Universität Zürich, Institut für Computerlinguistik
Referent: Prof. Dr. Michael Hess, Betreuer: Dr. Manfred Klenner
Lizenziatsarbeit der Philosophischen Fakultät der Universität Zürich

Machine Learning für Koreferenz-Auflösung

Don Tuggener
Rötelstr.19
8006 Zürich
don.tuggener@gmail.com
076 398 04 41

Abgegeben im Oktober 2009

Zusammenfassung

Koreferenz-Auflösung verfolgt das Ziel, verschiedene Einheiten in einem Text zu finden, die sich auf das gleiche Objekt ausserhalb des Texts beziehen. In verschiedenen Gebieten der automatischen Verarbeitung von natürlicher Sprache spielt die Auflösung von Koreferenz eine zentrale Rolle (*Question Answering, Information Extraction, Information Retrieval* etc.). Seit Mitte der 90er-Jahre gewinnen Machine Learning-Verfahren an Popularität und lösen zunehmend die auf manuell erstellten Regelsystemen basierenden Ansätze ab. Diese Arbeit verfolgt die Entwicklung Machine Learning-basierter Systeme zur Koreferenz-Auflösung mit einem speziellen Augenmerk auf der Rolle der semantischen Merkmale bei der Auflösung nominaler Anaphora. Ein System wird konzipiert und implementiert um experimentell festzustellen, inwiefern die Ermittlung von semantischer Ähnlichkeit zwischen einer Anapher und ihren möglichen Antezedenzien hilfreich für die Auflösung ist.

Abstract

Coreference resolution aims at identifying the entities in a text that refer to the same real-word objects. This plays a major role in different areas of natural language processing (Question Answering, Information Extraction, Information Retrieval etc.). Since the Mid-90ties, machine learning has gained increased popularity in the field and is continuously replacing approaches that implement hand crafted rule systems. This thesis retraces the development of machine learning systems for coreference resolution focusing on the role of semantic features in resolving nominal anaphora. A system is designed and implemented to evaluate the helpfulness of calculating the semantic similarity of an anaphor and its possible antecedences.

Danksagung

Ich möchte mich zuerst und vor allem beim Betreuer der vorliegenden Arbeit, Dr. Manfred Klenner, bedanken. Durch kritische Rückmeldungen, Hinweise, Ideen und Konzepte hat er massgeblich zu dieser Arbeit beigetragen. Auch durften Teile seiner Software-Applikationen für diese Arbeit verwendet werden (Teile des Programms zur Paargenerierung und Filterung der Paare und der CEAF-Scorer). Durch sein Interesse und seine kurzen Reaktionszeiten auf Anfragen hat er das Verfassen dieser Arbeit gefördert und motiviert.

Darüber hinaus bedanke ich mich bei meinem Referenten, Prof. Dr. Michael Hess, der den schriftlichen Teil der Arbeit begutachtet hat. Dr. Gerold Schneider danke ich für die Verfügungstellung seines Parsers und die technische Hilfe bei der Installation desselben.

Meinen Kolleg/Innen vom Fachverein *CLinZ/CH* danke ich für den Austausch über das Verfassen einer Lizenziatsarbeit. Adrian Schindler und Thomas Blumer danke ich für das kurzfristige Korrigieren der Arbeit. Schliesslich gilt mein Dank meiner Familie und meinem Freundeskreis, die mich während des Verfassens der Arbeit unterstützt haben.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

1 Einleitung

1.1 Motivation, Ziel und Aufbau der vorliegenden Arbeit

Die vorliegende Arbeit verfolgt das Ziel, die Entwicklung von automatisierten Verfahren zur Koreferenz- und Anaphernresolution aufzuzeigen und nachzuvollziehen. Dabei werden zwei Schwerpunkte gesetzt:

1. Der Fokus auf Verfahren, die Techniken aus dem *Machine Learning* (ML) benutzen.
2. Die Verwendung semantischer Ressourcen bei der Auflösung nominaler Anaphora.

In der Forschung ist weitgehend anerkannt, dass die Auflösung nominaler Anaphora sich schwieriger gestaltet als diejenige von pronominaler Anaphora. Ein Grund dafür ist, dass Nomen viel weniger oft anaphorisch sind als Pronomen. In ML-Verfahren wird versucht, relevante Merkmale zu ermitteln, die die Auflösung nominaler Anaphern verbessern. Nebst dem Bestimmen der Anaphorizität (ob ein Nomen anaphorisch ist oder nicht) und Ähnlichkeitsmassen auf den Oberflächenformen von Wörtern (*String Match*), spielen semantische Merkmale eine zentrale Rolle. Gerade bei der Bestimmung einer Koreferenzmenge, die Eigennamen enthält, sind sie wichtig, da syntaktische, positionelle und morphologische Merkmale nicht ausreichen. Man nehme folgenden Beispielsatz:

> „George W. Bush announced a meeting with Joe. He will have the opportunity to ask the president any questions he may have."

Konventionelle Merkmale, wie die Übereinstimmung von Person, Genus und Numerus (3. Person, Maskulin, Singular) oder die Parallelität der grammatikalischen Funktionen (Subjekt, Objekt), können „George W. Bush" im Gegensatz zu „Joe" nicht eindeutig als Antezedens für „the president" bestimmen. Die Koreferenz-Relation kann nur anhand von Weltwissen – dass George W. Bush Präsident war – eindeutig erkannt werden.

Semantische Merkmale sind in ML-Systemen für Koreferenz-Auflösung weitgehend etabliert. Üblicherweise werden semantische Klassen festgelegt (z.B. *Person, Organization, Location, Time, Money*), denen mögliche Antezedens- und Anaphern-Kandidaten untergeordnet werden. „George W. Bush" und „Joe" aus dem obigen Beispielsatz würden durch solch ein Verfahren als *Person* klassifiziert werden. Damit wäre eine ausreichende Disambiguierung zwischen „George W. Bush" und „Joe" in Bezug auf „president" noch nicht gewährleistet. Wenn man George W. Bush als *President* und Joe als *Plumber* klassifizieren würde, wären semantisch

gesehen immer noch beide mögliche Antezedenzien von „president“. Wir wissen aber, dass *President* im Gegensatz zu *Plumber* näher bei „president“ liegt, bzw. identisch ist. Mit der Ermittlung von semantischer Ähnlichkeit lassen sich solche Differenzierungen vornehmen.

Durch diese Beobachtungen motiviert, soll in dieser Arbeit ein System für Koreferenz-Auflösung konzipiert und implementiert werden, dass für Nominalphrasen (NPs) mit Eigennamen (*Named Entities*) möglichst spezifisch herausfindet, was sie „sind“. Dabei wird vorerst auf die traditionelle Festlegung von semantischen Hauptkategorien (wie *Person, Organization, Location, Time, Money*) verzichtet und in einem ersten Schritt untersucht, ob ein mögliches Antezedens mit einer Anapher in einem Hyponymie-, Synonymie- oder Meronymie-Verhältnis steht. In einem zweiten Schritt soll festgestellt werden, inwiefern die Messung von semantischer Ähnlichkeit zwischen einem Antezedens und einer Anapher die Koreferenz-Auflösung nominaler Anaphern verbessern kann.

Die folgende Arbeit ist grob in zwei Teile gegliedert. Im ersten Teil wird eine linguistische Definition von den Phänomenen Koreferenz und Anaphora gegeben (Kapitel 1.2), gefolgt von einer Einführung in die Techniken des Machine Learnings (Kapitel 2), respektive TiMBL (Kapitel 2.2). Die Verwendung von ML und semantischen Merkmalen für die Auflösung nominaler Anaphern anhand von Fachliteratur wird anschliessend beleuchtet (Kapitel 2.4). Dabei werden auch Verfahren betrachtet, die die Abbildung von Named Entities auf Nomen ermöglichen, die in elektronischen Wortnetzen gefunden werden können (Kapitel 2.5). Ausserdem wird ein Blick auf gängige Evaluationsmasse für die Koreferenz-Auflösung und deren Vor- und Nachteile geworfen (Kapitel 3). Aus der Betrachtung der historischen Entwicklung diverser Verfahren zeigt sich, dass die Auflösung nominaler Anaphora die schwierigste Aufgabe des Forschungsgebiets ist.

Im zweiten Teil der Arbeit wird aufgrund der im ersten Teil gewonnen Erkenntnisse ein ML-System zur Koreferenz-Auflösung konzipiert und implementiert (Kapitel 4). Experimente mit verschiedenen semantischen Ressourcen zeigen deren Bedeutung für die Auflösung nominaler Anaphora auf (Kapitel 5).

1.2 Linguistischer Hintergrund von Anaphora und Koreferenz

Koreferenz und Anaphora sind Phänomene, die auftreten, wenn auf sprachlicher Ebene mit verschiedenen Formen auf dasselbe Objekt ausserhalb eines Texts verwiesen wird. Die linguistischen Grundlagen sollen hier nach Linke *et al.* (2001, S. 215-218) eingeführt werden.

Eine einfache Form von Koreferenz ist die Rekurrenz. Sie entsteht, wenn mit demselben sprachlichen Ausdruck mehrmals auf ein Objekt verwiesen wird:

> (1) Gestern habe ich einen **Vogel** beim Nestbau beobachtet. Der **Vogel** war ganz klein, hat aber trotzdem ziemlich grosse Zweige angeschleppt. Als Nistplatz hatte sich der **Vogel** ausgerechnet die Nische über unserem Rolladen ausgesucht.

Die wörtliche Wiederholung in (1) wird stilistisch oft als unbefriedigend empfunden. Die wiederholte Referenz auf ein aussersprachliches Objekt wird daher oft anhand von Substitutionen oder Pro-Formen umgesetzt. Bei der Substitution werden bei der wiederholten Referenz oft Synonyme oder Unter- und Oberbegriffe (Hyponyme/Hyperonyme) der vorangehenden Referenz verwendet:

> (2) Auf dem Markt heute morgen gab es ganze Stände voll mit verschiedenen **Petunien**. Diese **Balkonpflanzen** sind für mich einfach die allerschönsten.
> (3) Das ***Gold*** wurde von einem **Drachen** bewacht. Der **Lindwurm** tötete jeden, der den ***Schatz*** erobern wollte.

Zu den Pro-Formen gehören alle Arten von Pronomen (*er, sie, mein, sich* etc.), Adverbien (*dort, da* etc.) und Pronominaladverbien (*wobei, darauf, womit* etc.). Sie haben gemein, dass sie erst im sprachlichen Kontext ihrer Verwendung Bedeutung durch ihre Referenz erhalten:

> (4) Das ist **Markus**. **Er** ist Linguist.
> (5) **Paul** ist in ***Griechenland***. Es gefällt **ihm** ***dort***.

Das heisst, Pro-Formen sind ohne Verwendungskontext meist semantisch leer[1] und sind daher meistens anaphorisch: Ausdrücke, die im Text zurückverweisen, werden als Anaphern bezeichnet (in Beispiel (4) „Er“) und die Ausdrücke, auf die zurückverwiesen wird, als Antezedenzien (in (4) „Markus“).

[1] Ausnahme ist die pleonastische Verwendung von *Es* in Aussagen wie „Es regnet/Es ist an der Zeit, dass ...“ etc., in denen *Es* keine spezifische Referenz auf ein Objekt ausserhalb des Texts hat, sprich immer sematisch leer ist.

Der Prozess der Anaphernauflösung kann nun grundsätzlich durch drei Schritte beschrieben werden:

1. Identifikation der Anapher
2. Bestimmung möglicher Antezedenzien
3. Auswahl eines Antezedens

Dieses Modell zeigt auf, wie Menschen während dem Lesen und Verstehen von Texten – bzw. bei der Auflösung von Anaphora – vorgehen und gilt gleichzeitig als Grundlage für die maschinelle Modellierung dieser Strategie. Dabei werden Anaphern in pronominale Anaphern (Beispiele 4,5), bei denen die Anapher mit einem Pronomen realisiert wird, und nominale Anaphern (Beispiele 1-3), wo die Anapher ein Nomen ist, unterteilt. Diese Aufteilung ist für die maschinelle Verabeitung sinnvoll, da die beiden Arten von Anaphern verschiedene Auflösungsstrategien erfordern (so z.B. zu erkennen, dass in (2) „Petunien“ ein Unterbegriff von „Balkonpflanzen“ ist, oder dass in (4) „Markus“ und „Er“ bezüglich Numerus, Genus und Person kongruent sind).

Koreferenz unterscheidet sich insofern von Anaphora, als dass ein Ausdruck zu einem anderen anaphorisch sein kann, ohne dass die Audrücke eine spezifische aussersprachliche Referenz haben. In Beispiel (6) ist „his“ anaphorisch zu „man“, wobei „man“ keine spezifische aussersprachliche Referenz hat.

(6) Every man is entitled to his opinion.

Die Anaphernresolution verfolgt das Ziel, einer Anapher das entsprechende Antezedens zuzuweisen. Koreferenz-Auflösung versucht alle Ausdrücke in einem Text, die sich auf die gleiche aussersprachliche Einheit beziehen, zu finden und in Äquivalenz- bzw. Koreferenzmengen zu fassen.

Nachdem die Grundlagen der Phänomene Koreferenz und Anaphora gegeben sind, wird im folgenden Kapitel eine genauere Unterscheidung der verschiedenen Arten von Anaphora vorgenommen.

1.3 Verschiedene Arten von Anaphora

Koreferenz und Anaphora erscheinen in natürlicher Sprache in verschiedenen Formen. Ševciková (2005) ermittelt vier Hauptkategorien, anhand derer verschiedene Arten von Anaphora beschrieben werden können.

- **Syntaktische Realisierung**: Anaphern in Form von Nominalphrasen (NP), Verbalphrasen (VP), Adverbien und Nullanaphern. Die NP-Anaphern bestehen aus vollständigen NPs und Pro-Formen. Letztere sind lexikalisch leer und erhalten erst im Kontext ihrer Verwendung Bedeutung (Personal-, Reflexiv-, Possessivpronomen). Vollständige NPs sind mit lexikalischer Bedeutung besetzt und tragen zusätzliche Information in die Koreferenzmenge, in die sie gehören, indem sie beispielsweise vorhandenes, Text externes Weltwissen im Text explizit machen („George W. Bush [...] the former president“).
  In Verbalphrasen-Anaphern ist die Anapher ein Verb, wie beispielsweise in „As the oil price **rose**, so **did** oil production.“ Nullanaphern bezeichnen Konstruktionen, in denen eigentlich anaphorische Ausdrücke durch Ellipse im Satz entfallen; so zum Beispiel in koordinierten Sätzen nach einer Konjunktion: „Ross carefully folded his trousers and ∅ climbed into bed.“ Adverbanaphern bezeichnen temporale und lokalitätsbezogene Adverbien, die deiktisch sind (*then, there*).

- **Positionelle Unterscheidung**: Intrasententielle (Antezedens und Anapher sind im gleichen Satz) und intersententielle (Antezedens und Anapher sind nicht im gleichen Satz) Beziehungen.

- **Referenzvermögen**: Unterscheidung zwischen referenzidentischer und referenzähnlicher Anaphora. In „The man who gave his **paycheck** to his wife was wiser than the man who gave **it** to his mistress.“ ist „it“ anaphorisch zu „paycheck“, wobei „paycheck“ zwei unterschiedliche Referenten hat, da zwei verschiedene Löhne gemeint sind. Auch in Beispiel (6) hat „man“ keinen bestimmten Referenten. Solche Beziehungen sind referenzähnlich im Gegensatz zu referenzidentischen Anaphern, die auf denselben, spezifischen Referenten ausserhalb des Texts verweisen, wie das Antezedens.

- **Erschliessbarkeit**: Direkte Anaphern sind Konstellationen, in denen die Anapher eine wörtliche Wiederholung des Antezedens ist (Beispiel 1) oder die Anapher eine Pro-Form ist. Bei assoziativen Anaphern wird die Koreferenz durch eine semantische Relation erschlossen. Das heisst, Anapher und Antezedens stehen beispielsweise in einem Hyponymie- oder Meronymie-Verhältnis (Beispiel 3).

Bei der maschinellen Verarbeitung der verschiedenen Arten von Anaphora ergeben sich unterschiedliche Probleme. In Kapitel 2.4 wird genauer auf diverse Lösungsansätze, bzw. auf Merkmale eingegangen, die die verschiedenen Konstellationen charakterisieren. Hier sollen kurz grundsätzliche Konzepte für die Auflösung von direkter und indirekter Anaphora gegeben werden.

- Direkte Anaphern: Anaphorische Pronomen sind oft bezüglich Genus, Person und Numerus mit ihren Antezedenzien kongruent. Auch positionelle Merkmale (Distanz) und die grammatikalische Rolle spielen bei ihrer Auflösung eine zentrale Rolle. Ein Spezialfall ist die pleonastische Verwendung von *It*, die jeweils nicht anaphorisch ist. Pleonastische Pro-Formen müssen erkannt und gefiltert werden. Dies ist anhand von Listen mit Adjektiven möglich, in deren Kontext *It* pleonastisch ist, so z.B. bei *It* in Subjektposition von *be* mit Adjektiven wie *neccessary/possible/difficult*[2]. Direkte Nominalanaphern können oft einfach über String Match-Verfahren aufgelöst werden.

- Um die Erschliessung von indirekten, assoziativen Anaphern nachzubilden, werden oft Wortnetze – z.B. *WordNet* (Fellbaum, 1998) – benutzt, die semantische Relationen zwischen Konzepten denotieren. Dabei werden Anapher und Antezedens auf Konzepte in den Wortnetzen abgebildet und dann nach Hypnonymie-, Synonymie- und Meronymie-Relationen zwischen ihnen gesucht.

Welche Arten anaphorischer Beziehungen für die Anaphernresolution relevant sind, ist abhängig vom Verwendungszweck eines Systems. Auflösung von Anaphora und Koreferenz wird seit Mitte der 90er-Jahre oft im Kontext von Text Mining betrieben, so z.B. anlässlich der *Message Understanding Conferences* (MUC). Ein aktuelles Anwendungsgebiet ist das Relation Mining beispielsweise im biomedizinischen Bereich. Im Relation Mining spielt die Auflösung von VP-, Adverb- und Nullanaphern eine weniger wichtige Rolle als diejenige von pronominaler und nominaler Anaphora, da oft mit Mustern, die Subjekte und Objekte von bestimmten Verben suchen, nach Relationen gesucht wird. Auch ist die Häufigkeit der verschiedenen Arten von Anaphora unterschiedlich (gerade für verschiedene Textgenres) und muss bei der Konzeption eines Systems berücksichtigt werden. Diese Häufigkeiten werden bei der Konzeption und Implementation eines ML-Systems zur Koreferenz-Auflösung in Kapitel 4 genauer betrachtet. Im Folgenden wird eine Einführung in das beispielbasierte Lernen gegeben, mit dem Fokus darauf, wie es für die automatische Auflösung von Koreferenz genutzt werden kann.

[2] Z.B. Lappin & Leass (1994, S. 538)

2 Machine Learning für Koreferenz-Auflösung

Ein gängiges Verfahren bei statistischen Methoden zur Koreferenz-Auflösung ist die Verwendung eines Machine Learning-Algorithmus. Für alle möglichen Koreferenz-Kandidaten in einem Text, die sogenannten *Markables*, werden Merkmalsvektoren generiert. Aus einem *Goldstandard*, einem Korpus, das manuell annotierte Koreferenzmengen enthält, wird ein Trainingskorpus generiert. Dann wird ein *Classifier* über dem Trainingskorpus trainiert. Dabei bewertet der ML-Algorithmus die Relevanz der mitgegebenen Merkmale anhand des jeweiligen *Gain Ratios*. Das Gain Ratio sagt aus, welchen Anteil ein jeweiliges Merkmal an der Entscheidung, ob zwei Koreferenz-Kandidaten tatsächlich koreferent sind, trägt. Ein Teil – üblicherweise jeweils 10-20% – des Trainingskorpus wird vom Training des Classifiers ausgeschlossen und dann als Testkorpus zur Evaluation verwendet.

Machine Learning kennt verschiedene Ansätze. Dabei kann zwischen regelbasiertem (*rule induction, rule-based learning*) und beispielbasiertem (*memory-based*) Lernen unterschieden werden. Die Lernverfahren können folgendermassen charakterisiert werden:

- Beim regelbasierten Lernen lernt der Classifier aus den positiven und negativen Beispielen anhand der jeweiligen Merkmale und deren Werte eine Klassifikationsstrategie, die auf unbekannte Situationen angewendet werden kann.
- Bei der beispielbasierten Methode vergleicht der Classifier eine neue Situation mit bereits bekannten und klassifizierten Situationen und bestimmt anhand der Ähnlichkeit der Merkmalswerte die Klasse der neuen Situation.

Die Lernverfahren können ausserdem in supervisierte und unsupervisierte Verfahren unterteilt werden. Bei supervisierten Verfahren wird der Classifier anhand eines Trainingskorpus trainiert, in dem – z.B. bei der Koreferenz-Auflösung – Koreferenzmengen manuell annotiert wurden. Unsupervisierte Verfahren hingegeben arbeiten ohne manuelle Annotation von Koreferenzmengen im Trainingskorpus. Die Klassifikation von möglichen Koreferenz-Paaren wird dabei als Aufgabe für ein *Clustering*-Verfahren betrachtet und beispielsweise mit Hilfe von *Support Vector Machines* oder anderen Clustering-Algorithmen gelöst.

Im Folgenden liegt der Fokus auf dem beispielbasierten Lernen[3], da im zweiten Teil der Arbeit ein solches Verfahren für die Koreferenz-Auflösung verwendet wird. Im folgenden

[3] Siehe z.B. Phan *et al.* (2005) für die Verwendung von *Maximum Entropy*-Modellen und *Predictive Association Rules* um ein regelbasiertes ML-System zu trainieren.

Kapitel werden verschiedene Aspekte des beispielbasierten Lernens genauer beleuchtet und der *Tilburg Memory-Based Learner* (TiMBL) gemäss Daelemans *et al.* (2007) eingeführt.

2.1 Grundlagen des Machine Learning: *k*-NN und C4.5

Beispielbasierte Lernverfahren folgen der Hypothese, dass kognitive Aufgaben bewerkstelligt werden, indem bei der Beurteilung einer neuen Situation auf gespeichertes Wissen aus früheren Erfahrungen zurückgegriffen wird. Grundsätzlich wird eine neue Situation also mit Bekanntem verglichen und aufgrund von Ähnlichkeit beurteilt.
Algorithmen für beispielbasiertes Lernen basieren auf dem sogenannten *k-Nearest Neighbor* (*k*-NN) Algorithmus, der ursprünglich von Cover & Hart (1967) entwickelt wurde. Die Idee dabei ist, dass bei der Beurteilung – oder im Sinne diverser NLP-Anwendungen: bei der Klassifikation – neuer Situationen eine Anzahl (*k*) ähnlicher Vorkommnisse (Nearest Neighbors) im Speicher oder „Gedächtnis“ (*Memory*) ermittelt wird und anhand deren Klassen ein Majoritätsentscheid über die Klasse der unbekannten Situation gefällt wird. *k*-NN-Verfahren fällen prinzipiell also quantitätsbasierte Entscheide.
Das Memory – oder spezifisch in NLP-Anwendungen: Trainingskorpus – besteht dabei aus Vektoren, die verschiedene Merkmale der gespeicherten Instanzen (der „Erfahrungen“) und deren Klassifikationen kodieren. Die Erfahrungen werden anhand der Merkmale, bzw. deren Werte, in den Vektoren charakterisiert. Bei der Klassifikation einer neuen Situation X wird folgendermassen vorgegangen:

1. Erstellen des Merkmalsvektors für X: Dabei werden alle Merkmale mit X entsprechenden Werten belegt. Die Struktur und Länge des Vektors ist durch die Vektoren der Instanzen aus dem Trainingskorpus vorgegeben.

2. Finden der *k*-Nearest Neighbors: Aufgrund der Ähnlichkeit der Merkmalswerte des Vektors von X mit jenen des Trainingskorpus werden *k* Vektoren aus dem Trainingskorpus selektiert.

3. Klassifikation: X erhält die Klasse, die in den Vektoren (den Nearest Neighbors) aus dem Trainingskorpus am häufigsten vertreten ist.

Ausschlaggebend für eine möglichst gute Klassifikation sind folglich drei Komponenten:

1. Relevanz der Merkmale: Ist das denotierte Wissen adäquat abgebildet? Liefern die festgehaltenen Merkmale eine adäquate und ausreichende – sprich zwischen den verschiedenen Klassen möglichst diskriminierende – Repräsentation?

2. Definition von Ähnlichkeit: Was ist Ähnlichkeit? Lassen sich mit den definierten Ähnlichkeitsmassen sinnvolle Differenzen zwischen den Merkmalswerten berechnen?

