Transduktor in die Eigennamenkontextgrammatik hineinkompiliert ([Did05]). Diese Vorgehensweise wird durch das oben erwähnte komplementierungsfreie Konstruktionsverfahren des Eigennamenkontextmarkieres ermöglicht, da gewichtete reguläre Sprachen nicht unter der Komplementierung abgeschlossen sind.

Regeln, wie sie beispielsweise für Eigennamenkontexte, werden in SynCoP in einer XML-Struktur als reguläre Ausdrücke notiert. Da das Verfahren komplementierungsfrei ist, können mit den Regeln zugleich auch Gewichte definiert werden. Die offline-Übersetzung der Grammatik in den Eigennamenmarkierer *MarkupNE* führt zu kompakten Transduktoren: der Automat für die von uns verwendete Grammatik für Personennamen weist 2.057 Zustände und 104.633 Übergänge auf.

Für die Eigennamenerkennung werden grundsätzlich zwei Arten von Grammatikregeln unterschieden: Regeln für sichere und für unsichere Eigennamen. Sichere Eigennamen sind nach einer Analyse immer sichtbar, unsichere nicht.

Sichere Eigennamen:

- Sichere Eigennamen sind z.B. ein oder mehrere aufeinander folgende Wörter mit der Kategorie NE, die nicht homograph sind oder zwei oder mehrere aufeinander folgende Wörter der Kategorie NE, wobei kein Wort homograph zu einem Funktionswort sein darf (bei Vernachlässigung der Groß- und Kleinschreibung).
- Sichere Eigennamen sind z.B. ein oder mehrere aufeinander folgende Wörter der Kategorie NE, denen ein passender semantischer Kontext voraus geht oder folgt. Bei einem entsprechendem semantischen Kontext können auch unbekannte Wörter als sichere Eigennamen angenommen werden. Hierbei können unbekannte Wörter auf die Kategorie NE umgeschrieben werden.
- Ein nicht absolut sicherer semantischer Kontext und eine nicht absolut sichere Abfolge für einen Eigennamen können einen sicheren Eigennamen bilden.

Unsichere Eigennamen:

- Unsichere Eigennamen sind Abfolgen von homographen Eigennamen und/oder unbekannten Wörtern in beliebiger Reihenfolge und Anzahl.
- Eigennamen können auch innerhalb von Kontexten unsicher bleiben. Ein Beispiel hierfür sind unbekannte Wörter oder Wörter der Kategorie NE innerhalb von Wortzusammensetzungen, die durch einen Bindestrich getrennt sind, und deren Kopf semantisch einen Eigennamen spezifiziert.

In der Grammatikspezifikation von SynCoP können sogenannte Trigger definiert werden. Ein Trigger ist eine als sicher markierte Kontextregel, die die kategorielle Zuordnung bestimmter Wörter innerhalb ihres Gültigkeitsbereichs ändern kann. Auf

diese Weise ist es möglich, Wörter, die für die Morphologie unbekannt sind oder nicht als Eigenname erkannt wurden, als Eigennamen zu markieren. Dieser Effekt wird durch die Anwendung von Umgewichtungregeln während der Analyse erzielt.

Für die verschiedenen Typen von Eigennamen (Personennamen, geographische Namen, Firmennamen) können unterschiedlichen Teilgrammatiken erstellt werden. Die aus diesen Teilgrammatiken erzeugten Transduktoren können dann entweder vereinigt oder kaskadiert werden. Die erste Variante entspricht einer Parallelschaltung, bei der alle Teilgrammatiken parallel über der Eingabe operieren. Mit der zweiten Variante können aufgrund der Nichtkommutativität der Komposition Präferenzstrategien implementiert werden.

4.3 Eigennamenmarkierer

Der Eingabetext wird im ersten Schritt tokenisiert. Tokens, die als Wörter und nicht als Satzzeichen klassifiziert wurden, werden von der TAGH-Morphologie analysiert. Anschließend wird berechnet, ob unter den alternativen Analysen Homographiebeziehungen vorliegen und von welcher Art diese sind.

Die den einzelnen morphologischen Analysen zugeordneten Lemmata werden eindeutig auf natürliche Zahlen abgebildet und der Analyse als ID hinzugefügt. So kann später das Stützen unsicherer Eigennamen anhand der Lemma-ID und sicheren bzw. unsicheren Eigennamen realisiert werden. Wörter, die durch die Morphologie nicht analysiert werden konnten, werden als unbekannt markiert. Auch diesen Wörtern wird eine eindeutige ID zugewiesen.

Die Satzzeichen, die analysierten oder unbekannten Wörter werden anschließend inkrementell mit der Eigennamengrammatik komponiert. Obwohl die Komposition zweier Transduktoren T_1 und T_2 im schlimmsten Fall $O(|T_1||T_2)$ ist, tritt dieser Fall kaum ein, da der Automat, der den Text repräsentiert, linear ist, also aus einer linearen Abfolge der Zuständen besteht, zwischen denen Übergänge für die Lemmata mit solchen für die morphologischen Analysen alternieren.

Durch die Komposition werden alle sicheren Eigennamen mit entsprechendem Kontext und alle unsicheren Eigennamen markiert und gewichtet.

Danach werden anhand der in der Grammatik definierten Trigger und der Lemma-ID bestimmte unsichere Eigennamen durch Gewichtungen gestützt. Das Umgewichten wird durch eine einfache Abbildung realisiert, die in linearer Zeit arbeitet.

Daraufhin wird die beste Analyse berechnet. Hierfür wird ein *Single-Source Shortest-Path*-Algorithmus verwendet. Dieser arbeitet im tropischen Semiring auf azyklischen Automaten[3] in linearer Zeit (Mohri, 2002).