3. Qualität des Trainingskorpus: Um konsistente Klassifikationen vornehmen zu können, muss das Trainingskorpus als Entscheidungsgrundlage konsistent sein.[4]

Nebst den *k*-NN-Algorithmen werden in verschiedenen NLP-Anwendungen auch sogenannte *Decision Trees* (Entscheidungsbäume) verwendet. Der populärste Algorithmus zum Erstellen von Entscheidungsbäumen ist *C4.5* (Quinlan, 1993). C4.5 basiert nicht auf Ähnlichkeit, sondern benutzt Entropie als Kriterium zum Erstellen eines Entscheidungsbaums[5]. Dabei wird ebenfalls ein Trainingskorpus verwendet, das Merkmalsvektoren von Instanzen und entsprechende Klassifikationen beinhaltet. Beim Erstellen eines Knotens im Entscheidungsbaum wird ermittelt, welches Merkmal der Merkmalsvektoren das Trainingskorpus lokal am effektivsten und eindeutigsten in zwei Subkorpora aufteilt. Die Effektivität eines Merkmals wird dabei anhand seines *Information Gain* ermittelt.
Information Gain ist ein Mass dafür, inwiefern ein Merkmal die Entropie des Trainingskorpus verringert, wenn es als Knoten in einem Baum eingesetzt wird. Ist das Merkmal mit maximalem Information Gain-Wert bestimmt, wird rekursiv in der sich allmählich ergebenden Verästelung des Baums nach den jeweiligen Information Gain-Maxima gesucht, bis die Entropie 0 ist. Diese Knoten entsprechen den Blättern des Entscheidungsbaums. Pfade vom Wurzelknoten zu den Blättern denotieren folglich eine Merkmalswerts-abhängige Klassifikation.
Bei der Klassifikation einer Instanz X wird nun ebenfalls zuerst der entsprechende Merkmalsvektor generiert. Dann wird beim Wurzelknoten des Entscheidungsbaums, der das Merkmal mit dem höchsten Information Gain beinhaltet, begonnen und – den Werten des Merkmalsvektors von X entsprechend – einem Pfad bis zu einem Blatt gefolgt, das eine entsprechende Klassifikation beinhaltet.

Entscheidungsbäume und Nearest Neighbor-Algorithmen sind zwei Klassifikationsverfahren, die ihre Gemeinsamkeit in der Lernressource – Merkmalsvektoren aus einem (annotierten) Trainingskorpora – und ihre Unterschiede in der Klassifikationsstrategie – Ähnlichkeit versus Verringerung von Entropie – haben. Ein Vorteil von Entscheidungsbäumen ist, dass die gelernte Klassifikationsstrategie vom Trainingskorpus abstrahiert und komprimiert

[4] Bei den oft manuell annotierten Trainingskorpora sind Lücken und Inkonsistenz kaum vermeidbar. Ein Mass für die Konsistenz von manuell annotierten Korpora ist beispielsweise das *Inter Annotator Agreement* (s. Kapitel 4.1).
[5] S. Abbildung 1 für ein Beispiel eines Entscheidungsbaums.

```
STR_MATCH = +: + 944 (66.3%)
STR_MATCH = -:
        :...J_PRONOUN = -:
            :...APPOSITIVE = +: + 111 (7.8%)
STR_MATCH = -:
        :...J_PRONOUN = -:
            :...APPOSITIVE = -:
                :...ALIAS = +: + 163 (11.5%)
STR_MATCH = -:
        :...J_PRONOUN = +:
            :...GENDER = I:
                :...I_PRONOUN = +: + 77 (5.4%)
STR_MATCH = -:
        :...J_PRONOUN = +:
            :...GENDER = I:
                :...I_PRONOUN = -:
                    :...DIST <= 0:
                        :...NUMBER = +: + 128 (9.0%)
```

Abbildung 1: C4.5 Entscheidungsbaum gemäss Soon *et al.* (2001, S. 536). Der Wurzelknoten beinhaltet das Merkmal mit dem höchsten Information Gain. Allein anhand dieses Merkmals (STR_MATCH) konnten bei Soon *et al.*'s Experimenten 66,3% der nominalen Anaphern aufgelöst werden.

abgespeichert werden kann. Beim beispielbasierten Lernen werden alle Beispiele aus dem Trainingskorpus gespeichert und abgefragt – deshalb werden k-NN-Verfahren auch als *Lazy Learner* bezeichnet.

Nach der grundlegenden Einführung in Aspekte des Machine Learning wird im folgenden Kapitel TiMBL (Daelemans *et al.*, 2007) als eine Implementation des k-NN-Ansatzes betrachtet, die das beispielbasierte Lernen mit der Verwendung von Entscheidungsbäumen verknüpft. Dabei wird auf verschiedene Details, wie die Bestimmung von Ähnlichkeit der Merkmalswerte, genauer eingegangen.

2.2 TiMBL

Die Software-Implementation des k-NN-Algorhitmus in TiMBL unterscheidet sich gemäss Daelemans *et al.* (2007, S. 21) insofern von ursprünglichen k-NN-Algorithmen, als dass k nicht für die Anzahl der Nearest Neighbors steht, die bei der Klassifikation einer neuen Instanz

X zu berücksichtigen sind. In TiMBL gibt *k* vielmehr die Anzahl unterschiedlicher Distanzen vor, in denen von X ausgehend Nearest Neighbors gesucht werden sollen. Distanz Δ wird dabei als *Overlap*-Wert definiert; das heisst als Summe der jeweiligen Ähnlichkeit δ der n Merkmalswerte (x_i, y_i) von X und einem Nachbar Y:

$$\Delta(X, Y) = \sum_{i=1}^{n} \delta(x_i, y_i)$$

Die Ähnlichkeit zwischen den Merkmalen $\delta(x_i, y_i)$ ist bei numerischen Werten für x_i und y_i gegeben durch die Differenz zwischen den spezifischen Merkmalswerten x_i und y_i, normalisiert durch die Differenz der maximalen und minimalen Werte des Merkmals $(max_i - min_i)$ im Trainingskorpus:

$$\delta(x_i, y_i) = abs\left(\frac{x_i - y_i}{max_i - min_i}\right)$$

Bei nicht numerischen Werten (Strings) für die Merkmale gilt $\delta(x_i, y_i) = 0$ wenn x_i und y_i nicht identisch sind (kein String Match), respektive $\delta(x_i, y_i) = 0$ wenn x_i und y_i identisch sind.

Das Overlap-Mass funktioniert nur bei numerischen Werten sinnvoll. Bei der Messung von Ähnlichkeiten zwischen Zeichenketten können nur binäre Entscheide gefällt werden. So erhält das Paar *bathe* und *bathes* den gleichen Ähnlichkeitswert wie beispielsweise *bathe* und *rumour*. Um eine feinere Messung von Ähnlichkeiten zwischen Strings zu ermöglichen, verwendet TiMBL die *edit* oder *Levenshtein*-Distanz[6]. Die Levenshtein-Distanz misst die Anzahl benötigter Veränderungen (Einfügen, Löschen, Ersetzen), um einen String in einen anderen zu transformieren. Für *bathe* → *bathes* ergäbe sich eine Levenshtein-Distanz von 1 (einen Buchstaben einfügen) und für *bathe* → *rumour* eine Levenshtein-Distanz von 6 (fünf Buchstaben ersetzen, einen einfügen).

In TiMBL kann die Anzahl der Nearest Neighbors je nach festgelegter Anzahl Distanzen *k* stark variieren. Die unterschiedliche Dichte in den verschiedenen Regionen des Vektorraums beeinflusst die Anzahl der Nearest Neighbors: In dicht besiedelten Regionen werden viele Instanzen zur Klassifikation herbeigezogen und in weniger dichten Regionen weniger. Um dies zu berücksichtigen, werden Klassifikationen anhand der Anzahl beteiligter Nearest Neighbors gewichtet (s. Kapitel 2.2.2).

[6] s. Levenshtein (1966)

2.2.1 Information Gain und Gain Ratio

Bei der Auswahl der Nearest Neighbors einer Instanz X kommt nun die Gewichtung der Merkmale hinzu. Gewisse Merkmale aus den Merkmalsvektoren tragen mehr Information zu einer Klassifikation von X bei als andere. Als Mass für den Beitrag eines Merkmals an die adäquate Beschreibung von X wird nun wieder der Information Gain verwendet. Der Information Gain legt fest, wie spezifisch die Merkmale und deren Werte für jeweils eine Instanz und deren Klassifikation sind. Information Gain kann in diesem Sinne als Wert für den Informationsgehalt eines Merkmals verstanden werden. Daelemans *et al.* (2007, S. 22) führen gemäss Quinlan (1993) als Beispiel ein hypothetisches Patientenregister eines Krankenhauses an. Jede Krankenakte ist dabei mit einer eigenen, zufälligen Identifikationsnummer versehen. Diese Nummer hätte in Bezug auf die eindeutige Identifikation der Krankenakte einen sehr hohen Informationsgehalt, da sie die Akte mit einem einmaligen Merkmalswert beschreibt und so eindeutig charakterisiert. Jedoch ist sie durch ihre Einmaligkeit im Korpus des Patientenregisters bei der Klassifikation neuer Instanzen unbrauchbar, da es nicht möglich ist, Ähnlichkeiten in Bezug auf zufällige, einmalige Zahlen zu definieren. Das heisst, Information Gain tendiert dazu, Merkmale mit vielen verschiedenen Werten zu stark zu gewichten.

Um diese Übergewichtung zu minimieren, definierte Quinlan (1993) das sogenannte *Gain Ratio*. Gain Ratio kann als normalisierter Information Gain verstanden werden. Information Gain wird dabei durch die Entropie der Merkmalswerte (*Split Info*) dividiert. Die resultierende Gewichtung w_i kann dann in die Berechnung der Distanz zwischen X und Y miteinbezogen werden:

$$\Delta(X, Y) = \sum_{i=1}^{n} w_i \delta(x_i, y_i)$$

Information Gain kann also als Wert verstanden werden, der den Informationsgehalt eines Merkmals in Bezug auf die eindeutige Charakterisierung einer Instanz und deren Klassifikation beschreibt. Gain Ratio hingegen hält fest, inwiefern ein Merkmal bei der Klassifikation von neuen, unbekannten Instanzen hilfreich ist und stellt somit ein wichtiges Mass für die Evaluation der Nützlichkeit von Merkmalen dar.

2.2.2 Distanzgewichtete Klassifikation

Nachdem gezeigt wurde, wie die Nützlichkeit der Merkmale bei der Klassifikation bestimmt werden kann, wird nun betrachtet, wie gelernte Klassifikationen selbst bewertet werden können.

Wie aufgezeigt wurde, basieren *k*-NN-Klassifikationen auf Majoritätsentscheiden. TiMBL berücksichtigt nicht die klassische Bedeutung von *k* als Anzahl der nächsten Nachbarn, die für eine Klassifikation konsultiert werden, sondern definiert *k* als vorgegebene Distanz zwischen den Merkmalsvektoren von X und Y. Aus der unterschiedlichen Dichte der Vektoren in den verschiedenen Regionen des Vektorraums ergibt sich, dass die Anzahl der hinzugezogenen Vektoren aus dem Trainingskorpus nicht immer gleich ist, sondern tendenziell immer verschieden. Das bedeutet, dass ein vorgegebener Distanzradius *r* in einer dichteren Region des Vektorraums mehr nächste Nachbarn ermittelt, als in einer weniger dichten Region.
Wenn man nun davon ausgeht, dass Entscheide prinzipiell besser werden, je mehr nahe Nachbarn vorhanden sind, drängt es sich auf, den Majoritätsentscheid danach zu gewichten. Daelemans *et al.* (2007, S. 26) verwenden folglich das *Inverse Linear*-Mass, um die Dichte, respektive die Distanz der nächsten Nachbarn zum Merkmalsvektor von X, festzuhalten. Dabei erhält der entfernteste Nachbar das Gewicht 0 und der nächste das Gewicht 1. Die Gewichte für die restlichen Nachbarn werden entsprechend ihrer Distanz zu X linear skaliert.

2.2.3 Baumbasiertes Indexieren

Wie in Kapitel 2.1 erwähnt, entsteht bei Lazy Learning-Verfahren ein hoher Rechenaufwand, da die Merkmalsvektoren des Trainingskorpus als Liste oder Array abgespeichert und beim Klassifizieren einzeln abgefragt werden. Der Rechenaufwand verhält sich proportional zur Anzahl der Trainingsvektoren multipliziert mit der Anzahl der Merkmale. Der Aufwand kann nun minimiert werden, indem das Trainingskorpus auf einen Entscheidungsbaum abgebildet wird.

Dabei folgt der Algorithmus IGTREE von Daelemans *et al.* (2007, S. 30) dem Verfahren von C4.5 (Quinlan, 1993) und benutzt Information Gain als Kriterium zur Bestimmung des jeweiligen Merkmals, das lokal als Knoten verwendet wird. Das Merkmal mit dem höchsten Information Gain wird als Wurzelknoten platziert. Rekursiv werden die weiteren Merkmale dem Wurzelknoten als Tochterknoten untergeordnet und durch Pfade, die Merkmalswerte denotieren, miteinander verbunden. Die Minimierung des Rechenaufwands wird dadurch erreicht, dass diverse Pfade mit gleichen Merkmalswerten in einen einzelnen Pfad zusammengefasst werden können, da ähnliche Instanzen gleiche Teilstrecken beim Traversieren des Baumes zurücklegen. Bei Abbildung 1 (S. 10) würden beispielsweise alle Kanten vom Wurzelknoten STR_MATCH mit dem Merkmalswert „-“ in eine Kante zusammengefasst, respektive alle Kanten mit Merkmalswert „-“, die vom Knoten J_PRONOUN ausgehen usw.

Zu jedem Knoten im Baum wird dabei zusätzlich zum Merkmal die wahrscheinlichste Klassifikation entsprechend dem bisherigen Pfad abgespeichert. Für die Klassifikation einer unbekannten Instanz X heisst das, dass der Baum traversiert wird bis entweder in einem Blatt die Klassifikation einer Instanz *Y* mit identischen Merkmalswerten gefunden wird, oder bis zu dem Knoten, der den letzten identischen oder ähnlichsten Merkmalswert von X und *Y* enthält. Die wahrscheinlichste Klassifikation dieses Knoten dient dann als Klassifikation von X.

Durch *Pruning* kann der Entscheidungsbaum weiter komprimiert werden. Dabei werden im Entscheidungsbaum bestimmte Knoten und deren Verästelungen entfernt. Danach wird die Performanz des Entscheidungsbaums untersucht. Dieser Vorgang wird wiederholt und solange fortgeführt, bis eine Verschlechterung in der Performanz des Entscheidungsbaums auftritt. Der „geprunte“ Baum vor der ersten Verschlechterung der Performanz ist das Resultat des Prunings. So kann festgestellt werden, welche Merkmale keinen Einfluss auf die Klassifikation haben oder gar falsche Klassifikationen produzieren.

Cardie (1996) wies auf die Abhängigkeit der Performanz von ML-Verfahren von der jeweiligen Merkmalsmenge (*Feature Set*) hin. Anhand der Anaphernresolution von Relativpronomen wurde untersucht, inwiefern die Auswahl der Merkmale für die Merkmalsvektoren automatisiert werden kann. Dabei wurde eine manuell erstellte Repräsentation mit 33 Merkmalen in den Merkmalsvektoren benutzt, um einen C4.5-Entscheidungsbaum zu trainieren. Per Pruning wurden jene Merkmale eliminiert, die für die Klassifikation nicht relevant waren. Mit dem reduzierten Feature Set erreichte Cardie (1996, S. 4) in verschiedenen Experimenten eine Verbesserung des ML-Verfahrens von 4.9% bis 11.8%. Dabei wurden die Anzahl der Merkmale bis um die Hälfte reduziert. TiMBL adaptiert dieses Verfahren. Als Richtmass für das Pruning wird von TiMBL das Gain Ratio des jeweiligen Knoten, dividiert durch die Anzahl aller Merkmale, errechnet.

TiMBL verwendet also beim Lernen und Klassifizieren Methoden des *k*-NN-Algorithmus und kombiniert diese mit Elementen aus dem Entscheidungsbaum-Verfahren, um das Trainingskorpus im Speicher abzubilden. Dieser Entscheidungsbaum kann als Classifier gespeichert und unabhängig vom Trainingskorpus verwendet werden.

Nachdem die Mechanik von ML-Verfahren und insbesondere TiMBL betrachtet wurde, soll der Blick sich nun auf die Trainingskorpora richten, anhand derer Classifier trainiert werden und welche Problematiken sich dabei ergeben.

2.3 Die Bedeutung von unbalancierten Trainingskorpora

Das Trainingskorpus dient als Grundlage für das Erlernen eines Classifiers. Neben einer möglichst konsistenten Annotation, spielen, wie in Kapitel 2.1 erwähnt, die Merkmale, die zur Repräsentation des denotierten Wissen gewählt werden, und sinnvolle Masse für die Ähnlichkeiten der Merkmalswerte zentrale Rollen beim Erlernen eines leistungsstarken Classifiers. Darüber hinaus hat das Verhältnis zwischen positiven und negativen Beispielen im Trainingskorpus einen starken Einfluss auf das Lernen eines Classifiers. Die Bedeutung von unbalancierten Trainingskorpora wird im Folgenden nach Hoste (2005) beleuchtet.

Ein Problem bei der maschinellen Koreferenz-Auflösung ist, dass aufgrund der Art und Weise, wie mögliche Antezedens-Anaphern-Paare generiert werden, im Trainingskorpus mehr negative Beispiele (folglich Majoritätsklasse) enthalten sind als positive (Minoritätsklasse)[7]. Ein Classifier hat demnach mehr „Erfahrung“ im Klassifizieren von negativen Instanzen und erbringt entsprechend eine hohe Leistung. Die Klassifikation von nicht koreferenten Instanzen ist aber für die Aufgabe der Koreferenz-Auflösung nicht von Interesse.

Im ML wurde mit *Under-sampling* (Eliminierung von Beispielen aus der Majoritätsklasse) und *Over-sampling* (Duplikation von Beispielen aus der Minoritätsklasse) experimentiert, um asymmetrische Trainingskorpora zu balancieren. Dabei stellte sich heraus, dass Over-sampling kaum zu Verbesserungen führt. Ein Problem besteht darin, dass durch die Duplikation von Beispielen die Minoritätsklasse immer spezifischer auf die Merkmalswerte der duplizierten Beispiele abgestimmt wird, was bei der Klassifikation von neuen Instanzen mit abweichenden Merkmalswerten wenig hilfreich ist. Beim Under-sampling stellen sich Verbesserungen ein, wobei von den jeweiligen Experimenten nicht auf ein generell gültiges Verfahren oder Verhältnis zwischen negativen und positiven Beispielen geschlossen werden kann[8]. Andere Herangehensweisen sind die Vergabe von unterschiedlichen „Kosten“ für die Klassifikation von *False Negatives* (positive Instanzen, die als negativ klassifiziert werden) und *False Positives* (negative Instanzen, die als positiv klassifiziert werden)[9] oder die Gewichtung von Klassifikationen aufgrund eines ermittelten Schwierigkeitsgrads[10].

[7] Für Personalpronomen werden beispielsweise die drei vorangehenden Sätze betrachtet und alle NPs als mögliche Antezedenzien extrahiert, die in Person, Numerus und Genus mit der Anapher übereinstimmen. Für jedes mögliche Antezedens wird ein Paar mit der Anapher generiert, wobei nur eines davon ein positives Beispiel ist. Dadurch vergrössert sich das Gefälle zwischen positiven und negativen Instanzen fortlaufend.

[8] Hoste (2005, S. 126) gibt verschiedene Experimente an und bestätigt die Ergebnisse durch eigene Forschung.

[9] Siehe z.B. Ng (2004), Kapitel 2.6

[10] Vgl. Hoste (2005, S. 127ff)

Solche Techniken werden in Systemen zur Koreferenz-Auflösung kaum angewendet. Vielmehr wird versucht, die Generierung von negativen Beispielen zu verhindern, indem linguistisch motivierte, harte Filter angewendet werden, die unwahrscheinliche Antezedens-Anaphern-Paare eliminieren – so z.B. bei Strube *et al.* (2002); Hendrickx *et al.* (2007); Ailloud & Klenner (2009) – oder indem nur definite NPs betrachtet werden (Vieira & Poesio, 2000). Soon *et al.* (2001) generieren nur positive Beispiele zwischen Anaphern und deren unmittelbar vorangehenden, also nächsten, Antezedenzien. Die negativen Beispiele ergeben sich aus allen NPs zwischen diesen Antezedens-Anaphern-Paaren. Dadurch ergeben sich dennoch stark asymmetrische Verhältnisse: Nur 6.5% und 4.4% der Trainingsvektoren aus dem MUC-6-, respektive MUC-7-Korpus sind positive Beispiele. Ng & Cardie (2002a) folgten der Hypothese von Harabagiu *et al.* (2001), dass die am häufigsten auftretenden Koreferenz-Relationen die am einfachsten aufzulösenden sind (z.B. durch String Match, Bestimmung von Appositionen, Alias Match). Solche Koreferenzen werden gelöscht, um das System auf die schwierigeren Arten von Koreferenz-Relationen zu trainieren (Under-sampling der Minoritätsklasse).

Die Balancierung der Trainingsdaten gestaltet sich, wie gezeigt, schwierig. Dennoch können durch harte Filter viele Markables, die nicht in einer Koreferenzmenge sein können, gelöscht und die Generierung von Instanzpaaren, die aufgrund von linguistischen Merkmalen nicht koreferent sein können, verhindert werden. Durch solche Filter kann die Anzahl von False Positives und False Negatives verkleinert und das Gefälle zwischen negativen und positiven Beispielen im Trainingskorpus reduziert werden. Diese Filter werden konkret in Kapitel 2.6.2 betrachtet.

Nachdem die zentralen Voraussetzungen für die ML-basierte Auflösung von Koreferenz ermittelt sind, wird in den folgenden Kapiteln betrachtet, welche Merkmale für die Koreferenz-Auflösung in der bisherigen Forschung verwendet wurden und sich als relevant erwiesen haben.

2.4 Ermitteln relevanter Feature Sets für die Koreferenz-Auflösung

Aus den vorangehenden Kapiteln geht hervor, dass die Performanz eines Classifiers stark von den Merkmalen abhängt, die für die Repräsentation des zu speichernden Wissens gewählt werden. Ziel ist es, Merkmale zu finden, die die Entropie im Trainingskorpus

möglichst stark verringern; das heisst einerseits, die Instanzen im Trainingskorpus möglichst genau charakterisieren und andererseits, die Klassifikation von neuen Instanzen begünstigen. Diverse Feature Sets für die Anaphern- und Koreferenz-Auflösung werden im Folgenden chronologisch in Betracht gezogen und diskutiert.

2.4.1 Feature Sets in regelbasierten Ansätzen für pronominale Anaphernauflösung

Den ML-Verfahren zur Koreferenz-Auflösung geht eine lange Tradition von regelbasierten Systemen zur Anaphernauflösung voraus. ML-Verfahren adaptieren viele Merkmale aus diesen Systemen in ihre Merkmalsvektoren. Erste Ansätze zur pronominalen Anaphernauflösung betrachteten vor allem syntaktische und positionelle Merkmale.

Die viel zitierte Pionierarbeit von Hobbs (1976), der *Naive*-Algorithmus, berücksichtige – neben der Übereinstimmung von Genus, Numerus und Person – die grammatikalische Rolle (Subjekt, Objekt etc.) von Antezedens-Kandidaten und Pronomen. Mit diversen Heuristiken und Gewichten traversiert der Algorithmus die Syntaxbäume von relevanten Sätzen, um nach Antezedenzien zu suchen. Lappin & Leass (1994) verwendeten in ihrem System RAP (*Resolution of Anaphora Procedure*) nebst einem morphologischen Filter, der die Kongruenz von Genus-, Person- und Numerus überprüfte, verschiedene Salienzmerkmale von möglichen nominalen Antezedenzien. Grammatikalische Rolle, Parallelität der grammatikalischen Rollen eines möglichen Antezedens und der Anapher, Frequenz im Text und Distanz zur Anapher wurden für jeden Antezedens-Kandidaten festgehalten. Gemäss eines Regelwerks, das die Merkmalswerte unterschiedlich gewichtet, wurde dann ein Antezedens ausgewählt. So erhielten Kandidaten in Subjektposition beispielsweise eine höhere Gewichtung als Kandidaten in Objektposition etc. Lappin & Leass (1994, S. 542) hielten ausserdem fest, ob eine Nominalphrase (NP) definit oder indefinit ist und folgerten daraus, ob die NP *discourse-new* oder *discourse-old* ist. Der zugrundeliegende Gedanke ist, dass indefinite NPs neue Entitäten in einen Diskurs einführen. Diese Entitäten sind nicht anaphorisch und können folglich von der Suche nach Antezedenzien ausgeschlossen werden.