Letztlich wird der Ergebnistransduktor in eine interne Baumstruktur überführt und als HTML formatiert.

Das System wurde auf einem Pentium 4 mit 2 GB Arbeitsspeicher unter Linux getestet und verarbeitet im Schnitt pro Sekunde 10.000 token.

5 Evaluation

Da für das Deutsche (noch) kein Korpus mit standardisiert annotierten Daten für die Informationsextraktion vorliegt (vgl. (Neumann, 2005)), haben wir die Qualität des Eigennamenerkenners anhand eines kleinen 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 etwa 20.000 token umfassenden Korpus haben wir per Hand 852 Personennamenkontexte und 352 verschiedene Personennamen ausgezeichnet. Somit kommen auf einen Personennamen knapp 2,5 Kontexte[4]. Von diesen 852 Kontexten hat das Verfahren 826 richtig erkannt. Dies entspricht einem Recall von 96,9%. Die Precision des Verfahrens liegt bei 95,9%, d.h. 44 Eigennamen wurden fälschlicherweise als richtig erkannt.

Ein Teil der hohen Erkennungsrate läßt sich durch die hohe Abdeckung der Namen im Bereich der Politik erklären. Darüber hinaus konnten aber auch etliche unbekannte Namen dadurch erkannt werden, dass Funktions- und Berufsbezeichnungen in deren Kontext richtig identifiziert werden konnten. Eine wichtige Rolle hierbei spielt hierbei das Zusammenspiel der Lexika von Funktions- und Berufsbezeichnungen und die Kompositazerlegungskomponente der TAGH-Morphologie. Hierdurch können beispielsweise nicht im Lexikon als Ganzes enthaltene Komposita auf bekannte Bezeichnungen zurückgeführt werden. Beispiele hierfür sind die im Korpus verwendeten *Grünen-Bundestagsabgeordnete*, *CDU-Haushaltsexperte*, *SPD-Präsidiumsmitglied*, *CSU-Landesgruppenvorsitzende*, *Handelsbeauftragte*, *Flüchtlingskommissarin*, *Vorstandssprecherinnen*, *Stadtschulratspräsident*, die als Determinativkomposita behandelt, und damit semantisch auf die im Lexikon enthaltenen *Abgeordnete*, *Experte*, *Mitglied*, *Vorsitzende*, *Beauftragte*, *Kommissar*, *Sprecher* und *Präsident* zurückgeführt werden.

Bei den nicht erkannten Kontexten handelt es

[3] das Ergebnis der Komposition von Text und Eigennamenerkenner ist stets azyklisch

[4] Eine Person, z.B. *Fischer* kann durch mehrere Kontexte referenziert werden: *Joschka Fischer*, *Joseph Fischer*, *Außenminister Fischer etc.*

sich vorwiegend um homographe bzw. unbekannte Namen, bei denen der Kontext nicht ausreichend innerhalb der gegenwärtigen Grammatik für eine Klassifizierung als Eigenname war. Beispiele hierfür sind *Buntenbach sagte* oder *Kurt Biedenkopf*. Im ersten Fall kann Buntenbach sowohl Eigenname (Grünen-Abgeordnete Buntenbach) als auch ein Adjektiv-Nomen-Kompositum sein. Im zweiten Fall kann Kurt sowohl Vor- als auch Nachname sein, und Biedenkopf bezeichnet sowohl einen Nachnamen als auch einen Luftkurort in Hessen. Ein ähnliches Beispiel stellt *Roland Ries* dar. Hier ist *Roland* sowohl Vor- als auch Nachname, *Ries* hingegen ist nur als Genitiv eines Vornamens.

Eine weitere Fehlerquelle ist auf die Vorverarbeitung zurückzuführen. Wörter, die ausschließlich mit Großbuchstaben geschrieben sind, werden von der Morphologie erst nach einer Normalisierungsoperation erkannt, was in manchen Fällen zu Fehlern führt.

6 Ausblick

Die erste Evaluation des Systems auf einem Testkorpus von Zeitungstexten aus dem Bereich Politik zeigte mit einem Recall von über 96,9% und einer Precision von 95,9% bei der Erkennung von Personennamen sehr ermutigende Ergebnisse. Erfahrungsgemäß - dies zeigt die Evaluation anderer Systeme - liegt die Erkennungsrate bei geographischen Namen und Firmennamen unter der Rate von Personennamen. In einem nächsten Schritt planen wir daher die Integration von Grammatiken geographischer Namen und Firmennamen in das System. Darüber hinaus planen wir, das System auf Zeitungstexte aus anderen Ressorts anzuwenden, um zu evaluieren, wie stark sich eine weniger hohe Abdeckung von Nachnamen auf die Erkennungsrate auswirkt.

References

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

A. Geyken and Th. Hanneforth. 2005. Tagh: A complete morphology for german based on weighted finite state automata. *In Proceedings of FSMNLP 2005, Lecture Notes in Artificial Intelligence*.

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

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

T. Hanneforth. 2005b. Potsdam finite state library: C++ library for finite state device operations.

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

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

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

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

1998. *Message Understanding Conference Proceedings: MUC-7*.

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

G. Neumann. 2005. A hybrid machine learning approach for information extraction from free texts. *In Spiliopoulou at al. (Eds). From Data and Information Analysis to Knowledge Engineering. Springer series Studies in Classification, Data Analysis, and Knowledge Organization*.

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

M. Rössler. 2002. Using markov madels for nemaed entity recognistion in german newspapers. *In Proceedings of the ESSLIS'02 Workshop on Machine Learning Approaches on Computational Linguistics*.

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

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

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