Mitkov (1998) präsentierte einen regelbasierten Ansatz mit „limited knowledge“. Dabei wurde das rechenintensive Parsen der Eingabe übersprungen und direkt mit der Ausgabe eines Part-of-Speech Taggers und eines Chunkers verfahren. Die grammatikalischen Rollen fielen demnach als Merkmale weg. Die Merkmale des Systems hielten – nebst der Definitheit – in Anlehnung an Ansätze aus der *Centering*- und *Focusing*-Theorie aus der Diskursanalyse

fest, ob eine NP das „Hauptthema“ eines Diskurssegments ist. Solche NPs erhalten eine höhere Gewichtung als mögliche Antezedens-Kandidaten, da es wahrscheinlich ist, dass sie in anaphorischen Beziehungen auftreten. Folgende Merkmale wurden verwendet um festzustellen, ob eine NP ein „Hauptthema“ ist:

- **Givenness**: Beschreibt, ob eine NP die erste NP in einem der Anapher vorangehenden Satz ist.
- **Indicating Verb**: Beschreibt, ob eine NP einem Verb aus einer gewissen Menge (*discuss, present, illustrate etc.*) folgt. Die Idee dabei ist, dass diese Verben die ihnen folgenden NPs semantisch aktivieren und fokussieren.
- **Lexical Reiteration**: Beschreibt, ob wörtliche Wiederholungen der NP vorkommen. Dabei werden auch wörtliche Wiederholungen des NP-Kopfs berücksichtigt.
- **Section Heading Preference**: Beschreibt, ob eine NP in einer Überschrift im Text vorkommt.
- **Non-prepositional Noun Phrase**: Beschreibt, ob eine NP ein PP-Attachment ist.
- **Collocation Pattern Preference / Immediate Reference**: Beschreibt, ob eine NP mit dem gleichen Verb wie die Anapher auftritt („Press the key [...] press it again“).
- **Referential Distance**: Gibt die positionelle Distanz zwischen NP und Anapher.
- **Term Preference**: Beschreibt, ob eine NP einem Term aus der Textdomäne entspricht.

Mitkov (1998) erzielte mit diesem „wissensarmen“ Ansatz ähnliche Resultate wie Lappin & Leass (1994). Diese hatten an Verfahren aus der Centering-Theorie bemängelt, dass durch die starke Gewichtung der aktivierten/fokussierten NPs vor allem intersententielle[11] Antezedenzien präferiert würden. Sie schätzen, dass aber nur rund 20% der Pronomen intersententiell aufgelöst werden müssen[12]. Mitkov gelang es – in der Kombination mit Distanz-Indikatoren – Ansätze aus der Centering-Theorie erfolgreich und rechentechnisch günstig für die pronominale Anaphernresolution zu nutzen.

[11] D.h. im Gegensatz zu *intra*sententieller Anaphora, dass das Antezedens nicht im selben Satz ist wie die Anapher.

[12] Dass diese Schätzung stark vom entsprechenden Korpus und Textgenre abhängt, zeigt eine Zählung der intersententiell aufgelösten Pronomen im *OntoNotes*-Korpus (s. Kapitel 4.1). Darin machen sie 48% aller anaphorischen Pronomen im Korpus aus.

Im späteren Ansatz MARS (Mitkov *et al.*, 2002) wurde der FDG-Parser (Tapanainen & Järvinen, 1997) hinzugezogen, um die Parallelität der grammatikalischen Rollen von möglichen Antezedenzien und Anaphern zu bestimmen. Ausserdem wurden die jeweils drei häufigsten NPs eines Texts zusätzlich gewichtet, da von Mitkov *et al.* (2002) empirisch festgestellt wurde, dass oft jeweils nur drei NPs in einem Text als Antezedenzien von Pronomen berücksichtigt werden müssen. Wesentlich war die Berücksichtigung der Belebtheit (*Animacy*) von NPs. Bei – speziell im Englischen – Nomen mit neutralem Genus (z.B. *Doctor*) kann nicht anhand des Genus festgelegt werden, ob sie sich auf belebte Pronomen mit spezifischem Genus (wie *he/she*) beziehen[13]. Mit diesen Veränderungen konnte die Leistung des ursprünglichen Ansatzes über diversen Korpora deutlich verbessert werden (Mitkov *et al.*, 2002, S. 12). Mitkov *et al.* demonstrierten zudem die Abhängigkeit der Performanz eines jeden Anaphernresolutions-Systems von der Qualität der Vorverarbeitung (*Preprocessing*). In einem Worst-Case-Szenario verringerte sich die Performanz von MARS um bis zu 25%[14] aufgrund von Parsing-Fehlern.

Weiter stellte Mitkov (1997) grundlegende Überlegungen zur Abhängigkeit der Merkmale untereinander an. Er stellte fest, dass es Merkmale gibt, zwischen denen eine stumme Abhängigkeit („mutual dependence“) besteht. Z.B. impliziert ein Merkmal x ein Merkmal y. Syntaktische und semantische Parallelität beispielsweise treten zwischen Antezedens und Anapher oft gemeinsam auf. Die Idee der stummen Abhängigkeit von Merkmalen wurde später in ML-Verfahren verwendet, um übergeordnete Merkmale zu definieren. Ng & Cardie (2002c) z.B. generierten Merkmale, die den Zusammenhang von Numerus- und Genus-Kongruenz festhielten.

Die positionellen (Distanz), syntaktischen (grammatikalische Rolle) und semantischen (Belebtheit, Fokussierung) Merkmale aus den regelbasierten Verfahren wurden weitgehend in die ML-basierten Verfahren zur Koreferenz- und Anaphernauflösung, die im Folgenden betrachtet werden, übernommen. Während der Entwicklung dieser Verfahren erhielten semantische Merkmale eine immer grössere Bedeutung, da nicht mehr nur pronominale, sondern zunehmend auch nominale Anaphern aufgelöst wurden.

[13] *Doctor* ist ein möglicher Antezedens-Kandidat für *he/she*, im Gegensatz zu unbelebten Nomen (z.B. *car*). Im Englischen wird Genus-Information von Nomen nicht in Artikeln kodiert und kann so nicht als diskriminierendes Merkmal verwendet werden. Nur durch die Bestimmung der Belebheit kann *car* als Antezedens von *he/she* ausgeschlossen werden.

[14] Mitkov verwendet das *Sucess Rate*-Mass zur Evaluierung. Dabei wird die Anzahl von einem System richtig aufgelöster Pronomen durch die Anzahl vorhandener Pronomen im Text dividiert. Heute wird dieses Mass als *Recall* bezeichnet und verwendet, s. Kapitel 3.

2.4.2 Feature Sets für Machine Learning-Verfahren zur Koreferenz-Auflösung

Eine der ersten Implementationen eines ML-Verfahrens für Koreferenz-Auflösung war diejenige von Aone & Bennett (1996). Sechs C4.5-Entscheidungsbäume wurden über einem Korpus aus japanischen Zeitungsartikeln trainiert. Semantische Merkmale waren in den Trainingsvektoren ein wesentlicher Bestandteil. Für alle NPs, die in anaphorischen Relationen auftreten konnten, hielten die Merkmalsvektoren – wo möglich – die Zugehörigkeit zu einer der fünf vorgegebenen semantischen Klassen (*Facility, Location, Organization, Person, Time*) fest. Insgesamt enthielten die Trainingsvektoren 66 Merkmale, die entweder unär (das Antezedens oder die Anapher einzeln betreffend) oder binär (die Relation zwischen Antezedens und Anapher betreffend) sein konnten (vgl. Tabelle 1). Die unären Merkmale stammten aus einem regelbasierten Anaphernresolutions-System, das gegen das ML-System evaluiert wurde. Bei der Evaluation schnitt das ML-System um bis zu 8% in F-Score[15] besser ab als das regelbasierte. Ausserdem erkannte das ML-System automatisch Typen von Anaphern (z.B. Reflexivpronomen), die im manuell erstellten Regelwerk nicht erfasst waren.

	unär	binär
Lexical	category	matching-category
Syntactic	topicalized	matching-topicalized
Semantic	semantic-class	subsuming-semantic-class
Positional		antecedent-precedes-anaphor

Tabelle 1: Beispiele von unären und binären Merkmalen bei Aone & Bennett (1996, S. 306)

Aone & Bennett (1996) zeigten auf, dass ML eine relevante Technik für die Anaphernresolution ist. Als Vorteil von ML-Systemen hoben sie hervor, dass Regeln automatisch besser aus einem Trainingskorpus gelernt werden, als dass sie manuell antizipiert werden können. Ausserdem betonten sie, dass die aufwändige Konzeption und Implementation eines Regelswerks bei ML-Verfahren wegfällt. Die Festlegung der fünf semantischen Klassen wurde, teilweise leicht modifiziert, in vielen ML-Systemen aufgegriffen, wobei die Autoren nicht klar beschrieben, welche semantischen Ressourcen oder Verfahren sie verwendeten, um einer NP eine relevante semantische Klasse zuzuweisen.

McCarthy & Lehnert (1995) benutzten ebenfalls Merkmale eines regelbasierten Systems um Merkmalsvektoren für das Training von C4.5-Entscheidungsbäumen zu generieren.

[15] Kapitel 3 gibt einen detaillierten Überblick über verschiedene Evaluationsmasse inklusive F-Score.

Das regelbasierte System wurde für die fünfte *Message Understanding Conference* (MUC-5)[16] implementiert, die sich mit Koreferenz-Auflösung mit dem Schwerpunkt Firmennamen beschäftigte. McCarthy & Lehnert (1995, S. 3) hielten fest, dass bei regelbasierten Systemen – nebst dem Finden adäquater Regeln – das Bestimmen einer Reihenfolge, in der die Regeln am effektivsten angewendet werden, ein fast unlösbares Problem darstellt. Deshalb sollte das ML-basierte System RESOLVE (McCarthy & Lehnert, 1995) diese Regeln selber ermitteln und adäquat kombinieren.

Zentrales Thema war bei diversen Systemen der MUC-5 die Minimierung der False Positives. Zugrunde lag die Idee, dass die Performanz eines Systems unter der Generierung von falschen Antezedens-Anapher-Paaren litt. Mit diversen Regeln versuchte man syntaktische und lexikalische Konstellationen festzuhalten, in denen NPs nie koreferent sein können (z.B. gibt es keine Nominalanapher, die ihr Antezedens c-kommandiert; unterschiedliche Firmennamen können nicht koreferent sein etc.). Für die Merkmalsvektoren wurden von McCarthy & Lehnert (1995) diesbezüglich vier unäre und vier binäre Merkmale bestimmt. Die unären hielten fest, ob und welche Art von Firmennamen Antezedens und Anapher enthielten. Die binären beschrieben, ob eine NP ein Alias oder ein *Substring* der anderen war und bei Mehrwort-NPs, ob eine NP Wörter der anderen enthielt. Ausserdem wurde in Anlehnung an die Regel, dass NPs, die vom gleichen Verb abhängen, nicht koreferent sein können, bestimmt, ob die NPs im gleichen Satz vorkamen. Grundsätzlich wurden die Merkmale in die Vektoren übernommen, anhand deren das regelbasierte System mögliche und unmögliche Koreferenz-Relationen bestimmte. Aus dem Korpus wurden 1230 Paare generiert, von denen 26% koreferent waren. Die Evaluation zeigte, dass RESOLVE das regelbasierte System um 7.6% in F-Score übertraf.

Cardie & Wagsta (1999) verfolgten einen ML-Ansatz, der ohne annotierte Beispiele im Trainingskorpus auskam (*unsupervised*). Sie nahmen das Problem der Koreferenz-Auflösung als Aufgabe für ein *Clustering*-Verfahren. Dabei spielen zwei Beobachtungen eine Rolle:

1. Koreferenz-Relationen sind symmetrisch, transitiv und reflexiv.[17]

2. Alle miteinander koreferenten NPs bilden eine Äquivalenzklasse. Alle NPs, die zu einer Äquivalenzklasse gehören und jeweils ein Aspekt des Konzepts der Klasse denotieren,

[16] Vgl. Sundheim (1993)

[17] Die Koreferenz-Relation gilt für A zu B wie für B zu A; wenn A und B, bzw. B und C koreferent sind, sind auch A und C koreferent; A ist mit A koreferent.

sind einander intuitiv „nahe“; bzw. ist die konzeptionelle Distanz zwischen diesen NPs „klein“.

Wenn es also möglich ist, anhand von Merkmalen der NPs Ähnlichkeiten bzw. Distanzen zu formulieren, können in einer Menge von Merkmalsvektoren mit Hilfe eines Radius r Äquivalenzklassen gebildet werden. Die Distanz zwischen zwei NPs wurde von Cardie & Wagsta (1999) dazu folgendermassen definiert:

$$dist(NP_i, NP_j) = \sum_{f \in F} w_f * incompatibility_f(NP_i, NP_j)$$

Distanz entspricht also der Summe der gewichteten Merkmale f, multipliziert mit einem Inkompatibilitätsfaktor. Der Inkompatibilitätsfaktor hing von vier Merkmalen (Numerus, Eigenname, semantische Klasse, Genus und Belebtheit) ab, die als Filter funktionierten und zur jeweiligen Anapher inkompatible Antezedens-Kandidaten eliminierten. Die verwendeten Merkmalsvektoren demonstrieren Cardie & Wagsta (1999, S. 84) anhand der NP „John Smith“ (Tabelle 2). Das Verfahren lag mit einem F-Score von 53.6% im Mittelfeld der in MUC-6 evaluierten Ansätze, von welchen die meisten regelbasiert waren. Gegenüber einer verbesserten Version von RESOLVE (47%) lag der Ansatz 6.6% weiter vorne.

NP (**Kopf**)	Position	Pronomen	Artikel	Numerus	Eigenname	Sem. Klasse	Genus	Animacy
John **Smith**	1	NONE	NONE	SING	YES	HUMAN	MASC	ANIM

Tabelle 2: Beispiel eines Merkmalvektors bei Cardie & Wagsta (1999, S. 84)

Vorteil des Clustering-Verfahrens ist, dass die mühselige und kostspielige manuelle Annotation von Koreferenz in einem Trainingskorpus wegfällt. Ein Nachteil ist, dass die Gewichtung der Merkmale – wie bei den regelbasierten Verfahren – manuell vorgenommen werden muss. Auch hängt der optimale Radius r von der Textstruktur ab und kann je nach Genre und Domäne stark variieren. Wie die Merkmalsgewichtung muss r manuell und empirisch ermittelt werden.

2.4.3 Das Standard-Feature Set von Soon *et al.*

Soon *et al.* (2001) berücksichtigten viele relevante Merkmale aus vorangegangenen Ansätzen und implementierten ein System, das von der späteren Forschung oft als Referenz und Baseline verwendet wurde. Das System wurde speziell auf die Auflösung nominaler Anaphora trainiert. Dazu wurde folgendes Feature Set für die Klassifikation der Koreferenz zwischen einem möglichen nominalen Antezedens (*i*) und einer nominaler Anapher (*j*) verwendet.

- **Distanz**: Misst die Distanz in Sätzen zwischen *i* und *j*.
- **String Match**: Hält fest, ob die Oberflächenformen von *i* und *j* identisch sind. Dabei werden die Artikel entfernt („a computer“ matcht „the computer“ etc.).
- **Definite Noun Phrase**: Hält fest, ob *j* definit ist (sprich *the* als Artikel hat).
- **Demonstrative Noun Phrase**: Hält fest, ob *j* einen Demonstrativ-Artikel hat (*this, that, these, those*).
- **Number Agreement**: Hält fest, ob *i* und *j* im Numerus übereinstimmen.
- **Gender Agreement**: Hält fest, ob *i* und *j* im Genus übereinstimmen. Der Genus von Eigennamen wird entweder anhand von Namenslisten oder anhand von *Designators* wie „Mrs.“ und „Mr.“ festgelegt. NPs wie „president“ oder „chief executive officer“ erhalten für das Genus-Merkmal den Wert „unknown“. Der Genus von Markables, die nicht der semantischen Klasse *Person* entsprechen, wird durch deren semantische Klasse (s. unten) bestimmt. NPs der semantischen Klasse *Object* erhalten den Wert „neutral“.
- **Both-Proper-Names**: Hält fest, ob *i* und *j* Eigennamen sind.
- **Alias**: Hält fest, ob *i* ein Alias von *j* ist – und umgekehrt. Die Alias-Überprüfung variiert dabei nach semantischer Klasse.
- **Appositive**: Hält fest, ob *j* eine Apposition zu *i* ist.
- **Semantic Class Agreement**: Hält fest, ob die semantischen Klassen von *i* und *j* identisch sind.

Für die Bestimmung der semantischen Klassen von *i* und *j* legen Soon *et al.* (2001) folgende Hauptkategorien fest: *Female, Male, Person, Organization, Location, Date, Time, Money, Percent, Object*. Diese Kategorien sind auf entsprechende WordNet-Synsets[18] abgebildet. *i* und *j* werden entsprechenden Synsets zugewiesen. Bei mehreren möglichen Synsets wird dasjenige mit der höchsten Wortfrequenz gewählt[19]. Sind die Synsets von *i* und *j* „Subklassen“[20] der selben Hauptkategorie, gehören sie zur selben semantischen Klasse.

[18] Eine detaillierte Beschreibung von WordNet gibt Kapitel 2.4.4.

[19] Wortfrequenzen wurden für WordNet durch Korpus-Recherchen ermittelt und sind in WordNet kodiert.

[20] Soon *et al.* (2001) definieren nicht, was eine „Subklasse“ in WordNet ist. Anhand ihres Beispiels (*chairman – person*) kann davon ausgegangen werden, dass sie die Hyponymie-Relationen zur Bestimmung der „Subklassen“ verwenden.

Das Verfahren wurde über den Testkorpora von MUC-6 und MUC-7 evaluiert. Dabei wurde ein F-Score von 62.2%, bzw. 60.4% erreicht. Auch wurde die Gewichtung der Merkmale bei der Generierung des Entscheidungsbaums[21] untersucht, respektive inwiefern die einzelnen Merkmale die Klassifikation eines möglichen Antezedens-Anaphern-Paares beeinflussten. Dabei stellten Soon *et al.* (2001, S. 535) fest, dass die binären Merkmale Distanz, Both-Proper-Names, Gender, Semantic Class und Number bei der Generierung des Entscheidungsbaums keinen Einfluss nahmen. Auch Baseline-Experimente, bei denen jeweils nur eines dieser Merkmale benutzt wurde, ergaben F-Scores von 0%. Der Entscheidungsbaum zeigte, dass gut zwei Drittel aller nominalen Anaphern über String Matching-Verfahren aufgelöst werden konnten. Diese Befunden suggerierten, dass semantische Klassen keinen Einfluss auf die Auflösung nominaler Anaphora haben. Gerade dieser Punkt wurde in der Forschung aufgegriffen und verfolgt. Ng (2007) argumentierte beispielsweise, dass die von Soon *et al.* verwendete Methode, bei mehreren möglichen Synsets aus WordNet jeweils dasjenige mit der höchsten Korpus-Frequenz zur Bestimmung der semantischen Klasse zu verwenden, nicht ausreichend sei.

Ng & Cardie (2002c) erweiterten das System von Soon *et al.* (2001) um Merkmale, die zusätzliches linguistisches Wissen kodierten. Die 41 neuen Merkmale basierten hauptsächlich auf den von Lappin & Leass (1994) aufgrund von linguistischer Intuition festgestellten, grammatikalischen Restriktionen, von denen Anaphorizität abhängt. Dabei wurden auch Meta-Merkmale verwendet, die z.B. festhielten, ob Genus und Numerus-Merkmale beide, nur eines oder keines übereinstimmten. Damit wurde dem Classifier ermöglicht z.B. Koreferenzen-Relationen zwischen Markables mit nicht kongruenten Numeri zu erkennen („the police [...] they“). Ng & Cardie (2002c, S. 6) verfeinerten und erweiterten die semantischen Merkmale für die Koreferenz-Relation zwischen zwei NPs (NP_i, NP_j) folgendermassen:

- **CLOSEST_COMP**: Wenn NP_i die nächste NP_j vorangehende NP ist, die gleiche sematische Klasse hat und alle linguistischen Restriktionen erfüllt sind.
- **SUBCLASS**: Wenn NP_i und NP_j verschiedene NP-Köpfe haben, aber in WordNet in einer Hyponymie-Relation sind.
- **WNDIST**: Wenn SUBCLASS erfüllt ist, hält WNDIST die Pfadlänge zwischen den Synsets von NP_i und NP_j fest.

Das System von Soon *et al.* (2001) wurde dupliziert und gegen das erweiterte Feature Set evaluiert. Dabei stellten Ng & Cardie (2002c) fest, dass bei der Verwendung aller neuen

[21] Vgl. Abbildung 1, S. 10.

Merkmale die Leistung deutlich zurückging. Hauptgrund für den Leistungsrückgang war die schlechte *Precision*[22] (40,1%) bei der Auflösung von nominalen Anaphern. Das Feature Set wurde darauf manuell auf 26 Merkmale reduziert, wobei jene Merkmale eliminiert wurden, welche die tiefe Precision bei den nominalen Anaphern verursacht hatten. Das verbesserte Feature Set produzierte schliesslich die bis anhin besten Resultate bei der Evaluation über den MUC-6 und MUC-7 Korpora.
Mit ihrer Arbeit zeigten Ng & Cardie (2002c), dass die Verwendung möglichst vieler Merkmale nicht unbedingt und automatisch zu einer Verbesserung in der ML-basierten Koreferenz-Auflösung führt, da die mitgegebenen Merkmale nicht per se automatisch in sinnvoller Weise in einen Entscheidungsbaum übersetzt werden können, der optimale Klassifikations-Schemata generiert.

Ng & Cardie (2002b) setzten ihre Forschung in Bezug auf die nominalen Anaphern fort. Eine mögliche Verbesserung lag laut Ng & Cardie (2002b) in der Bestimmung der Anaphorizität (*Anaphoricity Determination*) von NPs. Die Idee war, bevor mit der eigentlichen Koreferenz-Auflösung begonnen wird, jene NPs herauszufiltern, die aufgrund von lexikalischen, grammatikalischen, semantischen und positionellen Restriktionen als Antezedenzien für eine Anapher nicht in Frage kommen. Regelbasierte Ansätze (Lappin & Leass, 1994) und ML-Verfahren (Soon *et al.*, 2001) berücksichtigen solche Restriktionen ansatzweise, indem z.B. Letztere die Definitheit von NPs in den Merkmalsvektoren kodierten. Die Annahme war, dass indefinite NPs meistens nicht anaphorisch sind. Andere Ansätze gingen laut Ng & Cardie (2002b) entweder davon aus, dass alle anaphorischen NPs bereits vor der Koreferenz-Auflösung bekannt sind (Harabagiu *et al.*, 2001), oder es wurden nur definite NPs betrachtet (Vieira & Poesio, 2000).
Nun implementierten Ng & Cardie (2002b) ein auf C4.5 basierendes ML-System, das mit 53 Merkmalen die Anaphorizität von NPs bestimmte und das im Preprocessing für ein früheres ML-System zur Koreferenz-Auflösung (Ng & Cardie, 2002c) verwendet wurde. Die Feature Sets waren teilweise identisch (String Match, Definitheit, Numerus etc.). Die Anaphoricity Determination hielt ausserdem fest, ob eine NP durch einen Relativsatz oder eine Apposition postmodifiziert wurde, welchen syntaktischen Mustern eine definite NP entsprach oder ob die NP im Titel, ersten Paragraphen oder ersten Satz eines Texts enthalten war etc. Wieder zeigte sich durch Pruning, dass nur maximal 15 Merkmale relevant waren und in den Entscheidungsbaum projiziert wurden. Die Merkmale HEAD_MATCH und ALIAS waren

[22] *Precision* (Präzision) ergibt sich aus der Anzahl richtiger Klassifikationen dividiert durch die Anzahl vorgenommener Klassifikationen (s. Kapitel 3).

dabei die wichtigsten. Das heisst, wenn zwei NPs mit identischem Kopf auftreten, ist die Wahrscheinlichkeit gross, dass sie koreferent sind; ebenso wenn eine NP ein Alias einer anderen ist.

Die Evaluation ergab gemischte Resultate, vor allem in Bezug auf die Nomen. Precision konnte erhöht werden, *Recall*[23] ging aber im Vergleich zum ML-System ohne Bestimmung von Anaphorizität zurück, sodass der F-Score stark sank. Mit einer Regel wurde festgelegt, dass NP-Paare, die die Merkmale HEAD-, bzw. STRING_MATCH oder ALIAS erfüllen, nicht durch den Anaphorizitäts-Filter gingen. Mit dieser Massnahme konnte Recall aufgeholt werden (Tabelle 3). Das Baseline-Experiment beinhaltete das Koreferenz-Auflösungssystem, das keine Information über Anaphorizität hatte. *Anaphor (No Constraints)* war das System mit Bestimmung von Anaphorizität und *Anaphor (Constraints)* das System, bei dem die Merkmale STRING_MATCH und ALIAS manuell als Überbrückung des Anaphorizitäts-Test geschaltet wurden.

Common nouns	*MUC-6*			*MUC-7*		
	R	P	F	R	P	F
Baseline	**25.2**	40.1	**31.0**	**26.6**	45.2	**33.5**
Anaphor (No Constraints)	15.4	**56.2**	24.2	13.8	**77.5**	23.4
Anaphor (Constraints)	20.5	53.1	29.6	21.7	59.0	31.7

Tabelle 3: Resultate für Koreferenz-Auflösung von Nomen in MUC-6 und MUC-7 nach Ng & Cardie (2002b, S. 6)

Im Gegensatz zu Pronomen sind Nomen seltener anaphorisch (In den MUC-6 und MUC-7 Korpora sind 77% bzw. 78% der Nomen nicht anaphorisch[24]). Ng & Cardie (2002b) errechneten, dass wenn von den Nomen bekannt wäre, ob sie koreferent sind, das System einen F-Score von 73.1%, respektive 70.7% für MUC-6 und MUC-7 produzieren würde. Die Bestimmung von Anaphorizität ist folglich zentral für die Leistung eines Koreferenz-Auflösungssystems. Die Experimente von Ng & Cardie (2002b) zeigten, dass ML dafür nur bedingt verwendbar ist. Spätere Systeme greifen auf Regelwerke zurück um Ausschlussverfahren für nicht anaphorische NPs festzuhalten.

[23] *Recall* (Ausbeute) ergibt sich aus der Anzahl von einem System richtig klassifizierter Instanzen dividiert durch die Anzahl aller vorhandener Instanzen (s. Kapitel 3).

[24] Eine Analyse des OntoNotes-Korpus ergibt ein ähnliches Resultat, s.Kapitel 4.1

Strube *et al.* (2002) führten ein wichtiges Merkmal in das Feature Set von Soon *et al.* (2001) ein: die *Minimum Edit Distance* oder Levenshtein-Distanz (Levenshtein, 1966). Dabei wird gezählt, wie viele Zeichen ersetzt, gelöscht und eingesetzt werden müssen, um eine Zeichenkette in eine andere zu transformieren. Auch Strube *et al.* (2002) bestätigten, dass die Auflösung von Koreferenz-Relationen, die Named Entities (NEs) und definite NPs enthalten, besonders schwierig ist. Sie führten daher Experimente durch, bei denen sich das Korpus jeweils auf eine Art von Anapher beschränkte (Pronomen, Nomen, NEs). Die insgesamt moderate Leistung des Systems ergab sich durch die schlechte Performanz bei der Auflösung von definiten NPs. Um die Anzahl von negativen Beispielen im Trainingskorpus[25] zu minimieren, wurden harte Filter implementiert, die Paare eliminierten, die folgenden Kriterien entsprachen (Strube *et al.*, 2002, S. 3):

- Die Anapher ist eine indefinite NP.
- Wenn eine Instanz in die andere eingebettet ist, also z.B. das Antezedens Kopf der Anapher ist (und umgekehrt).
- Die Instanzen haben unterschiedliche semantische Klassen.
- Wenn eine der beiden Instanzen nicht den Merkmalen 3. Person Singular oder Plural entspricht.
- Bei anaphorischen Pronomen: Person, Numerus und Genus stimmen nicht überein.

Mit dieser Filterung konnten die negativen Beispiele im Trainingskorpus um 50% reduziert werden. Das Feature Set wurde weitgehend von Soon *et al.* (2001) übernommen und generierte in den Experimenten eine Precision von 88.60% und einen Recall von 45.32% (F-Score 59.97%). Die separaten Testläufe über den einzelnen Arten von Anaphern zeigten, dass die Leistung durch die Ergebnisse der Auflösung von definiten NPs (Recall 8.71%) beeinträchtigt wurde. Mit der Integration der Levenshtein-Distanz in die Merkmalsvektoren konnten die Resultate insgesamt um 8% auf einen F-Score von 67.98% verbessert werden. Das entsprach einer Verbesserung von 18% bei den definiten NPs. Ausgehend von den Beobachtungen von Soon *et al.* (2001), dass ein Grossteil der nominalen Anaphern per String Match-Verfahren aufgelöst werden können, war die Einführung der Levenshtein-Distanz – einem Merkmal, dass anhand der Strings der Markables operiert – sinnvoll und erfolgreich.

[25] Strube *et al.* (2002) verwendeten ein hausinternes Korpus in deutscher Sprache

Versley (2006) implementierte ein System mit harten und weichen Filtern nach Strube *et al.* (2002). Bei der Auflösung von nominalen Anaphern, die eine NE als Antezedens haben, werden als semantische Features die Übereinstimmung der semantischen Klasse und die Pfadlänge im deutschen Wortnetz *GermaNet* (Hamp & Feldweg, 1997) anhand der Hyponymie-Relationen zwischen zwei Markables festgehalten. Interessanterweise stellt Versley (2006, S. 5) fest, dass die beiden Merkmale alleine keine nominalen Anaphern auflösen, aber gemeinsam im Feature Set eine Verbesserung der Leistung erzielen. Ausserdem hält Versley (2006) fest, dass innerhalb der Auflösung nominaler Anaphora diejenigen Konstellationen am schwierigsten aufzulösen sind, in denen Antezedens und Anapher nicht den gleichen lexikalischen Kopf haben. Sie können nicht über String Matching-Verfahren gefunden werden und brauchen zusätzliches Weltwissen zur Auflösung (sog. *Coreferent Bridging*; z.B. „George W. Bush [...] the president“).

Versley (2007a) beschreibt neben dem Eliminieren von Antezedenzien mit unpassender semantischer Klasse und dem Bestimmen von *discourse new/old*[26]-Merkmalen zusätzliche semantische Indikatoren, die als Merkmale bei der Auflösung von Bridging Anaphora verwendet werden können. Dazu gehört die Bestimmung semantischer Ähnlichkeit. Versley (2007a) teilt die Verfahren zum Auffinden semantischer Ähnlichkeit in zwei Methoden auf.

1. Verfahren, die Wortnetze und deren Relationen (Hyponymie, Synonymie etc.) und darin gefundene (und teils gewichtete) Pfade verwenden, so z.B. Cardie & Wagsta (1999); Harabagiu *et al.* (2001)

2. Verfahren, die grosse, nicht annotierte Korpora (z.B. das World Wide Web) verwenden, um durch Bestimmung von Kollokation semantische Verwandtschaft zu ermitteln, z.B. Gasperin & Vieira (2004)

Zu den Vorteilen der ersten Methode zählt Versley (2007a) die genaue und eindeutige Bestimmung von Relationen durch die Wortnetze. Ein Nachteil ist die Abgeschlossenheit und teilweise geringe Abdeckung von Terminologien aus verschiedenen Domänen. Auch werden Named Entities gar nicht oder nur ansatzweise erfasst. Vorteil der korpusbasierten Methode ist das Auffinden von Assoziationen, die nicht in einem Wortnetz erfasst sind. Mit Suchmustern wie *X and other Ys* können in der Korpus-Recherche Hyponymie-Relationen gefunden werden oder Named Entities auf Nomen abgebildet werden. Die gefundenen Kollokationen sind aber nicht immer hilfreich für die Auflösung von Koreferenz. Im TüBa-Korpus (Telljohann *et al.*,

[26] Vgl. Kapitel 2.6.1

2004) werden z.B. durch das Lin-Mass[27] für den Begriff „Land“ die Begriffe „Staat, Stadt, Region, Bundesrepublik, Republik“ gefunden, während das Assoziationsmass nach Garera & Yarowsky (2006) die Begriffe „Regierung, Präsident, Dollar, Albanien, Hauptstadt“ findet. Solche assoziativen Begriffe, die intuitiv semantisch weit entfernt sind, aber in Korpora häufig zusammen vorkommen, sind für die Koreferenz-Auflösung nicht hilfreich.

Im System von Versley (2007a) wurde ein Merkmal implementiert, das Pfadlängen zwischen Markables[28] misst und ein Merkmal, das die Übereinstimmung der semantischen Klasse der Markables festhält, implementiert. Als weiterer Indikator wurde das Verfahren von Markert & Nissim (2005) implementiert, das Korpora – das World Wide Web und das BNC-Korpus (Francis & Kucera, 1979) – nach *other*-Mustern durchsucht, um eine mögliche Hyponymie-Relation zwischen Antezedens-Kandidat und Anapher zu finden. Zudem wurden zu allen Markables anhand des Lin-Masses Listen mit den ähnlichsten Wörtern erstellt und überprüft, ob ein möglicher Antezedens und die Anapher jeweils in der entsprechenden Liste auftreten, oder ob die beiden Listen gemeinsame Einträge haben. Die Evaluation der Feature Sets über dem TüBa-Korpus ergab folgende Resultate (Versley, 2007a, S. 500):

- Antezedenzien mit identischem Kopf wie die Anaphern können relativ einfach mit String Match-Techniken aufgelöst werden (Recall 49.8% und Precision 86.5% bzgl. aller nominalen Anaphern).
- Wenn bei den Bridging Coreferences jeweils das positionell nächste mögliche Antezedens gewählt wird (nur mit Numerus-Übereinstimmung), wird eine Precision von 12% erreicht.
- Wenn die Übereinstimmung der semantischen Klassen als harter Filter operiert und der jeweils nächst mögliche Antezedens gewählt wird, verbessert sich Precision auf 35% und Recall entspricht 61.1%.
- Wenn die Hyponymie-Relation aus GermaNet als Merkmal kodiert wird, erhöht sich die Precision auf 57.7% und Recall 67%.
- Die Verwendung von *other*-Mustern ergibt eine Precision von 55% und einen Recall von 54.3%.

[27] Vgl. Kapitel 2.4.4

[28] Versley (2007a) geht davon aus, dass alle Diskurs neuen Markables und somit alle *True Mentions* (d.h. alle Markables, die in einer Koreferenzmenge sind) bekannt sind und operiert nur über diesen.

Grundsätzlich liefern *other*-Muster zur Bestimmung von Hyponymie-Relationen eine schlechtere Leistung als Wortnetze. Versley (2007b) stellt dennoch fest, dass die *other*-Muster in den Fällen, in denen in GermaNet keine Hyponymie-Relation gefunden wird, hilfreich sein können – wenn auch nur in beschränktem Mass.

Ausgehend von den Befunden von Versley (2006) und Ponzetto & Strube (2006), dass semantische Ähnlichkeitsmasse relevante Merkmale für die Auflösung von Bridging Coreference sind, werden im nächsten Kapitel verschiedene Masse vorgestellt und diskutiert, die anhand von WordNet errechnet werden können. Die Masse werden in Kapitel 4 in einem System implementiert und evaluiert. Versley (2007a) stellte fest, dass semantische Merkmale (Ähnlichkeitmasse, Klassen, Relationen) erfolgreich als harte Filter genutzt werden können. In der Evaluation in Kapitel 5.3 wird diese Erkenntnis experimentell überprüft und weiter untersucht.

2.4.4 Masse zur Bestimmung semantischer Ähnlichkeit in WordNet

WordNet (Fellbaum, 1998) ist eine viel genutzte Ressource in verschiedensten Gebieten von NLP-Anwendungen. Synonyme Wörter, die jeweils identische Konzepte denotieren, werden in sogenannten *Synsets* zusammengefasst und mit einer Kurzbeschreibung (*Gloss*) – ähnlich wie in einem herkömmlichen Wörterbuch – versehen. WordNet kann insofern als Graph oder Baum verstanden werden, als dass zwischen den Synsets verschiedene Kanten bestehen, für die jeweils verschiedene semantische Beziehungen kodiert sind. So werden beispielsweise Hyponymie (*is a*-Relation), und Meronymie (*has part*-Relation) etc. zwischen Synsets festgehalten (Vgl. Abbildung 2).

Für die Messung von Distanz oder Ähnlichkeit zweier Wörter in WordNet werden Algorithmen aus der Graphen-Theorie adaptiert. Dabei wird die Hyponymie-Hierarchie von WordNet verwendet, die als Baum, bzw. Graph interpretiert wird. Pedersen *et al.* (2004) haben ein frei verfügbares Werkzeug – *WordNet::Similarity* – entwickelt, das verschiedene Ähnlichkeitsmasse zwischen zwei Konzepten in WordNet berechnet. Grundsätzlich können die Masse in zwei Kategorien aufgeteilt werden: In jene, die auf der Pfadlänge zwischen zwei Konzepten basieren und jene, die nebst den Pfadlängen den sogenannten *Information Content* des *Lowest Common Subsumer* (LCS) in die Berechnung miteinbeziehen. Im Folgenden werden die Distanz-Masse gemäss Patwardhan (2003) eingeführt und Vor- und Nachteile diskutiert.

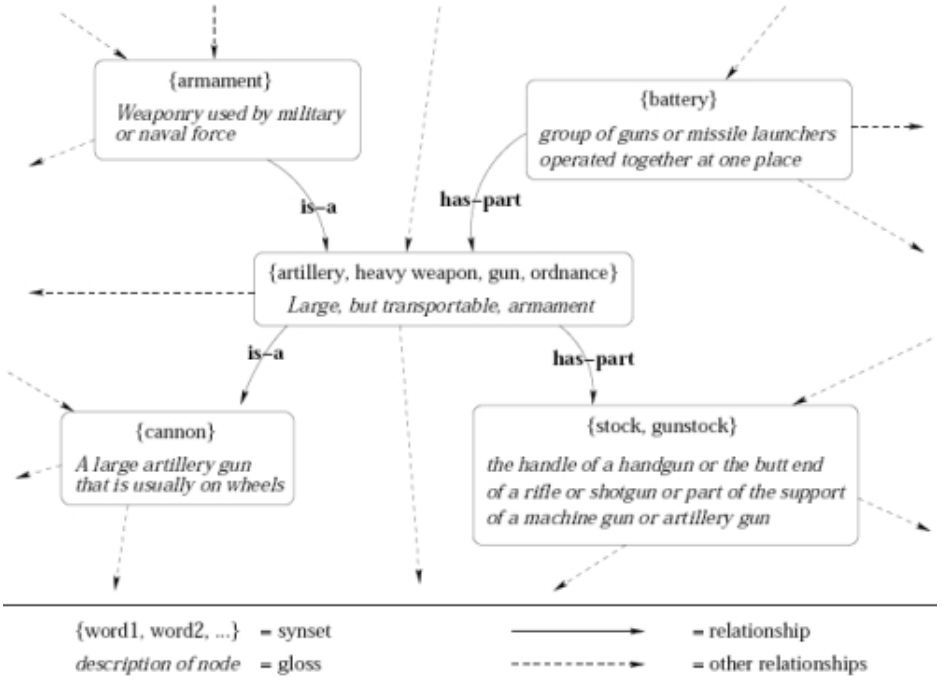

Abbildung 2: Illustration von Synsets und deren Relationen in WordNet gemäss Patwardhan (2003, S. 10)

Das Path-Mass kann als Baseline für die anderen Masse verstanden werden. Die Anzahl der traversierten Kanten entlang des kürzesten Pfads zwischen zwei Knoten wird gezählt und der inverse Wert zurückgeliefert. Nachteil dieses Masses ist die fehlende Komplexität in Bezug auf die vom Menschen empfundene Ungleichmässigkeit der Distanzen relativ zur Tiefe des Baumes. Menschen bewerten die Distanzen zwischen Wurzeln und Blättern in tieferen Regionen der Hierarchie als kleiner, als jene zwischen den immer abstrakter werdenden Konzepten in den oberen Regionen der Hierarchie. Die Distanz zwischen dem Synset „Sprinkler“ und „Mechanical device“ beispielsweise wird intuitiv als kleiner bewertet, als jene zwischen den Synsets „Cage“ und „Artefact“. Dennoch ist ihre Distanz anhand der in WordNet traversierten Kanten gleich.

Um diese Ungleichmässigkeit zu beheben, skalieren Leacock & Chodorow (1998) die Distanz in Anzahl Kanten zwischen zwei Knoten mit der maximalen Tiefe des Baums. Das Mass wird folgendermassen formalisiert:

$$related_{lch}(c_1, c_2) = -\log\left(\frac{shortestpath(c_1, c_2)}{2 \cdot D}\right)$$

Dabei entsprechen c_1 und c_2 den beiden Knoten bzw. den Synsets, $shortestpath(c_1, c_2)$ repräsentiert die Länge des kürzesten Pfads zwischen c_1 und c_2 und D gibt die maximalen Tiefe des Baums wieder. Damit wird die Tiefe des Baums zwar miteinbezogen; die ungleichmässige Wahrnehmung von identischen Pfadlängen in den oberen und unteren Regionen wird aber noch nicht berücksichtigt. Laut Patwardhan (2003, S. 11) erzielt dieser Ansatz dennoch relativ gute Resultate.

Wu & Palmer (1994) vernachlässigen bei der Berechnung ihres Ähnlichkeitsmasses zwischen zwei Konzepten den kürzesten Pfad. Stattdessen verwenden sie die jeweilige Distanz zum sogenannten Lowest Common Subsumer (LCS). LCS repräsentiert den im Baum am tiefsten gelegenen und damit spezifischsten Knoten, mit dem jeweils beide Konzepte in einer Hyponymie-Relation stehen. Um die Tiefe des LCS im Baum zu berücksichtigen, wird die Formel mit der Distanz zwischen dem LCS und dem Wurzelknoten des Baums skaliert:

$$ConSim(C1, C2) = \frac{2 \cdot N3}{N1 + N2 + 2 \cdot N3}$$

$N1$ repräsentiert dabei die Anzahl der Knoten auf dem Pfad von $C1$ zum LCS, respektive ist $N2$ die Anzahl Knoten auf dem Pfad von $C2$ zum LCS. $N3$ bezeichnet die Distanz des LCS zum Wurzelknoten. Dieser Ansatz wurde verwendet, um ein maschinelles Übersetzungssystem zu erweitern, das englische Verben ins Chinesische übersetzt. Dabei wurden den Argumenten der Verben Konzepte aus einer Taxonomie zugewiesen. Durch die Berechnung der Ähnlichkeiten der Verb-Argumente konnte die Leistung des Systems laut Wu & Palmer (1994, S. 138) um bis zu 57% verbessert werden.

Resnik (1995) führte für die Berechnung semantischer Ähnlichkeit den sogenannten *Information Content* (IC) ein. Dieses Mass gibt bezogen auf ein Textkorpus an, wie spezifisch oder generell ein darin vorkommendes Konzept ist. Dabei wird im Korpus nicht nur das Vorkommen eines Konzepts selbst gezählt. Bei jedem Vorkommen eines Konzepts werden die Häufigkeiten aller Konzepte inkrementiert, die es durch eine Hyponymie-Relation subsumieren. Das bedeutet: Je weiter oben in der Hierarchie, also je abstrakter ein Konzept ist, bzw. je mehr Konzepte es subsumiert, desto häufiger wird es tendenziell gezählt. Der Wurzelknoten wird demzufolge bei jedem Vorkommen eines beliebigen Konzepts gezählt. Der Information Content wird für ein Konzept c anhand seiner Frequenz in Bezug auf die Frequenz der übergeordneten Knoten $root$ berechnet:

$$IC(c) = -\log\left(\frac{freq(c)}{freq(root)}\right)$$

Resnik verwendet für die Berechnung der Ähnlichkeit zweier Konzepte ebenfalls den LCS. Er definiert Ähnlichkeit nun als den Information Content des LCS zweier Konzepte:

$$related_{res}(c_1, c_2) = IC(lcs(c_1, c_2))$$

Dabei bezeichnet IC den Information Content des LCS und $lcs(c_1, c_2)$ den LCS zweier Konzepte. Das Resnik-Mass verwendet also Frequenzen von Konzepten in einem grossen Textkorpus und berücksichtigt die Hierarchie einer Taxonomie. Die Pfadlängen der jeweiligen Konzepte zum LCS werden vernachlässigt.

Jiang & Conrath (1997) beziehen, nebst dem Information Content des LCS, diejenigen der von ihm jeweils subsumierten Konzepte in die Berechnung mit ein. Die Vernachlässigung der Pfadlängen der Konzepte zu ihrem LCS bei Resnik wird behoben, da sie in die Berechnung ihres jeweiligen Information Content einfliessen:

$$distance_{jcn}(c_1, c_2) = IC(c_1) + IC(c_2) - IC(2 \cdot lcs(c_1, c_2))$$

Auch wurden Experimente mit diversen Parametern für die Tiefe und die lokale Dichte des Baums durchgeführt. Dabei konnten laut Jiang & Conrath (1997, S. 11) keine grösseren Verbesserungen erzielt werden. Die Berechnung des Masses ist so angelegt, dass die erhaltene Distanz die Verschiedenheit zweier Konzepte beschreibt. Pedersen *et al.* (2004) verwenden daher den inversen Wert um ein Ähnlichkeitsmass zu erhalten.

Lin (1998) übernahm die Erkenntnisse von Resnik (1995) und formulierte ein Distanzmass basierend auf dem *Dice Coefficient*, einem Ähnlichkeitsmass, das im Information Retrieval angewendet wird. Der Dice Coefficient kann beispielsweise als Ähnlichkeitsmass für Zeichenketten verwendet werden. Dabei werden anhand folgender Formel Bigramme von Strings verglichen:

$$similar_{dice}(x, y) = \frac{2n_t}{n_x + n_y}$$

Dabei entspricht n_t der Anzahl der Bigramme beider Strings und n_x, respektive n_y jeweils der Anzahl der Bigramme in x und y. Lin verband Information Content (IC) mit dem LCS (lcs) und setzte sie in die Formel des Dice Coefficient ein:

$$related_{lin}(c_1, c_2) = \frac{2 \cdot IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)}$$

Er verglich den Ansatz mit denjenigen von Wu & Palmer (1994) und Resnik (1995) über einem Testset von Wortpaaren, deren Ähnlichkeit von Menschen beurteilt worden war (Miller

& Charles, 1991). Dabei wurde ein Wert von 0.834 in Bezug auf den Höchstwert 1 erreicht, was gegenüber Resnik (1995) eine Verbesserung von 0.039 und gegenüber Wu & Palmer (1994) eine Verbesserung von 0.031 bedeutete.

Hirst & St-Onge (1997) schufen für ihr Ähnlichkeitsmass drei Hauptkategorien: *extra-strong*, *strong* und *medium-strong*. Die Relation *extra-strong* wird an Wortpaare vergeben, deren Oberflächenform identisch ist, also an literale Wiederholungen („man [...] man“). Die *strong*-Relation ist in drei Subkategorien aufgeteilt:

1. Die beiden Wörter sind im gleichen Synset („car“ – „automobile“)
2. Zwischen den Synsets der Wörter in WordNet besteht eine horizontale Verbindung (z.B. *Antonymy*, *Similarity* oder *See also*; „hot“ – „cold“)
3. Ein Wort ist Teil des anderen und zwischen den Synsets beider Wörter besteht eine Hyponymie-Relation („school“ – „private school“)

Bei der *medium-strong*-Kategorie wird in WordNet nach vordefinierten, erlaubten Pfaden zwischen den Synsets zweier Wörter gesucht. Solche Pfade müssen zwischen zwei und fünf Kanten lang sein und vorgegebenen Richtungsänderungen beim Traversieren der Knoten entsprechen. Die Richtungsvorgaben verhindern beispielsweise, dass ein Pfad über einen gemeinsamen Tochterknoten zweier Synsets gefunden wird. Die Kategorien *extra-strong* und *strong* haben fixe Gewichte. *medium-strong*-Relationen werden anhand einer Konstante, der Pfadlänge und Anzahl Richtungswechsel gewichtet:

$$weight = C - pathlength - k \times numberofchangesofdirection$$

WordNet::Similarity vernachlässigt die Kategorie *extra-strong*, da sie anhand der Oberflächenformen der Wörter operiert. Für die Konstanten C und k werden Hirst und St-Onge entsprechend die Werte 8 bzw. 1 vergeben.

Lesk (1986) entwarf als Mass zur Disambiguierung von Wortbedeutungen den sogenannten *Extended Gloss Overlap*. Der Ansatz nimmt die Wörter aus einer Wörterbuch-Definition und vergleicht sie mit den Wörtern, die im Kontext des ambigen Worts vorkommen. Dann wird diejenige Wortbedeutung gewählt, bei der mehr Wörter aus der Wörterbuch-Definition mit den Wörtern des Kontexts übereinstimmen.
Der Lesk-Algorithmus wurde von Banerjee & Pedersen (2002) für WordNet adaptiert. Anstelle von Wörterbuch-Definitionen verwendeten sie die WordNet-Glossen, die jedem

Synset zugewiesen sind. WordNet-Glossen enthalten eine Definition und oft ein Verwendungsbeispiel für die Wörter im Synset. Der Ansatz berücksichtigt eine Vielzahl von Relationen in WordNet: In einem ersten Schritt werden die Kontextwörter eines ambigen Worts ermittelt. Dann werden alle Synsets inklusive deren Glossen ermittelt, mit denen die Kontextwörter in WordNet in einer Relation stehen. Zur Disambiguierung wird nun die Übereinstimmung der Glossen verglichen. Diejenige Bedeutung wird dann gewählt, bei der die Glossen der Kontextwörter – und die Glossen der Wörter, die mit den Kontextwörtern in einer taxonomischen Relation stehen – mit der Glosse des ambigen Worts am meisten übereinstimmen. Die Übereinstimmung wird dabei anhand von String Matching ermittelt.

Pedersen *et al.* (2004) verwenden dieses Disambiguierungs-Verfahren nun als Ähnlichkeitsmass. Patwardhan (2003, S. 18) veranschaulicht das Verfahren nochmals detaillierter:
Man nehme ein Konzept c_1 und eine Menge von Glossen C_1, die anfänglich die Glosse von c_1 enthält. Dann werden jeweils alle Glossen der Wörter, die mit c_1 in einer Relation r stehen, konkateniert und C_1 beigefügt. C_1 enthält schliesslich die Glosse von c_1 und die jeweils konkatenierten Glossen pro Relation r. In gleicher Weise wird C_2 für c_2 erstellt. Per String Matching wird dann die Ähnlichkeit von C_1 und C_2 mit den Kontextwörtern des zu disambiguierenden Worts berechnet, wobei bei Mehrwort-Übereinstimmungen das Quadrat der Anzahl übereinstimmender Wörter zum Ähnlichkeitswert addiert wird (z.B. „space shuttle“ $\rightarrow$ 4). Das Mass des Extended Gloss Overlaps beinhaltet allerdings zwei Nachteile:

1. Gleichheit wird durch restriktives String Matching ermittelt. Numerus wird dadurch zu einem diskriminierenden Faktor („spoon“ – „spoons“ gilt demnach nicht als Übereinstimmung).

2. Konzeptuelle Übereinstimmungen von Wörtern wie „spoon“ – „silverware“ in den Glossen werden gar nicht berücksichtigt.

Die Erkenntnisse von Jiang & Conrath (1997) zeigen, dass die Verwendung von Kontextwörtern aus einem grösseren Textkorpus die Berechnung von Ähnlichkeit begünstigt. Pedersen *et al.* (2004) nehmen diesen Ansatz auf und verknüpfen ihn mit der Verwendung der WordNet-Glossen bei Banerjee & Pedersen (2002). Die Idee ist, die WordNet-Glossen selbst als Korpus zur Generierung von Kontextvektoren zu verwenden.

Pedersen *et al.* (2004) folgen Schütze (1998) beim Erstellen der Kontextvektoren. Dabei wird zuerst ein sogenannter *Word Space* erstellt. Dieser besteht aus allen Inhaltswörtern der

WordNet-Glossen (Substantive, Verben etc.), die bei einer Filterung eine gewisse Signifikanz[29] aufweisen. Funktionswörter (Präpositionen, Verbpartikel etc.) werden anhand von Stopplisten eliminiert. Anschliessend werden für die Wörter aus dem Word Space Wortvektoren generiert. Dabei werden alle Vorkommnisse eines Worts im Korpus gefunden und deren Kontexte betrachtet. In diesen Kontexten wird dann nach Inhaltswörtern gesucht. Diejenigen Dimensionen des Wortvektors, die die gefundenen Inhaltswörter kodieren, werden inkrementiert. So entsteht für jedes Inhaltswort ein Wortvektor, der die Kollokation mit den anderen Inhaltswörtern des Word Space beschreibt. Pedersen *et al.* (2004) nehmen nun als Kontexe für die Suche benachbarter Inhaltswörter die WordNet-Glossen.
Sind die Wortvektoren erstellt, können daraus die sogenannten *Gloss Context Vectors* generiert werden. Dabei wird jedem Inhaltswort der Glosse eines Konzepts der entsprechende Wortvektor zugewiesen. Patwardhan (2003, S. 24) fügt als Beispiel den Gloss Context-Vektor von „lamp“ an. Die WordNet-Glosse von „lamp“ ist „an artificial source of visible illumination“. Folglich besteht der Gloss Context-Vektor von „lamp“ aus den Wortvektoren von „artificial“, „source“, „visible“ und „illumination“. Die Ähnlichkeit zweier Konzepte entspricht nun dem Kosinuswert des Winkels zwischen den beiden Gloss Context-Vektoren der Konzepte:

$$related_{vector}(c_1, c_2) = \cos(angle(\vec{v}_1, \vec{v}_2))$$

Dabei entsprechen $\vec{v}_1$ und $\vec{v}_1$ den Gloss Context-Vektoren. Durch die Verwendung des Vektorprodukts kann das Ähnlichkeitsmass folgendermassen formuliert werden:

$$related_{vector}(c_1, c_2) = \frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1||\vec{v}_2|}$$

Einen Überblick über Werte der Ähnlichkeitsmasse gibt Tabelle 4. Dabei werden die unterschiedlichen Werte für die verschieden ähnlichen Wortpaare aufgezeigt. In Kapitel 5.3.4 wird die Nützlichkeit der semantischen Ähnlichkeitsmasse bei der Auflösung von Bridging Coreference evaluiert.

Zur Auflösung nominaler Anaphora gehört auch die Behandlung von Markables, die aus Named Entities bestehen oder NEs enthalten. In vielen Systemen wird das Erkennen und Klassifizieren von NEs als Aufgabe des Preprocessings und als von der Koreferenz-Auflösung getrennte Aufgabe betrachtet. Um sinnvoll Ähnlichkeitsmasse zwischen NEs und nominalen Anaphern berechnen zu können, müssen die NEs auf Synsets aus WordNet abgebildet werden. Die traditionelle Verwendung von semantischen Hauptklassen wie *Person, Location, Object, Facility, Percent, Time, Number* etc. ist für die Berechnung der Ähnlichkeitsmasse zu ungenau

[29] Dafür verwendet Schütze den X^2-Assoziationstest und gewisse Wortfrequenzwerte als Grenzen.

	„car“ – „bike“	„car“–„fork“	„car“–„human“
Path	0.33	0.14	0.06
Leacock & Chodorow	2.59	1.74	0.92
Wu & Palmer	0.92	0.7	0.4
Resnik	6.81	3.45	1.37
Jiang & Conrath	0.22	0.08	0.08
Lin	0.74	0.37	0.19
Hirst & St-Onge	5	2	0
Lesk	469	59	189
Gloss Context Vectors	0.51	0.16	0.26

Tabelle 4: Beispielwerte verschiedener Ähnlichkeitsmasse in WordNet (auf zwei Kommastellen gerundet)

(s. Kapitel 1.1, S. 1). Im Hinblick auf die Abbildung von NEs auf möglichst spezifische Konzepte aus WordNet werden im Folgenden verschiedene Verfahren zur *Named Entity Recognition* (Eigennamen-Erkennung) besprochen.

2.5 Eigennamen-Erkennung und -Klassifikation

Die Erkennung und Klassifikation von Eigennamen spielt eine zentrale Rolle in Koreferenz-Auflösungssystemen. Aufgrund der ständig wachsenden Anzahl der NEs (z.B. Firmen- oder Produktenamen) können NEs nicht abschliessend in Listen festgehalten werden (was bei gewöhnlichen Nomen annäherungsweise möglich ist), die über unrestringierten Texten als Ressource benutzt werden können. Dennoch müssen für die Koreferenz-Auflösung auch für NEs Merkmalswerte wie Numerus, Genus, semantische Klasse etc. generiert werden, um sie in die Koreferenz-Auflösung miteinzubeziehen. Im Folgenden werden einige Ansätze zur Merkmalswert-Generierung für NEs diskutiert, die in Systemen zur Koreferenz-Auflösung verwendet wurden.

2.5.1 Gazetteers, Listen und WordNet als Ressourcen

Erste Ansätze zur Named Entity Recognition (NER) für Koreferenz-Auflösung benutzten vorgefertigte Ressourcen (Listen, *Gazetteers*) um Merkmalsvektoren zu generieren, wobei von restringierten Texten mit erfassten NE-Klassen ausgegangen wurde. Koreferenz-Relationen zwischen NEs wurden mit modifizierten String Match-Verfahren gelöst (Bontcheva *et al.*, 2002). Bikel *et al.* (1999) wendeten zusätzlich ein Hidden-Markov-Modell auf gewisse Indikatoren an (z.B. „Mr.“ als Indikator für die Klasse *Person*) um NEs zu klassifizieren.

Evans & Orăsan (2000) wiesen auf die Nachteile und Einschränkungen solcher handgefertiger Ressourcen und Heuristiken hin:

> „In most cases, systems have consulted domain specific gazetteers or databases in order to obtain gender information about NPs in texts. This solution suffers from the drawbacks that the construction of such resources is labour-intensive, may require expert domain knowledge, and must be maintained in order to prevent the information from becoming outdated. For example, as times change, bin men become refuse operatives and debt collectors become debt factors. In general, it is desirable to minimize or even eliminate the amount of human intervention necessary to bring a system to an effective operating capacity. [...] Recognition of animate entities must, at some point be grounded in world-knowledge.“ (S. 2,6)

Um dieses Weltwissen in ein Anaphernresolutions-System einzubauen, zogen Evans & Orăsan (2000) WordNet als Ressource hinzu. Ziel war es, die Belebtheit von NPs zu bestimmen und sie als Merkmal in ein Anaphernresolutions-System zu integrieren. WordNet enthält diverse sogenannte *Unique Beginners*, die als generellste Konzepte gelten und denen alle anderen Einträge hierarchisch untergeordnet sind. Im Falle der Nomen besteht zwischen den Unique Beginners und denen ihnen untergeordneten Konzepten eine Hyponymie-Relation. Bei den Verben führen Folgebeziehungen (*Entailment*) zu den Unique Beginners hin. Für die Belebtheit von Nomen bestimmten Evans & Orăsan (2000) *Animal, Person, Relation* als Unique Beginners und für die Verben entsprechend *Cognition, Communication, Emotion, Social*.
Named Entities sind in der Nomen-Taxonomie von WordNet nicht enthalten; anhand der Verb-Hierarchie ermitteln Evans & Orăsan (2000) aber, ob ein vorliegendes Verb zur oben genannten Menge gehört oder durch die Entailment-Hierarchie in WordNet von ihr subsumiert wird. Alle diese Verben subkategorisieren als Subjekt eine Person. Wenn eine NE Subjekt eines solchen Verbs ist, ist sie folglich belebt. Mit einem Algorithmus zur Bestimmung der Belebtheit von NPs, der die genannte WordNet-Überprüfung miteinbezieht, konnte über einem Segment des BNC-Korpus (Francis & Kucera, 1979) ein Recall von 75.88% und eine Precision von 77.25% erreicht werden.

Magnini *et al.* (2002) verwendeten ebenfalls WordNet als Ressource für ein NER-System und verzichteten vollständig auf die Verwendung von Gazetteers. Sie benutzten interne (den String betreffende) und externe (den Kontext betreffende) Hinweise für die Klassifikation von NEs. Ein interner Hinweis ist das Auffinden einer NE in Wordnet (z.B. „Galileo“ oder „New York“). Durch die Hyponymie-Relationen von WordNet wird nachgeschlagen, welcher festgelegten

semantischen Klasse die NE untergeordnet ist. Da mit einer simplen Heuristik, die Klein- und Grossschreibung untersucht, nur 3876 NEs aus WordNet extrahiert werden konnten, wurde zusätzlich der Kontext der NEs betrachtet. Für die Kontext-Betrachtung wurden Schlüsselwörter aus WordNet für jede semantische Klasse extrahiert, indem in der Hierarchie deren jeweiligen Tochterknoten bestimmt wurden. Mit einem Regelsystem wurde dann die Klassifikation gesteuert. Eine einfache Regel besagte z.B., dass eine NE, die dem Muster *NE be-Verb DET NP_Person* entspricht, eine Person ist („Hansteen was an astronomer"); ebenso wenn einer NE ein Komma und *who* folgt („Pangborn, who flew across the Pacific Ocean"). Das System wurde über dem MUC-7-Korpus evaluiert und erreichte ein F-Score von 84.86%. Damit lagen Magnini *et al.* (2002) hinter den besten Systemen (gegen 93% F-Score), die intensiv komplexe Gazetteers und Listen verwendeten. Magnini *et al.* zeigten auf, dass WordNet eine nützliche Ressource für NER ist, die das manuelle Erstellen von Listen, wenn nicht ganz ablösen, immerhin stark unterstützen kann.

Da sehr wenige NEs von WordNet erfasst werden, ist die direkte Verwendung von WordNet zur NER wenig sinnvoll. Dennoch kann WordNet im Rahmen von NER sinnvoll genutzt werden, indem, wie bei Evans & Orăsan (2000), die Entailment-Hierarchie der Verben benutzt wird, um auf belebte Subjekte zu schliessen oder indem, wie bei Magnini *et al.* (2002), aus den Tochterknoten der semantischen Klassen Schlüsselwörter extrahiert werden, die bei der Betrachtung des Kontext der NE verwendet werden.

De Meulder & Daelemans (2003) verwendeten TiMBL[30] um NEs aus deutschen und englischen Texten zu klassifizieren. Das System wurde mit einem Feature Set trainiert, das Vektoren mit 37 Merkmalen enthielt. Die Vektoren wurden aus nicht annotiertem Text generiert und kodierten Kontext-Informationen (N-Gramme) und Part Of Speech-Tags der NEs. Auch morphologische Information in Form von Prä- und Suffixen (die ersten, bzw. letzten drei Buchstaben eines Wortes) wurde festgehalten. Darüber hinaus wurde in den Vektoren kodiert, in welchen der verwendeten Gazetteers (*Organizations, Persons, Locations, Miscellaneous*) die Wörter enthalten waren. Mit diesem Ansatz konnte ein F-Score von 85.89% erreicht werden. Das System wurde von Hoste (2005) für die Koreferenz-Auflösung verwendet.

Ng (2007) trainierte einen NE-Classifier über dem BBN Entity Type-Korpus (Weischedel & Brunstein, 2005) mit den entsprechenden sechs semantischen Klassen (*Person, Organization,*

[30] s. Kapitel 2.2

Geographical-political Region (GPE), Facility, Location, Others) aus ACE (Mitchell *et al.*, 2003). Dabei wurden die Resultate von vier Verfahren zur NER jeweils als Merkmal in den Trainingsvektoren abgebildet:

1. Der *BBN IdentiFinder* (Bikel *et al.*, 1999) wurde benutzt, der NEs semantische Klassen gemäss den Richtlinien von MUC zuweist. Die MUC-Klassen wurden dabei auf ACE-Klassen abgebildet.

2. Für jede ACE-Klasse wurde eine Synonym-Liste mit Wörtern aus WordNet generiert und überprüft, ob eine NP Hyponym eines solchen Worts ist.

3. Durch Korpus-Recherche wurden Appositionen zwischen NE-Klassen und Nomen ermittelt, die über einem festgelegten Schwellenwert für die Häufigkeit lagen. Es wurde überprüft, ob die Klasse der NE, mit der das Nomen auftritt, der häufigsten Kollokation des Nomen entspricht.

4. Für jede NP wurden nach Lin (1998) die zehn ähnlichsten NPs im Korpus ermittelt.

Für die Merkmalsvektoren wurden zwei Typen von Merkmalen generiert. Der eine kodierte die gefundenen semantischen Klassen der oben genannten Verfahren und der andere, ob eine semantische Klasse (ausser *Others*) für die NP gefunden wurde und sie somit als ACE-Mention gilt. Das Verfahren wurde dann in ein ML-System zur Koreferenz-Auflösung integriert. Als die Mention-Erkennung als harter Filter für die Markables benutzt wurde, konnte eine Baseline nach Soon *et al.* (2001) um 5-7% in F-Score verbessert werden. Das binäre Merkmal *Semantic Class Agreement (SCA)* wurde in verschiedenen Testläufen entweder als Merkmal oder als Filter benutzt. Als Merkmal konnte es die Baseline um 5-8% in Recall verbessern.

Neben WordNet und diversen Textkorpora wurde in den letzten Jahren auch *Wikipedia*[31] als Ressource für die Named Entity Recognition-Forschung interessant. Einige NER-Verfahren, die Wikipedia verwenden, werden im nächsten Kapitel betrachtet.

2.5.2 Wikipedia als Ressource

Wikipedia ist ein mehrsprachiges Online-Lexikon, das von einer offenen Benutzer-Gemeinde fortlaufend ergänzt und editiert wird. Die Anzahl der Artikel und Themengebiete steigt ständig (Stand 5.8.2009: über 2979000 Artikel allein in englischer Sprache). Das Potential

[31] http://www.wikipedia.org (Stand 30.9.2009)

von Wikipedia als Ressource für NLP-Anwendungen wurde von diversen Forschungsarbeiten erkannt (s. unten). Gerade für die Klassifikation von Named Entities ist Wikipedia eine interessante Ressource, da viele öffentliche Personen, Orte, Organisationen und Firmen erfasst sind. Im Folgenden wird die Benutzung von Wikipedia in diversen NLP-Anwendungen im Hinblick auf die Koreferenz-Auflösung betrachtet.

Bunescu & Pasca (2006) waren eine der ersten Forschergruppen, die Wikipedia als Ressource für eine NLP-Anwendung verwendeten. Sie benutzten die *Redirect-* und *Disambiguation-*Seiten und die *Categories* für die Eigennamen-Erkennung und -Disambiguierung. Strube & Ponzetto (2006) benutzten Wikipedia um die semantische Verwandtschaft zwischen Wörtern zu messen. Sie sahen als Vorteile von Wikipedia die grosse Abdeckung von verschiedenen NEs und Konzepten, die beispielsweise von WordNet nicht erfasst werden. Wikipedia bildet durch das Kategorien-System eine Art Taxonomie. Diese kann aber nicht als strukturierte Ontologie in Form einer Baumstruktur verstanden werden, da Artikel in verschiedenen Kategorien erfasst sein können und teilweise verschiedenen Überkategorien untergeordnet sind. Das Kategorien-System formt also eher einen gerichteten Graphen.

Wikipedia listet für Wörter mit ambigen Bedeutungen Disambiguierungs-Seiten. Um den entsprechenden Artikel zu einem Wort i in Bezug zu einem Wort j zu finden, überprüfen Strube & Ponzetto (2006) das Vorkommen des Worts j in den Kurzbeschreibungen der Disambiguierung von i. Wenn j selbst ambig ist, werden diejenigen Artikel von i und j genommen, die in den Kurzbeschreibungen ein gemeinsames Wort k enthalten. Für $i =$ „King“ und j= „Rook“ beispielsweise ist $k =$ „Chess“. In diesem Fall werden die Artikel von i und j extrahiert, in denen „Chess“ vorkommt. Wenn kein k gefunden wird, wird der erste Artikel der Disambiguierungs-Seite genommen.

Um semantische Ähnlichkeit zu messen, werden nun die Kategorien der gefundenen Artikel von i und j extrahiert. Zwischen den einzelnen Kategorien von i und j werden Pfade im gerichteten Graphen des Kategorien-Systems mit einer maximalen Länge von vier Kanten gesucht. Die Masse für die Distanz wurden von Pedersen *et al.* (2004)[32] übernommen. Für die pfadbasierten Ähnlichkeitsmasse wird der kürzeste Pfad ermittelt, für die Information Content-basierenden Masse derjenige Pfad, der den Information Content[33] maximiert.

[32] Vgl. Kapitel 2.4.4
[33] Vgl. Kapitel 2.4.4

Dieses Vorgehen wurde in einem System zur Koreferenz-Auflösung (Strube *et al.*, 2002) verwendet um semantische Ähnlichkeit zwischen den Markables zu messen und anhand des ACE Trainingskorpus 2003 gegenüber einem Baseline-System nach Soon *et al.* (2001) evaluiert. Zusätzlich zu den WordNet- und Wikipedia-Merkmalen wurden für die Markables ihre semantischen Rollen (SRL) anhand des ASSERT-Parsers (Pradhan *et al.*, 2004) ermittelt und in den Trainingsvektoren kodiert. Das Baseline-System konnte mit dem erweiterten Feature Set um 3.4% in MUC F-Score verbessert werden. Die Verbesserung beruhte auf dem erhöhten Recall, während die Precision tendenziell leicht abnahm.

Kazama & Torisawa (2007) benutzten Wikipedia ebenfalls als Ressource für ein System zur Eigennamen-Erkennung. Anders als Bunescu & Pasca (2006) verwendeten sie nicht die Redirect- und Disambiguierungs-Seiten oder das Kategorien-System, sondern betrachteten den jeweils ersten Satz der Artikeltexte. Als Vorteil von Wikipedia als Ressource für NER gegenüber freiem Text hoben Kazama & Torisawa (2007) heraus, dass Wikipedia eine gewisse strukturelle Konsistenz aufweist. So kann der erste Satz eines Artikels jeweils als Definition des Gegenstands verstanden werden, der im Artikeln beschrieben wird:

> „Jimi Hendrix (November 27, 1942, Seattle,Washington – September 18, 1970, London, England) was an American **guitarist**, **singer** and **songwriter**."

Die Artikel selbst wurden durch die Konkatenation von den Wörtern in der NE aufgefunden (z.B. *Jimi Hendrix* → *Jimi_Hendrix*). Nachdem das Markup im Artikeltext entfernt wurde und ein PoS-Tagger die Wortarten bestimmt hatte, extrahierten Kazama & Torisawa die erste NP nach *is/was/are/were* als semantische Klasse der NE. Wenn mehrere NPs gelistet waren, wurde die letzte genommen. Diese Heuristik wurde angewandt, wenn der Kopf einer solchen NP nicht *one/type/kind/sort/name* gefolgt von *of* war. In solchen Fällen wurde die zweite NP extrahiert:

> „Jazz is [a kind]NP [of]PP [**music**]NP characterized by swung and blue notes."

Mit diesem Verfahren konnten Kazama & Torisawa (2007) den F-Score einer Baseline, die nur mit Gazetteers aus freien Texten über dem CoNLL 2003-Korpus (Tjong Kim Sang & De Meulder, 2003) operierte, um maximal 3.03% auf 88.08% erhöhen.

Nachdem in Kapitel 2.4 relevante Merkmale und Ressourcen für die Auflösung von Koreferenz und in diesem Kapitel Verfahren für die Abbildung von Named Entities auf Nomen präsentiert wurden, wird im Folgenden die in Kapitel 2.3 erwähnte Balancierung und Filterung

des Trainingskorpus im Hinblick auf die Implementation eines ML-Systems zur Koreferenz-Auflösung in Kapitel 4 aufgegriffen.

2.6 Balancierung der Trainingsdaten

Ausgehend vom Versuch, False Positives und False Negatives zu reduzieren (z.B. im Rahmen von MUC-5) und ein balanciertes Trainingskorpus[34] anzunähern, ist festzuhalten, dass die Bestimmung von Anaphorizität für die Koreferenz-Auflösung von zentraler Bedeutung ist. Regelbasierte Systeme hielten Konstellationen fest, in denen ein Markable unmöglich Antezedens einer Anapher sein konnte, z.B. aufgrund von Genus-Inkongruenz (Lappin & Leass, 1994; Mitkov, 1998). Ng & Cardie (2002b) versuchten mit mässigem Erfolg diese Regeln und weitere Merkmale in einem ML-Verfahren zu integrieren. Ng (2004) benutzte deshalb die Merkmale aus Ng & Cardie (2002b) als harte Filter. Die Prämisse war, dass ein zu konservatives System, das hart filtert – also Paare, die den Filter aktivieren, löscht – zu viele True Mentions (Markables, die in einer Koreferenzmenge sind) eliminiert. Ein zu liberales System hingegen filtert zu wenige nicht anaphorische Markables und balanciert das Trainingskorpus nicht aus. Deshalb definierte Ng (2004) den *Conservativeness Parameter*. Die *Conservativeness* wird dabei anhand eines *Cost Ratio*-Parameters definiert.

$$Cost\ Ratio := \frac{Kosten\ für\ Fehlklassifikation\ einer\ positiven\ Instanz}{Kosten\ für\ Fehlklassifikation\ einer\ negativen\ Instanz}$$

Über diesen Parameter lässt sich nun die *Conservativeness* eines Classifiers bestimmen: Je höher die Kosten für eine Fehlklassifikation gesetzt wird, desto „konservativer“ wird der Classifier. Wie konservativ ein Classifier ist, lässt sich ausserdem anhand der Anzahl klassifizierter NPs im Verhältnis zu allen vorhandenen NPs bestimmen. Je mehr NPs ein Classifier als nicht anaphorisch bestimmt, desto konservativer ist er. Trainiert man ein Wahrscheinlichkeits-Modell für die Bestimmung von Anaphorizität, so kann man bei der Klassifikation von neuen Instanzen ein Schwellenwert (*Threshold*) für die Wahrscheinlichkeit definieren, der mindestens erreicht werden muss, um eine NP als nicht anaphorisch zu klassifizieren.

Mit dieser Parametrisierung des Classifiers von Ng & Cardie (2002b) konnte Ng (2004) eine Verbesserung des F-Scores um 3-4% erreichen. Wie oft bei Systemen, die mit Filtern arbeiten, wurde dabei tendenziell die Precision auf Kosten des Recalls erhöht. Dies liegt am Umstand, dass Filter nie mit hundertprozentiger Zuverlässigkeit arbeiten und True Mentions

[34] Vgl. Kapitel 2.3

eliminieren. Dies führt zu einer Reduktion des Recalls. Aufgrund des ausgeglicheneren Trainingskorpus erhöht sich dafür die Precision. Die Parametrisierung des Classifiers ist folglich ein Verfahren, das benutzt werden kann, um das Verhältnis von Precision und Recall festzulegen. Anzumerken bleibt, dass die Bestimmung von sinnvollen Werten für die Parameter empirisch in mehreren Trainings-Durchgängen ermittelt werden muss.

2.6.1 *discourse-new* vs. *discourse-old*

Eine Heuristik zur Bestimmung der Anaphorizität von nominalen Markables ist, wie aufgezeigt, die Untersuchung der Definitheit. Weitgehend ist die Annahme verbreitet, dass indefinite NPs nicht anaphorisch sind. Dass bedeutet umgekehrt nicht, dass definite NPs prinzipiell anaphorisch sind. Vieira & Poesio (2000) stellten fest, dass nur ca. 50% der definiten NPs in ihrem Korpus anaphorisch waren. In der differenzierten Unterscheidung der Anaphorizität von definiten NPs liegt also ein grosses Potenzial um Markables zu filtern, die nicht in einer Koreferenzmenge sind.

Um die Anaphorizität von definiten NPs zu bestimmen, untersuchten Poesio *et al.* (2004) in Anlehnung an Prince (1992), ob sie bei ihrem Auftreten in einen Diskurs neu eingeführt werden (*discourse-new*), oder ob sie schon einmal aufgetreten sind (*discourse-old*). Eine diskursneue NP ist praktisch nie anaphorisch, sie kann nur als Antezedens fungieren. Zudem wird überprüft, ob eine NP *unique* ist, das heisst entweder einer sogenannten *Large Situation* (z.B. *„the pope“, „the sun“, „the world“*), also einer allgemein etablierten Entität, entspricht oder eine Named Entity (z.B. *„the Mount Everest“, „the Grand Canyon“*) ist. Als weiteres Merkmal wurden für die NPs deren Definitheits-Wahrscheinlichkeit nach Uryupina (2003) bestimmt. Dabei gilt: X ist eine NP mit einem Artikel, Y ist die NP ohne Artikel und H ist der Kopf der NP. Es werden vier Quotienten anhand der Anzahl der von der Suchmaschine *AltaVista*[35] gefundenen Websites berechnet:

$$\frac{\#the\,Y}{\#Y}, \frac{\#the\,H}{\#H}, \frac{\#the\,Y}{\#a\,Y}, \frac{\#the\,H}{\#H}$$

Die Quotienten sind ein korpusbasiertes Mass für die definite und indefinite Verwendung einer NP. Diese und die *definiteness*- und *uniqueness*-Merkmale wurden in ein Feature Set für das GUITAR-System zur Koreferenz-Auflösung (Vieira & Poesio, 2000) integriert. Dazu kamen Merkmale, die beschrieben, ob eine NP per String Matching auf eine andere NP im Text abgebildet werden kann, ob die NP in einer Apposition oder Kopula vorkommt, ob die NP durch eine Relativ- oder Präpositionalphrase postmodifiziert wird, und ob die NP im Titel, im

[35] http://www.altavista.com (Stand 30.9.2009)

ersten Satz oder im ersten Paragraphen vorkommt. GUITAR wurde mit und ohne Bestimmung der Anaphorizität der definiten NPs über dem GNOME-Korpus (Poesio, 2004) evaluiert. Das erweiterte System übertraf das Baseline-System dabei um knapp 3% in F-Score.

2.6.2 Filtern des Trainingskorpus

Die Bestimmung der Anaphorizität ist ein Verfahren, um Markables von der Generierung von möglichen Koreferenz-Paaren auszuschliessen. Die im Folgenden betrachteten Filterungen hingegen werden vorgenommen, während die Paare generiert werden, oder nach der Generierung der Trainingsvektoren.

Hendrickx *et al.* (2007) evaluierten detailliert, inwiefern die linguistisch motivierten Filterungen des Trainingskorpus die Performanz eines Classifiers beeinflussen. Es wurden fünf Filter implementiert:

1. $fdef$: Indefinite NPs können nie Anapher sein.
2. $fhead$: Antezedenzien und Anaphern können nicht mehr als drei Sätze voneinander entfernt sein, ausser ihr lexikalischer Kopf ist identisch.
3. $fagree$: Numerus- und Genus-Kongruenz von Antezedens-Kandidaten und Pronomen muss erfüllt sein.
4. $fmatch$: String Match: Wenn die Strings zweier Instanzen identisch sind (ohne Artikel), werden sie als koreferent erachtet und nicht mehr durch den Classifier klassifiziert. Dies ist der einzige „positive“ Filter.
5. $f3s$: Die Distanz zwischen einem möglichen Antezedens und einem Pronomen darf drei Sätze nicht überschreiten.

Aus der Kombination dieser Filter ergab sich eine Reduktion des verwendeten Trainingskorpus KNACK-2002 (Hoste, 2005) um bis zu 91.7%. Dabei wurde der Anteil von positiven Instanzen von 8.5% auf maximal 17.7% erhöht. Die Performanz des Classifiers konnte in Bezug auf die Bestimmung einzelner Koreferenz-Relationen von 46.7% auf maximal 70.8% in F-Score erhöht werden. Die MUC-Evaluation, die nicht einzelne Klassifikationen, sondern Koreferenzmengen evaluiert[36], ergab eine Verbesserung des F-Score um 1.4%, wobei das Ungleichgewicht von Recall und Precision augenfällig ist. Im Anwenden der Filter wurde sukzessiv Recall zugunsten der Precision verringert (Tabelle 5).

[36] Vgl. Kapitel 3

	Recall	Precision	F-Score
keine Filter	**60**	**35.2**	**44.4**
fdef	49.2	46.7	47.9
f3s	58	36.8	45.1
fagree	50.2	40.4	44.7
fhead	39.8	60.3	47.9
fmatch	46.7	48.4	47.5
combi1	40.7	46.1	43.2
combi2	**36.7**	**61**	**45.8**

Tabelle 5: MUC-Scores für das gefilterte Trainingskorpus bei Hendrickx *et al.* (2007)

Die Verwendung von Filtern auf der Ebene der Markables und bei der Paargenerierung führen zu einer Reduktion der zu lernenden und zu klassifizierenden Vektoren im Korpus. Wenn weniger negative Instanzen erstellt werden, klassifiziert ein System weniger negative Instanzen als positiv, das heisst die Precision steigt tendenziell. Das Verhalten des Recalls ist weniger prognostizierbar. Durch Filter werden meistens positive Instanzen entfernt, was eigentlich zu einer Reduktion des Recalls führt. Dennoch kann eine Filterung den Recall über die Precision erhöhen: Wenn der Classifier durch eine verbesserte Precision mehr positive Instanzen richtig klassifiziert, steigt der Recall, auch wenn durch die Filterung positive Instanzen gelöscht wurden (d.h. die Anzahl der positiven Instanzen reduziert wird)[37].

Nachdem Verfahren und Techniken diskutiert wurden, die vor der eigentlichen Klassifikation von möglichen Koreferenz-Paaren eingesetzt werden, wird im folgenden Kapitel ein jüngeres Forschungsgebiet betrachtet, das über der Ausgabe von Classifiern operiert. Dabei werden ebenfalls linguistisch motivierte Restriktionen in Bezug auf die Eigenschaften von Koreferenz-Relationen ermittelt und auf die Ausgabe von Classifiern angewendet. Die Leistungen der Classifier selbst werden dadurch nicht verbessert. Die Verbesserung der Resultate wird durch Korrekturen der Ausgaben erreicht.

2.6.3 Konsistenz von Äquivalenzklassen

Ein aktuelles Forschungsgebiet in der Koreferenz-Auflösung ist die Sicherstellung der Konsistenz von Äquivalenzklassen anhand linguistischer Restriktionen (Klenner, 2007; Finkel & Manning, 2008; Ailloud & Klenner, 2009; Denis & Baldridge, 2009). Traditionelle ML-Verfahren betrachten Koreferenz-Auflösung als Klassifikationsproblem zwischen zwei

[37] Dies bestätigt die Evaluation in Kapitel 5.3

Markables (ob diese koreferent sind oder nicht). Äquivalenzklassen werden anschliessend über der Ausgabe der Classifier generiert. Die einfachste Methode dazu besteht darin, die Transitivitäts-Eigenschaft von Koreferenz-Relationen zu benutzen. Wenn eine Instanz A mit einer Instanz B koreferent ist und B mit einer Instanz C koreferiert, so sind auch A und C koreferent. Klenner (2007, S. 1) gibt folgendes Beispiel, das die Problematik solcher transitiven Verkettungen aufzeigt:

„A [man] stole/sold [him] [his] car. [Peter] was angry/happy."

Die Binding-Theorie gibt vor, dass „man" und „him" exklusiv sind, da „him" das indirekte Objekt des Verbs ist, von dem „man" Subjekt ist. „his" hingegen kann sich sowohl auf „man" als auch auf „Peter" beziehen. Die Koreferenz hängt vom Verb, bzw. dessen Bedeutung ab.

Wenn nun ein harter Filter implementiert wird, der die Koreferenz zwischen „man" und „him" eliminiert, ist die Exklusivität von weiteren Koreferenzen zwischen Instanzen, die mit „man" oder „him" koreferent sind, nicht gewährleistet (z.B. dass „man" und „Peter" nicht koreferent sein können). Der transitive Charakter dieser Exklusivität wird in vielen ML-Verfahren zur Koreferenz-Auflösung nicht berücksichtigt. Auch wird Transitivität kaum benutzt um False Negatives zu verhindern. Es ist denkbar, dass ein Classifier A und B, bzw. B und C richtig als koreferent, und A und C fälschlicherweise als nicht koreferent klassifiziert.

$Peter_A$ said: „I_B saw $myself_C$ in the mirror."

Ein Classifier könnte beispielsweise $Peter_A$ und $myself_C$ als nicht koreferent klassifizieren. Die Fehlklassifikation kann durch Transitivität entdeckt und behoben werden, da durch sie vorgegeben ist, dass $Peter_A$ und $myself_C$ koreferent sein müssen, wenn es $Peter_A$ und I_B, bzw. I_B und $myself_C$ sind.

Um solche Inkonsistenzen in Äquivalenzklassen zu beheben, wird die Ausgabe eines binären Classifiers mit *Integer Linear Programming (ILP)* untersucht. ILP kann eine numerische Lösung eines Problems in numerischer Repräsentation unter Berücksichtigung gewisser Restriktionen optimieren[38]. Als Restriktionen definiert Klenner (2007) nun die Restriktionen aus der Binding-Theorie und deren Transitivität auf die jeweilige Äquivalenzklasse.
Mit dieser Optimierung erhöht Klenner (2007) die Ausgabe eines Classifiers über dem ACE-NWIRE-Korpus von 47% auf 53.3% in CEAF-F-Score[39], wobei sowohl Precision (um 7.8%) als auch Recall (um 3.9%) erhöht werden. Denis & Baldridge (2009) berichten mit

[38] Vgl. Roth & Yih (2004) zur Anwendung von ILP in NLP
[39] Vgl. Kapitel 3.3

einem ähnlichen ILP-Verfahren (ohne die Binding-Restriktionen und deren Transitivität zu berücksichtigen) eine maximale Verbesserung von 7.5% in CEAF-F-Score bei der Evaluation über dem gesamten ACE-Korpus, wobei die Verbesserung durch ein Rückgang des Recalls und eine Erhöhung der Precision erreicht wird.
Die Überprüfung der Konsistenz von Äquivalenzklassen ist folglich ein wichtiges Verfahren, das grundlegende Eigenschaften von Koreferenz-Relationen (Transitivität und Exklusivität) auf die Ausgabe von Classifiern projiziert. Dadurch können augenfällige Fehler in der Ausgabe behoben werden.

Nachdem nun die Voraussetzungen für die Implementation eines ML-Systems zur Koreferenz-Auflösung gegeben sind, werden im nächsten Kapitel wichtige Evaluationsmasse, die für die Evaluation des zu implementierenden Systems verwendet werden, und deren Vor-und Nachteile betrachtet.

3 Evaluationsmasse für Koreferenz-Auflösung

Um die Performanz von Systemen zu messen, die Anaphern- oder Koreferenz-Resolution betreiben, wurden diverse Masse entwickelt und diskutiert. Hobbs (1976) gab die Leistung seines „naiven“ Algorithmus in der Anzahl richtig aufgelöster Pronomen im Vergleich zu allen vorhandenen, anaphorischen Pronomen (respektive deren Prozentsatz) an. Auch Lappin & Leass (1994) und Mitkov (1998) verwendeten die sogenannte *Success Rate*, wobei Mitkov (1998, S. 872) schon auf die Masse Precision und Recall verwies und sie als äquivalent betrachtete, da robuste Systeme immer für alle Anaphern Antezedenzien bestimmen. Die Success Rate ist äquivalent mit Recall und ergibt sich, wie erwähnt, aus dem Quotient von den von einem System richtig aufgelösten Anaphern im Vergleich zu allen im Testkorpus vorhandenen Anaphern:

$$Recall = \frac{Anzahl\, richtig\, aufgeloester\, Anaphern}{Anzahl\, im\, Testkorpus\, vorhandene\, Anaphern}$$

Um die Performanz eines Classifiers zu messen wird darüber hinaus die Precision verwendet. Precision gibt die Anzahl richtig aufgelöster Anaphern im Verhältnis zur Anzahl aufgelöster Anaphern wieder:

$$Precision = \frac{Anzahl\, richtig\, aufgeloester\, Anaphern}{Anzahl\, aufgeloester\, Anaphern}$$

Recall kann also als Mass für die Vollständigkeit einer Systemausgabe und die Precision als Mass für die Genauigkeit der Ausgabe betrachtet werden. Um diese beiden Masse miteinander zu verbinden, wird F-Score (auch *F1-/F-measure*) berechnet. F-Score ergibt sich aus:

$$F - Score = \frac{2 * (Recall * Precision)}{Recal + Precision}$$

Mitkov (2000) wies auf folgende Mängel dieser Masse hin und propagierte neue Evaluationsmethoden für die Anaphernresolution.

- **Systeme und Algorithmen** werden nicht separat getestet. Einige AR-Systeme benutzen manuell aufbereite und daher (fast) fehlerfreie Eingaben, während andere voll automatisiert sind. Mitkov *et al.* (2002) zeigten die Abhängigkeit von AR-Systemen vom Preprocessing auf. Durch Fehler in automatisch aufbereiteten Eingaben entsteht eine obere Grenze für die mögliche Höchstleistung eines Systems (*Ceiling*). Auch verwenden teilweise ähnliche Systeme unterschiedliche Preprocessing-Tools. Solche Unterschiede werden in Evaluationswerten nicht wiedergegeben.

- **Unterschiedliche Definitionen von *Recall***: Aone & Bennett (1996) definierten Recall als $\frac{Anzahl\,richtig\,aufgelöster\,Anaphern}{Anzahl\,vom\,System\,erkannten\,Anaphern}$, während Baldwin (1997) als Divisor die Anzahl aller Anaphern im Text[40] nimmt. Aone & Bennett's Wert gibt keinen Aufschluss darüber, wie gut ein System z.B. pleonastische Pronomen behandelt. Ein System könnte auch nur einfach aufzulösende Anaphern identifizieren und so hohe Werte erzielen.

- **Kein Schwierigkeitsmass für unterschiedliche Arten von Anaphern**: Fälle, in denen Merkmale wie z.B. Numerus und Genus als Indikatoren für die Bestimmung eines einzelnen Antezedens nicht ausreichen, werden gleich bewertet wie Anaphern, die nach der Numerus-Genus-Filterung nur noch ein mögliches Antezedens haben. Mitkov's Idee war richtig getroffene, schwierige Entscheidungen stärker zu gewichten als einfachere.

Mitkov strebte eine Verfeinerung der Evaluationsmethoden an, in denen die Schwierigkeit der Aufgabe skaliert und das Test-Setting berücksichtigt werden. Er schlug erweiterte Success Rates vor (*Non-trivial Success Rate, Critical Success Rate*) und legte die Evaluation eines Systems in Bezug auf drei Kategorien nahe:

1. Evaluation gegenüber einem Baseline-Modell
2. Evaluation gegenüber ähnlichen Verfahren
3. Evaluation gegenüber „klassischen" Ansätzen

Ausserdem empfahl Mitkov die Evaluation des Einflusses einzelner Merkmale auf die Leistung eines Systems. Dabei sollte der jeweilige Beitrag eines Merkmals an die richtigen Entscheidungen (*Decision Power*) ermittelt und die Leistung des Systems ohne dieses Merkmal (*Relative Importance*) gemessen werden. Die drei Evaluations-Kategorien wurden in der Forschung weitgehend standardisiert. In ML-Verfahren werden die Beiträge der einzelnen Merkmale zur Leistung eines Systems in Anlehnung an *Decision Power* und *Relative Importance* beispielsweise in Information Gain und Gain Ratio gemessen und deklariert. Bei der Verwendung von C4.5-Entscheidungsbäumen wird für die Knoten die Anzahl (oder der Prozentsatz) der Fälle angegeben, die vom jeweiligen Knoten (mit-) entschieden wurden.

Die oben genannten Masse evaluieren die Leistung von Systemen bei der Anaphernauflösung. Ziel der Koreferenz-Auflösung ist es aber, nicht nur den Anaphern Antezedenzien zuzuweisen, sondern alle koreferenten Entitäten in einem Text in Äquivalenzklassen zu fassen. Für die Koreferenz-Auflösung werden aber Masse benötigt, die nicht die Klassifikationen zwischen

[40] D.h. alle von Menschen annotierten Anaphern in einem Text

jeweils zwei Instanzen bewerten, sondern welche die Qualität der von einem System generierten Äquivalenzklassen in Bezug auf diejenigen aus einem Goldstandard[41] misst; sprich evaluiert, wie ähnlich die Ausgabe eines Systems der Vorgabe des Goldstandards ist.
Die Masse müssen skaliert sein, da beim Vergleich von Äquivalenzklassen keine binären Entscheidungen gefällt werden können. Das heisst, wenn ein System im Vergleich zum Goldstandard ein Element mehr oder weniger in einer Äquivalenzklasse platziert, ist das intuitiv weniger gravierend, als wenn nur ein Element der Klassen übereinstimmt etc. Für MUC-6 formulierten Vilain *et al.* (1995) ein Modell-theoretisches Mass, das die Anzahl benötigter Verknüpfungen (*Links*) berechnet, um die Äquivalenzklassen des Goldstandards (der *Key*) in jene der Ausgabe eines Systems (der *Response*) und umgekehrt zu transformieren.

3.1 MUC-6-Score nach Vilain *et al.*

Das Mass von Vilain *et al.* (1995) folgt dem Grundsatz, dass die Ähnlichkeit der Äquivalenzklassen des *Keys* mit denjenigen der *Response* anhand der minimalen Anzahl Links skaliert werden kann, die nötig sind, um beide Klassen gleichzusetzen.
Um Recall zu messen wird ermittelt, in wie viele Partitionen $p(S)$ die Response den Key aufteilt. Dabei teilt eine Response-Klasse {A C} eine Key-Klasse {A B C} in die Partitionen {A C} und {B} auf. Dann wird gezählt, wie viele Links benötigt werden, um die durch die Response partitionierte Äquivalenzklasse S des Keys wiederherzustellen. Folgende Funktionen spielen dabei eine Rolle:

- $c(S)$: Die Anzahl minimal benötigter Links um die Äquivalenzklasse S herzustellen. $c(S)$ entspricht folglich der Kardinalität von S-1, also $c(S) = (|S| - 1)$
- $m(S)$: Die Anzahl fehlender Links der Response im Vergleich mit der Klassse S des Keys. Die Anzahl Links in der Response entspricht der Anzahl Links zwischen den Komponenten in den Partitionen $p(S)$ der Response und ergibt sich folglich aus der Anzahl Komponenten der Partitionen -1, also $m(S) = (|p(S)| - 1)$.

Der Recall-Fehler kann nun einfach bestimmt werden durch $\frac{m(S)}{c(S)}$ und Recall selbst ist folglich $\frac{c(S)-m(S)}{c(S)}$. Durch Einsetzten ergibt sich $\frac{(|S|-1)-(|p(S)|-1)}{|S|-1}$, respektive $\frac{|S|-|p(S)|}{|S|-1}$. Vilain *et al.* (1995, S. 47) veranschaulichen das Vorgehen anhand des folgenden Beispiels:

[41] Der Goldstandard entspricht dem (manuell) korrekt annotierten Text. Automatische Systeme versuchen diese Annotation möglichst genau anzunähern. Die Diskrepanz zwischen Goldstandard und der Annäherung dient dann als Grundlage für die Beurteilung des jeweiligen Systems. Sinnvolle Beurteilungen für diese Diskrepanz zu definieren ist Aufgabe der Evaluationsmasse.

- Key: <A-B B-C>
- Response: <A-C>

Die Äquivalenzklasse S des Keys ist folglich {A B C}. Die minimale Anzahl Links $c(S)$ wird anhand der Kardinalität von $|S| = 3$ bestimmt: $c(S) = (|S| - 1) = 2$. Die Response partitioniert S nun in zwei Teile, nämlich {A C} und {B}. Die Recall-Formel ergibt $\frac{3-2}{3-1} = 1/2$. Das Resultat entspricht der Intuition, dass eine von zwei möglichen richtigen Partitionen gefunden wurde.
Um Precision zu berechnen, wird nun die Richtung umgekehrt, in der partitioniert wird. Es wird betrachtet, in welche Partitionen der Key die Response aufteilt. Dann wird gezählt wie viele Links benötigt werden, um die partitionierte Response wiederherzustellen. Die Formeln für den Recall können also übernommen werden, wobei S, die Klasse des Keys, durch S', diejenige der Response, ersetzt wird. Das ergibt gemäss Vilain *et al.* (1995, S. 48) für das Beispiel

- Key S: {B C D}
- Response S': {A B C}

also $Precision = \frac{|S'|-|p(S')|}{|S'|-1}$ folglich $\frac{3-2}{3-1} = 1/2$. Das ist intuitiv richtig, da von den zwei minimalen Links, die für das Generieren der Response-Klasse S' nötig sind, einer gefunden wurde.

3.2 B-CUBED-Evaluation nach Bagga & Baldwin

Bagga & Baldwin (1998) bemängelten am MUC-Score von Vilain *et al.*, dass bei der Beurteilung der Precision alle Klassifikationsfehler gleich stark bewertet werden. Wenn ein Key drei Äquivalenzklassen vorgibt und eine erste Response zwei der Klassen verknüpft und eine zweite Response zwei andere Klassen miteinander verbindet, werden beide Fehler gleich stark gewichtet. Dabei werden z.B. bei der Klassifikation durch die erste Response weniger nicht koreferente Entitäten als koreferent klassifiziert als bei der zweiten (Vgl. Tabelle 6). Beide erhalten dennoch denselben Wert für die Precision, da gleich viele Partitionen generiert werden. Anstatt also die Links als Grundlage für ein Evaluationsmass zu nehmen, betrachteten Bagga & Baldwin das Vorhandensein oder Fehlen jeder einzelnen Instanz relativ zu den anderen Elementen einer Äquivalenzklasse. Folglich werden für jede Instanz i Recall und Precision berechnet.

Key:	{1,2,3,4,5}	{6,7}	{8,9,A,B,C}
Response 1:	{1,2,3,4,5,6,7}		{8,9,A,B,C}
Response 2:	{1,2,3,4,5,8,9,A,B,C}	{6,7}	

Tabelle 6: Unterschiedliche Klassifikationsfehler nach Bagga & Baldwin (1998, S. 2-3)

$$Precision_i = \frac{Anzahl\, korrekter\, Elemente\, der\, Response - Klasse\, von\, Instanz_i}{Anzahl\, der\, Elemente\, der\, Response - Klasse\, von\, Instanz_i}$$

$$Recall_i = \frac{Anzahl\, korrekter\, Elemente\, der\, Response - Klasse\, von\, Instanz_i}{Anzahl\, der\, Elemente\, der\, Key - Klasse\, von\, Instanz_i}$$

Die finalen Recall- und Precision-Werte für ein System ergeben sich aus der Summe der Anzahl aller Entitäten N im entsprechenden Text multipliziert mit den Precision- und Recall-Werten von $i_1...i_n$ und einem Gewicht w_i[42].

$$Final\, Precision = \sum_{i=1}^{N} w_i * Precision_i$$

$$Final\, Recall = \sum_{i=1}^{N} w_i * Recall_i$$

Anknüpfend an die Modell-theoretische Einbettung von Vilain *et al.* (1995) definieren Bagga & Baldwin (1998) für eine Äquivalenzklasse S eines Keys und den Äquivalenzklassen $R_1...R_m$ der Response folgende Funktionen.

- $p(S)$ ist eine Partitionierung von S entsprechend der Überschneidungen der Klassen R_i der Response mit S. $p(S)$ enthält folglich alle Mengen $P_1...P_m$, die sich aus der Überschneidung der Response-Klassen R_i mit S ergeben, $p(S) = \{P_1, P_2, ..., P_m\}$. Nimmt man eine Klasse R_1 mit {A B C } und S mit {A B D E F}, dann entspricht P_1 ={A B} usw.

- $m_j(S)$ ist die Anzahl fehlender Elemente jedes Sets P_j relativ zu den Mengen aus S, $m_j(S) = (|S| - |P_j|)$.

Für alle Elemente e einer Menge P_j wird nun die Anzahl fehlender Elemente dieser Menge genommen und addiert. Daraus ergibt sich für die Anzahl aller fehlender Elemente in S:

$$\sum_{j=1}^{m} \sum_{for\, each\, e \in P_j} m_j(S)$$

[42]Bagga & Baldwin (1998) definierten verschiedene Werte für w_i für die Evaluation verschiedener NLP-Aufgaben wie Information Retrieval und Information Extraction.

Der Recall-Fehler wird analog zu Vilain *et al.* (1995) als Bruch zwischen der Anzahl fehlender Elemente und der Anzahl Elemente formuliert: $\frac{m_j(S)}{|S|}$. Da alle Instanzen und deren Mengen einzeln betrachtet werden ($|P_j| * \frac{m_j(S)}{|S|}$), ergibt sich der gesamte Recall-Fehler schliesslich aus der Summe der einzelnen Fehler:

$$Recall - Fehler = \frac{1}{|S|} \sum_{j=1}^{m} \sum_{for\ each\ e \in P_j} m_j(S)$$

$$Recall = 1 - \frac{1}{|S|} \sum_{j=1}^{m} \sum_{for\ each\ e \in P_j} m_j(S)$$

Der finale Recall ergibt sich schliesslich aus dem Durchschnitt der Recall-Werte aller Äquivalenzklassen. Für Precision werden, wie beim MUC-6-Score, die Rollen von Key S und Response S' in den Funktionen invertiert.

Nach Bagga & Baldwin (1998, S. 5) ergeben sich für die Beispiele aus der Tabelle 6 im Vergleich zum MUC-6-Score nun folgende Werte für die Precision (Tabelle 7).

	MUC-6-Score	*B-CUBED*
Response 1	$\frac{9}{10}(90\%)$	$\frac{16}{21}(76\%)$
Response 2	$\frac{9}{10}(90\%)$	$\frac{7}{12}(58\%)$

Tabelle 7: Precision-Werte von MUC-6-Scorer und B-CUBED für die Beispiele aus Tabelle 6

Der tiefere Wert für Response 2 ist intuitiv richtig, da diese fünf zusätzliche Elemente zur Äquivalenzklasse des Keys hinzufügt im Gegensatz zu Response 1, die nur zwei zusätzliche Elemente als koreferent klassifiziert. Gegenüber dem MUC-6-Score bewertet B-CUBED die Precision also genauer, da das Ausmass von Fehlklassifikationen bestimmt und in die Berechnung miteinbezogen wird.

3.3 Constraint Entity Alignment F-Measure nach Luo

Luo (2005) erweiterte die beispielhaften Klassifikationsfehler von Bagga & Baldwin (1998) aus Tabelle 7 um eine Response 3, die alle Instanzen einer einzigen Klasse zuweist und um eine Response 4, die keine Äquivalenzklassen bildet (Tabelle 8). Für die Response 3 und 4 wurden MUC-Score und B-CUBED F-Score generiert und die Schwächen der Evaluations-Algorithmen aufgezeigt. Das MUC-6-Score ergibt für Response 3 einen Recall von 100%,

Key:	{1,2,3,4,5}	{6,7}	{8,9,A,B,C}
Response 1:	{1,2,3,4,5,6,7}		{8,9,A,B,C}
Response 2:	{1,2,3,4,5,8,9,A,B,C}	{6,7}	
Response 3:	{1,2,3,4,5,8,9,A,B,C,6,7}		
Response 4:	{1},{2},{3},{4},{5},{8},{9},{A},{B},{C}		

Tabelle 8: Erweiterte Klassifikationsfehler nach Luo (2005)

da alle Links zwischen den Instanzen gefunden werden. Die Precision ergibt 81,2% da 9 von 11 Links korrekt sind. Das F-Score beträgt schliesslich 90%. Für Response 4 kann kein MUC-6-Score berechnet werden, da keine Links generiert werden. Response 3 ergibt ebenfalls einen Recall von 100% für den B-CUBED-Algorithmus, da sich alle Äquivalenzklassen des Keys mit derjenigen der Response überschneiden. Das entspricht nicht dem intuitiv erwarteten Resultat, da die Äquivalenzklassen des Keys Teilmengen der Klasse der Response – und nicht umgekehrt – sind. Für Response 4 ergibt sich dasselbe Problem bei der Precision.

Gemäss Luo (2005, S. 29) liegen die aufgewiesenen, nicht intuitiven Resultate des MUC-Scores und des B-CUBED-Algorithmus daran, dass bei der Überschneidungs-Prozedur Mengen der Response und des Keys mehrmals verwendet werden können. Luo legte daher nahe, dass bei der Messung von Ähnlichkeit zwischen Response und Key alle Mengen jeweils nur einmal verwendet werden dürfen (daher *Constraint Entity Alignment*[43]). Für die Ähnlichkeit definierte Luo (2005) ein Mass auf der Ebene der Mentions[44] ($\phi_{Mentions}$), das prozentual beschreibt, wie viele Mentions richtig klassifiziert wurden. Für die Ebene der Äquivalenzklassen wird Ähnlichkeit ($\phi_{Entities}$) definiert durch den prozentualen Anteil der korrekt gefundenen Klassen. Für eine Menge R an Äquivalenzklassen einer Response und der Menge S der Klassen eines Keys ist $\phi_{Mentions}$ folglich die Kardinalität der Schnittmenge von R und S. $\phi_{Entities}$ benutzt die F-Score-Formel als Ähnlichkeitsmass für R und S[45].

$$\phi_{Mentions}(R, S) = |R \cap S|$$

$$\phi_{Entities}(R, S) = \frac{2 * |R \cap S|}{|R| + |S|}$$

[43] Luo (2005) verwendet im Gegensatz zur MUC-Terminologie *Entity* zur Bezeichnung von Koreferenzmengen.
[44] Luo (2005) benutzt *Mentions* als Synonym von *Markables*.
[45] Die Alignierung der Mengen von R und S wird als Maximum Bipartite Matching Problem betrachtet und nach dem Kuhn-Munkres-Algorithmus gelöst.

Im Vergleich der F-Scores der drei Masse für die beispielhaften Klassifikations-Fehler zeigte Luo (2005) auf, dass das CEAF-Mass der intuitiv unterschiedlichen Beurteilung der Fehler am nächsten kommt (Tabelle 9).

	MUC-6	B-CUBED	$\phi_{Mentions}$	$\phi_{Entities}$
Response 1	0.947	0.865	0.822	0.733
Response 2	0.947	0.737	0.583	0.667
Response 3	0.900	0.545	0.417	0.294
Response 4	–	0.400	0.250	0.178

Tabelle 9: F-Scores für MUC-6-Score, B-CUBED und CEAF nach Luo (2005, S. 29)

3.4 Das Ermitteln statistischer Signifikanz

Wie von Mitkov (2000) vorgeschlagen, werden Verfahren und Feature Sets oft gegenüber einer Baseline evaluiert. Um in statistischen Analysen die Signifikanz von Veränderung, bzw. Verbesserungen zu überprüfen, wird ein sogenannter *t-Test* (Gosset, 1908) durchgeführt. Dabei wird überprüft, ob die Verbesserungen auch durch Zufall hätten zustandekommen können.

Mit dem t-Test wird überprüft, ob die Nullhypothese, die besagt, dass die Ergebnisse vor und nach einer Manipulation nicht unterschiedlich sind, verworfen werden kann. Der t-Test betrachtet Mittelwerte, Standardabweichung und Stichprobenumfang in Datensätzen und liefert als Resultat einen p-Wert. Anhand eines festgelegten Signifikanzniveaus α und p wird ermittelt, ob Resultate statistisch signifikant sind. Ein weitgehend verbreiteter Wert für α ist 0.05 (5%). Das heisst, wenn $p < 0.05$ ist, werden verbesserte Resultate als statistisch signifikant interpretiert. Als Stichproben dienen in NLP-Anwendungen beispielsweise die Resultate von n-fachen Kreuzvalidierungen. Bei n = 10 werden die zehn Resultate der einzelnen Evaluationen als Proben genommen und miteinander verglichen. Oft wird ein sogenannter „gepaarter“ t-Test durchgeführt, da die einzelnen Resultate über den gleichen Datensätzen erzeugt werden und paarweise miteinander verbunden sind.

Diese Art von Überprüfung wird in Evaluationen von Systemen zur Koreferenz- und Anaphernauflösung benutzt, wenn durch Veränderungen eines verwendeten Verfahrens minimale Verbesserungen erzielt werden und festgestellt werden soll, ob sie aufgrund der Eigenschaften der verwendeten Korpus auch durch Zufall entstehen könnten.

Aufgrund der Nähe des CEAF-Masses zur intuitiven Beurteilung der beispielhaften Klassifikationsfehler, wird es in der vorliegenden Arbeit als Evaluationsmass für das im nächsten Kapitel implementierte System verwendet. Auch ist das CEAF-Mass in der aktuellen Forschung zur Koreferenz-Auflösung verbreitet (Klenner, 2007; Ailloud & Klenner, 2009; Denis & Baldridge, 2009).

4 Implementation eines Systems zur Koreferenz-Auflösung

Die aus den vorangehenden Kapiteln gewonnenen Erkenntnisse sollen nun als Grundlage für die Konzeption und Implementation eines ML-Systems zur Koreferenz-Auflösung dienen. Grundsätzlich orientiert sich das Verfahren am Feature Set von Soon *et al.* (2001). Um das Trainingskorpus zu balancieren[46], werden in Anlehnung an Strube *et al.* (2002); Hendrickx *et al.* (2007); Klenner (2007) diverse Filter angewendet. Die Verwendung von semantischen Klassen wie *Person, Location, Facility, Organization* etc. und semantischen Relationen aus Wortnetzen (Hyponymie, Meronymie etc.) als Merkmale ist in der Forschung weitgehend verbreitet. Ponzetto & Strube (2006) und Versley (2007a) führten die semantischen Ähnlichkeitsmasse als Merkmale bei der Auflösung von Bridging Anaphora ein. In der Evaluation des hier präsentierten Systems soll ermittelt werden, inwiefern solche semantische Merkmale nicht nur als Merkmale, sondern auch als harte Filter verwendet werden können und welche Auswirkungen diese zusätzliche Filterungen auf die Performanz des Classifiers haben. Die Hypothese ist, dass da im Vergleich mit den Pronomen nur ein kleiner Prozentsatz der nominalen Markables anaphorisch ist, durch Filterung das Ungleichgewicht zwischen negativen und positiven Instanzen im Trainingskorpus ausgeglichen werden kann. Dadurch soll hypothetisch eine Verbesserung des Classifiers erreicht werden.

Im folgenden Kapitel wird eine Übersicht über das verwendete Korpus *OntoNotes 1* (Hovy *et al.*, 2006) gegeben. Im Anschluss wird die Architektur des Systems in Abbildung 4 aufgezeigt und die Ressourcen besprochen, die das System benutzt. Parallel dazu wird auf Gemeinsamkeiten und Unterschiede zu bisherigen Systemen zur Koreferenz-Auflösung hingewiesen.

4.1 OntoNotes-Korpus

Grundlage für supervisierte ML-Verfahren sind Goldstandards mit annotierten Beispielen, anhand deren ein System lernt. Aus dem Goldstandard wird das Trainingskorpus generiert, das Beispiele für positive und negative Klassifikationen enthält. Der hier verwendete Goldstandard ist OntoNotes.

[46] Vgl. Kapitel 2.3 und Kapitel 2.6.2

Das OntoNotes-Korpus entstand aus einer Kollaboration zwischen BBN Technologies und den Universitäten von Colorado, Pennsylvania, und Southern California. Grundsatz von OntoNotes ist eine qualitativ robuste Ressource zu sein, d.h. mindestens einem *Inter-Annotator Agreement* von 90% zu entsprechen. Ausgangslage für OntoNotes bildete die Penn Treebank, aus der Parsing-Informationen extrahiert wurden, und die Penn Proposition Bank, aus der die Argument-Strukturen für Verben entnommen wurden. Nebst den syntaktischen Annotationen sind auch semantische Informationen kodiert, so z.B. Koreferenzmengen und Named-Entity-Klassifikationen.
Das Korpus ist auf den Wall Street Journal-Artikeln basierend gegliedert. Pro Artikel wird ein standartisiertes Format festgehalten, die *OntoNotes Normal Form* (ONF). Die ONF-Dateien enthalten pro Artikel eine ID, pro Satz den jeweiligen Parse-Baum und die Argument-Strukturen für die Verben. Am Ende jeder ONF-Datei sind die Koreferenzmengen gelistet (s. Abbildung 3), gefolgt von den Named Entity-Klassifikationen.

```
CHAIN: IDENT@10@wsj/00/wsj_0020@wsj@en@on
LINKS:
 IDENT@1:32:36@IDENT@10@wsj/00/wsj_0020@wsj@en@on: U.S. Trade Representative Carla Hills
 IDENT@3:0:1@IDENT@10@wsj/00/wsj_0020@wsj@en@on: Mrs. Hills
 IDENT@3:10:10@IDENT@10@wsj/00/wsj_0020@wsj@en@on: she
 IDENT@4:0:0@IDENT@10@wsj/00/wsj_0020@wsj@en@on: She
 IDENT@6:0:1@IDENT@10@wsj/00/wsj_0020@wsj@en@on: Mrs. Hills
 IDENT@7:15:15@IDENT@10@wsj/00/wsj_0020@wsj@en@on: she
 IDENT@10:35:36@IDENT@10@wsj/00/wsj_0020@wsj@en@on: Mrs. Hills
 IDENT@17:0:1@IDENT@10@wsj/00/wsj_0020@wsj@en@on: Mrs. Hills
 IDENT@18:0:0@IDENT@10@wsj/00/wsj_0020@wsj@en@on: She
 IDENT@19:5:6@IDENT@10@wsj/00/wsj_0020@wsj@en@on: Mrs. Hills
 IDENT@20:3:4@IDENT@10@wsj/00/wsj_0020@wsj@en@on: Mrs. Hills
```

Abbildung 3: Koreferenzmenge in der *OntoNotes Normal Form* (ONF).

Neben koreferenten NPs, sind auch koreferente Verben annotiert. Dies gilt für Verben, die in einer Wiederholung nominalisiert werden und Verben, die mit einer synonymischen NP für den gleichen Anlass koreferieren:

> (i) „Sales of passenger cars [grew]x 22%. [The strong growth]x followed year-to-year increases.“
> (ii) „Japan’s domestic sales of cars, trucks and buses in October [rose]x 18% from a year earlier to 500,004 units, a record for the month, the Japan Automobile Dealers’ Association said. [The strong growth]x followed year-to-year increases of 21% in August and 12% in September.“

Koreferente Verben werden im hier implementierten System nicht berücksichtigt. Nebst den IDENT-Koreferenzen, werden auch Appositionen in OntoNotes festgehalten. Markert &

Nissim (2005) folgend, werden für das vorliegende System Appositionen nicht extrahiert, da Appositionen Koreferenz strukturell herstellen[47] (Markert & Nissim, 2005):

> „We also exclude appositives, which provide coreference structurally and are therefore not anaphoric.“ (S. 21)

Für das System wird der Teil des Korpus benutzt, der Zeitungsartikel aus dem Wall Street Journal enthält (NWIRE-Korpus). Die eigentlichen Artikeltexte werden mit regulären Ausdrücken extrahiert und eventuelles Markup eliminiert. Die extrahierten Texte werden dann satzweise geparst, um Segmentierungsfehlern vorzubeugen. Werden ganze Texte geparst, kommt es vor, dass zwei Sätze als Einheit geparst werden, oder dass ein Satz in mehrere Sätze aufgeteilt wird. Solche Fehler treten beispielsweise auf, wenn eine Abkürzung wie „Calif.“ (für „California“) auftritt. Die Interpunktion wird in solchen Fällen fälschlicherweise als Satzgrenze erkannt. Vor dem Parsen wird deshalb mit einem Skript und einer Liste mit Abkürzungen versucht, solche Interpunktionen zu entfernen. Dennoch treten Segmentierungsfehler auf. Sie bewirkten eine Verschiebung der Satznummerierung, wodurch die Stand-Off-Annotation der Koreferenzmengen unbrauchbar wird. Von den insgesamt 597 Artikeln aus dem OntoNotes NWIRE-Korpus sind 195 von solchen Fehlern im Preprocessing betroffen und werden entfernt. Es bleiben folglich 402 Artikel im Goldstandard.

4.2 Preprocessing

Abbildung 4 gibt die Architektur des Systems wieder. Im Folgenden werden die einzelnen Schritte der Verarbeitung, inklusive der dabei verwendeten Werkzeuge und Ressourcen, ihrer Reihenfolge entsprechend präsentiert.

4.2.1 TTT2

Die Artikeltexte aus OntoNotes werden als erstes dem NLP-Toolkit *TTT2* (Grover, 2008) übergeben. TTT2 generiert das nötige Eingabeformat für das Parsing (Part-of-Speech-Tagging, Lemmatisierung, Chunking; s. Abbildung 6). Dazu verwendet TTT2 den *C&C*-Tagger (Curran & Clark, 2003) und den Lemmatiser *Morpha* (Minnen *et al.*, 2001). Der C&C-Tagger ist ein „state of the art“ Maximum Entropy-Tagger und erreicht in der Evaluation über der WSJ-PENN-Baumbank (Marcus *et al.*, 1999) einen F-Score von 97%. Morpha ist ein *Finite State Machine*-Morphologie-Tool und erreicht in einer Evaluation gegenüber

[47] Das Erkennen solcher Strukturen ist eine Aufgabe für das Parsing. Der verwendete Parser *Pro3Gres* eignet sich für das Erkennen von Appositionen nur bedingt.

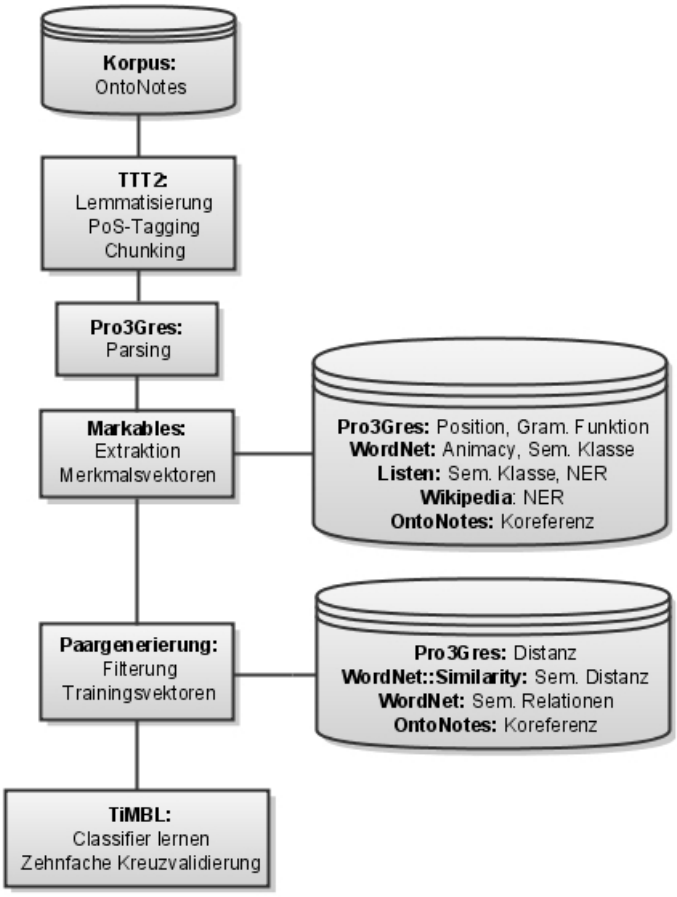

Abbildung 4: Pipeline des ML-Systems zur Koreferenz-Auflösung

der lexikalischen Datenbank CELEX (Baayen *et al.*, 1995) einen F-Score von bis zu über 99%. UMLS, Wikipedia, das Gutenberg-Projekt und die Berkley- und Alexandria Digital Library-Gazetteers wurden als Ressourcen zur Aufbereitung der statischen Daten von TTT2 verwendet. Eine typische TTT2-Pipeline zeigt Abbildung 5. Das Erkennen und Klassifizieren von Eigenamen (*nertag*) wird in der vorliegenden Arbeit ausgelassen, da NER während der Generierung der Merkmalsvektoren der Markables durch ein separates Modul vollzogen wird, um mit einer spezifischeren Klassifikation von NEs experimentieren zu können.

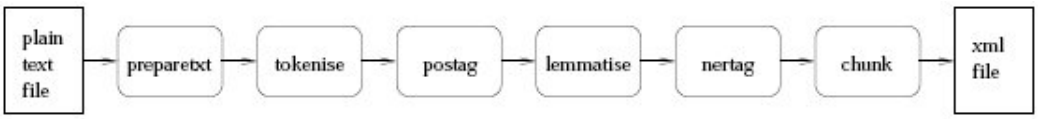

Abbildung 5: TTT2 Pipeline

```
sent(0,[
[dan, 'NNP', ['Dan_NNP'], w1],
[like, 'VBZ', [likes_VBZ], w5],
[mary, 'NNP', ['Mary_NNP'], w11],
[c, 'COMMA', [c_COMMA], w15],
[he, 'PRP', [he_PRP], w17],
[think, 'VBZ', [thinks_VBZ], w20],
[she, 'PRP', [she_PRP], w27],
[be, 'VBZ', ['\'s_VBZ'], w30],
[girl, 'NN', [a_DT, cute_JJ, girl_NN], w40],
['.', '.', ['._.'], w44]
]).
```

Abbildung 6: Ausgabe von TTT2 für den Beispielsatz: „Dan likes Mary, he thinks she's a cute girl."

4.2.2 Pro3Gres

Pro3Gres (Schneider, 2007) ist ein Dependenz-Parser und steht für *PRObability-based, PROlog-implemented Parser for RObust Grammatical Relation Extraction System*. Die Architektur von Pro3Gres ergibt sich wie folgt. Eine manuell erstellte *Functional Dependency Grammar* bestimmt aufgrund von PoS-Tags Dependenzen zwischen zwei aufeinanderfolgen Wörtern. Kern von Pro3Gres ist ein CYK-Chartparser[48], der mit einem Suchfenster Wort-Paare in die *Chomsky Normal Form*[49] kombiniert.
Jede vom CYK-Parser gefundene Chart wird mit einer lexikalischen Wahrscheinlichkeit versehen, die aus der PENN-Treebank gewonnen wird. Mit Pruning und verschiedenen Restriktionen werden unwahrscheinliche Charts eliminiert. Die Ausgabe von Pro3Gres besteht schliesslich aus Prolog-Prädikaten, die Dependenzen zwischen den Köpfen der Chunks denotieren (Abbildung 7). Schneider (2007, S. 171-211) gibt eine detaillierte Evaluation von Pro3Gres. Grundsätzlich erzielen Parser gute Precision- und Recall-Werte für Subjekt- und Objekt-Relationen (um 90%). Schwierigere Aufgaben sind die Disambiguierung von PP-Attachments und das Bestimmen von Dependenzen, die über mehrere Chunks hinweg bestehen; das heisst im Satz weit auseinander stehen, so z.B. Subjekte und indirekte Objekte.

[48] Vgl. z.B. Kasami (1965)
[49] Vgl. Chomsky (1956)

```
detmod1('girl#9','a#9',_,'(<-)', 0).
ncmod1('girl#9','cute#9',_,'(<-)', 0).

subj('like#2', 'dan#1', _G1719, '(<-)', 0).
obj('like#2', 'mary#3', _G1867, '(->)', 0).
comma('think#6', 'c#4', _G2091, '(<-)', 0).
subj('think#6', 'he#5', _G2254, '(<-)', 0).
subj('be#8', 'she#7', _G2460, '(<-)', 0).
obj('be#8', 'girl#9', _G2600, '(->)', 0).
sentobj('think#6', 'be#8', 'she#7', '(->)', 0).
bridge('like#2', 'think#6', 'c#4', '(->)', 0).
```

Abbildung 7: Ausgabe von Pro3Gres für den Beispielsatz: „Dan likes Mary, he thinks she's a cute girl."

Ein Vorteil des Depedenz-Parsens ist für die Koreferenz-Auflösung, dass anhand der Pro3Gres-Prolog-Prädikate leicht ermittelt werden kann, ob zwei Markables vom gleichen Verb abhängen und als mögliche koreferente Einheiten eliminiert werden können; ausser die Anapher ist ein Reflexiv- oder Possessivpronomen.

4.3 Generierung der Merkmalsvektoren

Ist das Parsen abgeschlossen, kann mit der Generierung der Merkmalsvektoren der Markables begonnen werden. Dabei werden die Markables in der Ausgabe von TTT2 identifiziert und extrahiert. Anschliessend werden die Merkmalsvektoren anhand weiterer Ressourcen aufgebaut.

4.3.1 Extraktion der Markables

Die Generierung der Merkmalsvektoren für die Markables wird pro Artikel im Goldstandard durchgeführt. Die Markables in den Texten werden anhand der folgenden Part-of-Speech-Tags aus der Ausgabe von TTT2 (vgl. Abbildung 6) extrahiert:

- **NN**: Nomen (NNS für Plural)
- **NNP**: Eigennamen (NNPS für Plural)
- **PRP**: Pronomen, Reflexivpronomen
- **PRP$**: Possessivpronomen

Die Extraktion verläuft jeweils pro Chunk, wobei als erstes der Kopf des Chunks betrachtet wird. Ein Chunk enthält neben dem Kopf oft zusätzliche Markables. Deshalb wird der gesamte Chunk nach folgenden Kriterien untersucht:

- Possessivpronomen sind selten Kopf des Chunks. Ausser in Fällen wie „I saw a friend of mine“[50]. Possessivpronomen müssen aus den Chunks extrahiert werden, da sie Teil von Koreferenzmengen sind.
- Koordinationen („and_CC“), Disjunktionen („or_CC“) oder Kommata-Auflistungen („c_COMMA“) enthalten oft Markables. In solchen Fällen werden die koordinierten NPs getrennt, rekursiv untersucht und allenfalls extrahiert.
- Chunks mit Genitiv-Konstruktionen (z.B. „Peter’s mother“) werden am Genitiv-Suffix getrennt und rekursiv untersucht.

Die Markable-Extraktion aus koordinierten NPs gestaltet sich schwierig, da die koordinierten NPs einzeln, alle oder nur ein Teil von ihnen in Koreferenzmengen auftreten können. Ein Ansatz wäre zwei Gruppen von Markables zu extrahieren: Die einzelnen NPs mit individuellem Numerus und die koordinierte NP mit Numerus Plural. Das Problem kann an folgendem Beispielsatz verdeutlicht werden.

> „Peter’s mother and her friend sent him a postcard they bought during their holidays.“

Hier muss der Chunk „Peter’s mother and her friend“ in „Peter“ und „mother and her friend“ segmentiert werden, um „Peter“ auf „him“ und „mother and her friend“ auf „they“ Genus- und Numerus-kongruent abzubilden. „their“ ist insofern ambig, als dass es sich auf den gesamten Chunk oder nur auf „mother and her friend“ beziehen kann. Dass „they“ sich nur auf „mother and her friend“ bezieht, ergibt sich nur aus dem Weltwissen, dass man üblicherweise niemandem eine Postkarte schickt, die man zusammen gekauft hat. Werden nun alle möglichen Kombinationen von einzelnen und koordinierten NPs extrahiert, entstehen viele Markables, die nicht in Koreferenzmengen sind und folglich ergeben sich bei der Paargenerierung viele negative Beispiele. Um dies zu verhindern, werden im vorliegenden System koordinierte NPs segmentiert und die einzelnen NPs separat extrahiert. Das bedeutet, dass das System Anaphern mit Numerus Plural nicht auf koordinierte Antezedenzien auflösen kann.

[50] Für solche Konstruktionen liefert TTT2 fehlerhafte Ausgaben. „mine“ wird als Nomen (NN) getaggt, „hers/yours“ als Nomen Plural (NNS) und „ours“ als Adjektiv (JJ)

4.3.2 Feature Set und Generierung der Merkmalsvektoren

Bei der Extraktion werden in Anlehnung an das Standard-Feature Set von Soon *et al.* (2001) folgende 15 Merkmale für die Markables generiert. Einige davon dienen als Merkmale für den Classifier, andere werden als harte Filter bei der Generierung von möglichen Anapher-Antezedens-Paaren verwendet. Das Feature Set unterscheidet sich grundsätzlich vom von Soon *et al.* (2001) verwendeten bzgl. der festgehaltenen Verb-Dependenz und der spezifischen Abbildung von NEs auf Nomen aus WordNet.

- **Positionelle Merkmale:** Drei Merkmale – Satznummer, Markable-ID und Position im Satz – halten positionelle Merkmale fest. Die Sätze sind pro Artikel von Null an nummeriert. Die Markable-IDs werden von links nach recht fortlaufend und satzübergreifend im Text vergeben. Die Position im Satz ergibt sich aus dem Listenindex des Chunks, zu dem ein Markable gehört.

- **Part-of-Speech:** Die Wortarten (PoS) der Markables werden festgehalten, da die Generierung der möglichen Koreferenz-Paare PoS-basiert funktioniert. Dabei werden nicht immer alle Merkmale instantiiert (z.B. gibt es keine semantische Klasse für Pronomen). Die PoS-Angaben spielen bei der Paargenerierung eine zentralle Rolle, da für verschiedene Kombinationen von Wortarten verschiedene Merkmale berechnet werden. (z.B. macht das Ermitteln von String Matching oder der Levenshtein-Distanz zwischen Named Entities und Pronomen keinen Sinn, etc.)

- **Person:** Die Angabe gibt die Person (1.,2.,3.) des Markables wieder. Das Merkmal wird bei der Paargenerierung als harter Filter verwendet (ausser in Fällen, wo direkte Rede auftritt).

- **Numerus:** Numerus ergibt sich bei Nomen und Named Entities direkt aus den PoS-Tags. Bei Pronomen wird Numerus anhand von Listen bestimmt (z.B. „they“ → Plural).

- **Animacy:** Hat fünf mögliche Werte, von denen zwei Genus spezifisch sind: „animate“ („manimate“, „fanimate“), „inanimate“, „ambiguous“. Bei den Pronomen wird die Belebtheit wieder direkt anhand von Listen ermittelt („she“ → „fanimate“). Ein Spezialfall sind „they/their“, die sich sowohl auf belebte, wie unbelebte Antezedenzien beziehen können. Hier wird der Wert „ambiguous“ vergeben.
  Für die Bestimmung der Belebtheit von Nomen wird in Anlehnung an Evans & Orăsan (2000) in WordNet überprüft, ob sie vom Synset *Person* über eine Hyponymie-Relation (indirekt) subsumiert werden. Bei Eigennamen wird die Belebtheit anhand von

Namenslisten und Wikipedia ermittelt. Eine genauere Beschreibung der Eigennamen-Klassifikation gibt Kapitel 4.3.3.

- **Chunk:** Der Chunk wird im Zusammenhang mit der Wikipedia-Klassifikation von Eigennamen benutzt und im Merkmalsvektor festgehalten.

- **Grammatikalische Rolle:** Anhand der Ausgabe von Pro3Gres wird ermittelt, welche grammatikalische Rolle ein Markable im Satz übernimmt. Diverse Untersuchungen haben gezeigt, dass die Parallelität der grammatikalischen Rollen einer Anapher und eines möglichen Antezedens ein starker Indikator für Koreferenz ist (Lappin & Leass, 1994; Mitkov *et al.*, 2002). Wird kein Wert gefunden, wird „nogf“ vergeben.

- **Artikel/Determiner:** Anhand der Pro3Gres-Ausgabe wird der Artikel von Nomen bestimmt („nodet“, wenn kein Artikel vorhanden ist, sonst „def“/„indef“ oder „dem“ für demonstrative Artikel).

- **Abbildung auf Nomen:** Dieses Merkmal ist nur für Eigennamen relevant. Es gibt an, auf welches in WordNet enthaltene Nomen ein Eigenname durch die Wikipedia-Klassifikation abgebildet wurde (z.B. „Dorothy L. Sayers“ → *writer*). Bei allen anderen Wortarten, oder wenn kein entsprechendes Nomen gefunden wurde, wird „nosemtype“ als Platzhalter vergeben; vgl. Kapitel 4.3.3.

- **Semantische Klasse:** Für Eigennamen und Nomen wird in Anlehnung an gebräuchliche Verfahren in der Forschung eine übergeordnete semantische Klassse festgehalten (*Person, Organization, Location, Object*). Dazu werden werden in den Chunks nach gewissen Schlüsselwörtern (z.B. „city“ → Location, „manufacturer“ → Organization) gesucht und die Hyponymie-Relationen in WordNet betrachtet.

- **Verb-Dependenz:** Über der Ausgabe von Pro3Gres wird die Verb-Dependenz des Markables bestimmt.

- **Kopf-String:** Hält den String des Kopfs des Chunks fest, in dem das Markable enthalten ist.

- **Markable-String:** Hält den String des Markables fest.

- **Koreferenz-ID:** Hält fest, ob das Markable gemäss OntoNotes in einer Koreferenzmenge ist und falls ja, in welcher.

Die Verteilung der generierten Markables entsprechend ihrer Wortarten zeigt Tabelle 10. Aus ihr geht hervor, dass die meisten Markables Nomen sind (69.2%). Von ihnen sind aber nur ein Fünftel (20.5%) – im Vergleich zu den anderen Wortarten am wenigsten – in einer Koreferenzmenge. Durch ihre hohe Anzahl machen sie dennoch 41.1% aller koreferenten Markables aus.

Wortart	*Anzahl*	*Davon koreferent*	*% aller Koref.*
Markables insgesamt	41679 (100%)	14406 (34.6%)	100%
Nomen	28846 (69.2%)	5920 (20.5%)	41.1%
Eigennamen	8897 (21.3%)	5047 (56.7%)	35.0%
Pronomen	2647 (6.4%)	2258 (85.3%)	15.7%
Poss./Refl.-Pronomen	1289 (3.1%)	1181 (91.6%)	8.2%

Tabelle 10: Verteilung der Wortarten im Trainingskorpus

Ein Beispiel eines Merkmalsverktors gibt Tabelle 11. Im Folgenden wird das Verfahren diskutiert, das Named Entities über Wikipedia auf Nomen abbildet.

Satznummer	Markable-ID	Index	Part-of-Speech	Person	Numerus	Animacy	Chunk	Gram. Rolle	Artikel	Abbildung auf Nomen	Semantische Klasse	Verb-Dependenz	Chunk-Kopf	Markable	Koreferenz-ID
0	1	1	nnp	3	sg	inanimate	'The_DT', 'U.S._NNP'	subj	the	country	location	['remove#8', 0]	'u.s.'	'U.S.'	120020.

Tabelle 11: Merkmalsvektor für *"The U.S."*

4.3.3 Named Entity-Klassifikation mit Wikipedia

Die Generierung der Merkmalsvektoren für NEs wird im vorliegenden System von einem eigenen Modul ausgeführt, da auch zwischen NEs und Nomen semantische Ähnlichkeitsmasse berechnet werden sollen. Dazu müssen die NEs auf Nomen aus WordNet abgebildet werden. Bei allen NEs, die aus mehreren Wörtern bestehen, wird zuerst anhand des Kopfs des Markable-Chunks[51] zu ermitteln versucht, ob es sich um eine Firma handelt. Das System enthält eine Liste mit gängigen Abkürzungen in Firmennamen (z.B. „Corp.", „Co.", „Inc.",

[51] Die Chunks entsprechen hier bei koordinierten NPs schon den einzelnen NPs. Diese können auch aus mehreren Wörtern bestehen.

„Ltd.“ etc.). Besteht das Markable aus nur einem Wort, wird mit Listen mit Kontinenten, Ländern, Hauptstädten, Zeiteinheiten (Monate, Wochentage etc.), weiblichen und männlichen Vornamen und Demonymen eine Klassifikation versucht. Ist eine Klassifikation anhand dieser Listen nicht möglich, setzt die Abbildung der NE auf ein Nomen über Wikipedia ein. In Anlehnung an Kazama & Torisawa (2007) wird ein Verfahren implementiert, das NEs anhand ihrer „Definitionssätze“ in den entsprechenden Wikipedia-Artikeln Nomen aus WordNet zuweist.

Ein oft genannter Vorteil von Wikipedia ist das ständige Hinzukommen neuer Artikel, das ein stetes Wachstum des abgebildeten Wissen bedeutet. Die meisten NLP-Systeme, die Wikipedia als Ressource verwenden, arbeiten mit einer statischen Version der Enzyklopädie[52]. Um dem ständigen Wachstum von Wikipedia beizukommen, operiert das vorliegende Verfahren auf Basis der Online-Version von Wikipedia. Ein entsprechender Artikel zu einer NE wird folgendermassen gesucht (Beispiel „U.S. Trade Reprasentative Carla Hills“).

- Kazama & Torisawa (2007) finden den entsprechenden Artikel, in dem sie die Wörter der NE mit „_“ konkatenieren und so den Identifikator für den Artikel generieren („Jimi Hendrix“ → „Jimi_Hendrix“). So können auch URLs generiert werden, die zu den Artikeln der Online-Version von Wikipedia führen. Artikel für komplexere NPs (wie das Beispiel) können so aber direkt nicht gefunden werden. Der Chunk

  ```
  ['U.S._NNP', 'Trade_NNP', 'Representative_NNP', 'Carla_NNP', 'Hills_NNP']
  ```

  wird von Markup bereinigt und eine entsprechende Google-Anfrage mit „wikipedia“ und der NE generiert:

  http://www.google.com/search?hl=en&as_qdr=all&q=wikipedia+U.S.+Trade+Representative+Carla+Hills

  Mit dem Parameter „&as_qdr=all“ wird sichergestellt, dass alle Suchbegriffe berücksichtigt werden. Die entsprechende Resultat-Seite von Google wird heruntergeladen. Grund für das Auffinden von Wikipedia-Artikeln über Google ist die realtiv unflexible Suchmaschine von Wikipedia („U.S. Trade Reprasentative Carla Hills“ liefert keine Resultate, da „Repr**a**sentative“ im Artikel falsch geschrieben ist. Die korrigierte Anfrage liefert als erstes Resultat einen Artikel über „Office of the United States Trade Representative“).

- Aus den Google-Resultaten wird mit regulären Ausdrücken der oberste Wikipedia-Link extrahiert, der Bestandteile der Named Entity enthält („Carla“ in

[52] Regelmässige Images von Wikipedia können heruntergeladen werden unter http://download.wikipedia.org/ (Stand 1.9.2009)

„http://en.wikipedia.org/wiki/Carla_Anderson_Hills“). Wenn keine Suchresultate von Google zurückgeliefert werden, wird die Wikipedia-Suche abgebrochen.

- Der gefundene Wikipedia-Artikel wird heruntergeladen und der erste Abschnitt, der den „Definitionssatz“ enthält, anhand von zwei regulären Ausdrücken, die die HTML-Struktur von Wikipedia-Artikeln berücksichtigen und dem Muster

  `'.*(is|are|was|were|has been|have been|serves as|served as|became).*? (a|an|the|one).*'`

  extrahiert. Inhalte in Klammern werden im Artikeltext entfernt, um die folgenden Schritte zu vereinfachen[53]. Es wird nochmals überprüft, ob der gefundene Abschnitt mindestens einen Teil der NE vor dem gefundenen Verb aus dem Suchmuster enthält.

  > „Carla Anderson Hills is an American lawyer and public figure. She served as United States Secretary of Housing and Urban Development in the Gerald Ford administration, and as U.S. Trade Representative. She was the first woman to serve as Secretary of Housing and Urban Development and the third woman to serve as a Cabinet officer in a U.S. Presidential Administration.“

- Der extrahierte Abschnitt wird von Pro3Gres geparst. In der Parser-Ausgabe wird zuerst im ersten, dann im zweiten Satz nach dem Objekt eines der Verben „be“, „serve“, „become“ gesucht. Beim gefundenen Objekt (hier „lawyer“) wird überprüft, ob es sich um ein Nomen oder um eine weitere NE handelt.

- Der Chunk des gefundenen Objekts wird auf Koordinationen („and“) und Kommata-Auflistungen untersucht. Sind solche vorhanden, werden die koordinierten Nomen einzeln extrahiert.

- Es wird überprüft, ob das gefundene Nomen in WordNet vorhanden ist. Handelt es sich um ein Mehrwort-Token, wird versucht, die längste Entsprechung in WordNet zu finden, da diese am spezifischsten ist. Eine möglichst spezifische WordNet-Entsprechung ermöglicht eine möglichst genaue Bestimmung der semantischen Ähnlichkeit (z.B. wird „mystery novel“ gegenüber „novel“ bevorzugt etc.).

- Zusätzlich liefert der gefundene Definitionsabschnitt die Möglichkeit, die Belebtheit und das Genus einer NE zu bestimmen. Es wird untersucht, ob „Mr“/„Mrs“ oder ein weiblicher oder männlicher Vorname im entsprechenden Chunk enthalten ist. Stärkster

[53] Gewisse Wikipedia-Artikel weisen inkonsistente Klammerungen auf, d.h. z.B. schliessende Klammern fehlen. In solchen Fällen werden die Klammern belassen.

Indikator für die Belebtheit ist das Vorhandensein von „he“/„she“ in Subjekt-Position, bzw. „him“/„her“ in Objekt-Position in den ersten zwei Sätzen des Abschnitts.

- Kann durch Wikipedia für eine NE eine Nomen-Entsprechung in WordNet gefunden werden, so wird sie in einer Datei abgelegt, die gefundene Entsprechungen festhält.

Schlägt auch die Klassifikation durch Wikipedia fehl, wird überprüft, ob das letzte Wort des Chunks ein in WordNet enthaltenes Nomen ist (z.B. „St. Paul's Cathedral“ → „cathedral“) und wird allenfalls als Abbildung genommen. Wird kein Nomen gefunden, wird als letztes im Chunk nach weiblichen oder männlichen Vornamen gesucht. Ist die Suche erfolglos, wird der Wert „nosemtype“ im Merkmalsvektor abgelegt und angenommen, dass das Markable unbelebt ist. Wenn alle Markables und deren Merkmale in den Vektoren abgelegt sind, kann mit der Erstellung des Trainingskorpus begonnen werden.

4.3.4 Generierung der Trainingsvektoren

Bei der Generierung der Trainingsvektoren werden, den Ansätzen von beispielsweise Strube *et al.* (2002); Klenner (2007); Hendrickx *et al.* (2007); Ailloud & Klenner (2009) folgend, gewisse Restriktionen etabliert, um dem Problem von unausgewogenen Trainingskorpora entgegenzuwirken; sprich die Anzahl von negativen Beispielen möglichst gering zu halten. Um den verschiedenen Eigenschaften von Koreferenz-Relationen zu entsprechen, werden die Trainingsvektoren anhand von Regeln ind Filtern, basierend auf den PoS-Tags der Markables, generiert. Die Trainingsvektoren werden jeweils für ein mögliches Antezedens-Anaphern-Paar (i und j) erstellt und enthalten 33 Merkmale, die hier zusammenfassend wiedergegeben werden.

- **Distanz:** Distanz zwischen i und j in Anzahl Markables und Sätzen.
- **Grammatikalische Rollen:** Hält die grammatikalischen Rollen von i und j fest und ob sie identisch sind.
- **Part-of-Speech:** Enthält die PoS-Tags von i und j.
- **Markables-Strings:** Enthält die Strings von i und j.
- **Levenshtein-Distanz:** Berechnet die Levenshtein-Distanz im Sinne von Strube *et al.* (2002) zwischen den Strings von i und j (nur wenn die PoS-Tags von i und j identisch sind).

- **Semantische Merkmale:** Die semantischen Merkmale können nur zwischen Nomen, bzw. NEs berechnet werden. Bei ambigen Nomen, die mehrere Entsprechungen in WordNet haben, werden alle Synsets extrahiert und betrachet. Wird eine semantische Relation mit einem der Synsets gefunden, gilt sie als vorhanden.
  - **Ident:** Hält fest, ob i und j identisch sind (String Match).
  - **Syn:** Hält fest, ob i und j ein gemeinsames Synset teilen.
  - **Hyp:** Hält fest, ob eine Hyponymie-Relation zwischen i und j besteht (gegebenenfalls durch Rekursion).
  - **Gloss:** Hält fest, ob i in einer Glosse von j ist und umgekehrt.
  - **MM:** Hält fest, ob eine Meronymie-Relation zwischen i und j besteht (gegebenenfalls durch Rekursion).
  - **Semantische Ähnlichkeit:** Die weiteren Merkmale enthalten alle im Kapitel 2.4.4 diskutierten Ähnlichkeitsmasse zwischen i und j.
- **Koreferenz:** Das letzte Merkmal hält fest, ob der Trainingsvektor ein positives oder negatives Beispiel ist.

Während der Generierung der Trainingsvektoren werden folgende harte Filter auf i und j angewendet.

- Die Distanz zwischen i und j darf drei Sätze nicht überschreiten. Lappin & Leass (1994) verwenden für die intersententielle Pronomenauflösung nur den vorangehenden Satz; Mitkov *et al.* (2002) verwenden ein Satzfenster mit Grösse zwei; Hendrickx *et al.* (2007) betrachten für die Pronomen- und Nomenauflösung den Kontext von drei Sätzen. Ausnahme sind Konstellationen, in denen i und j beides Named Entities sind. In diesem Fall gelten keine Restriktionen bezüglich der Distanz.
- i und j müssen bzgl. Person, Numerus und Belebtheit unifizieren. Bei der Unifikation ist zulässig, dass ein Genus unspezifisches Markable („animate“; z.B. „doctor“) mit einem Markable unifiziert, das ein spezifischen Genus hat („fanimate“/„maninmate“; z.B. „she/he“). Pronomen im Plural („they“) können mit belebten und unbelebten Markables im Plural unifizieren. Die Unifikation von unterschiedlichen Werten im Person-Merkmal ist dann zulässig, wenn direkte Rede vorliegt („Peter said: ’I like Mary.’“). Mit einer einfachen Heuristik wird in solchen Fällen untersucht, ob zwischen i und j ein Anführungszeichen vorhanden ist.
- Wenn j eine nominale Anapher ist, darf sie nicht indefinit sein; s. Vieira & Poesio (2000); Strube *et al.* (2002); Hendrickx *et al.* (2007).

- Markables, die vom gleichen Verb c-kommandiert werden, sind exklusiv. (Klenner (2007, S. 3) definiert dafür das Prädikat *clause_bound*). Ausnahme sind Reflexiv- und Possessivpronomen. Auf sie wird der Filter nicht angewendet, da sie vom selben Verb wie ihr Antezedens abhängen können.

Wird das Trainingskorpus mit diesen Filtern generiert, so besteht es aus insgesamt 98205 Trainingsvektoren, wovon 8389 positive Beispiele sind (8.54%). Soon *et al.* (2001) arbeiteten mit 6.5%, bzw. 4.4% positiven Beispiele in MUC-6 und MUC-7, mit insgesamt 20910, respektive 48872 Trainingsvektoren. Ng & Cardie (2002c) reproduzierten Soon *et al.* (2001)'s System als Baseline und verwendeten dieselben Korpora. Strube *et al.* (2002) arbeiten nach einer Filterung mit 72093 Trainingsvektoren aus einem eigenen Goldstandard. Hendrickx *et al.* (2007) verwenden 76920 Trainingsvektoren aus dem KNACK-Korpus, wovon 8.5% positive Beispiele sind. Durch Filterung wird das Korpus um maximal 91.7% reduziert[54]. Ailloud & Klenner (2009) arbeiten nach einer Filterung mit 60414 Trainingsvektoren aus dem TüBa-Korpus.

Anhand des generierten Trainingskorpus wird nun mit TiMBL ein Classifier trainiert. Dabei werden die Standardwerte und -Parameter verwendet und ein *k*-Wert von fünf gesetzt. Die Evaluation des Systems gibt das folgende Kapitel.

[54]Hendrickx *et al.* (2007) geben nicht an, wie viele positive Beispiele durch die Filterung eliminiert werden. Sie halten fest (Hendrickx *et al.*, 2007, S. 11), dass ein (kleiner) Teil der positiven, und ein grosser Teil der negativen Beispiele eliminiert wird.

5 Evaluation und Experimente zur Auflösung nominaler Anaphora

Im Folgenden wird eine Evaluation der verschiedenen für das System implementierten Komponenten vorgenommen und insbesondere die Leistung bezüglich der Auflösung von Bridging Coreference untersucht.

5.1 Named Entity-Klassifikation

Die Named Entitity-Klassifikation mit dem im Kapitel 4.3.3 diskutierten Verfahren, das NEs über Wikipedia auf Nomen aus WordNet abbildet, ist zentraler Bestandteil des Systems. Eine qualitative Evaluation ist schwierig, da die Abbildungen auf die Nomen nicht automatisch evaluiert werden können. Es werden hier einige quantitative Angaben gegeben.

Das Trainingskorpus generiert für 8899 Named Entities Merkmalsvektoren. Für 559 (6.7%) davon kann keine Abbildung auf WordNet gemacht werden (d.h. Wert „nosemtype“ im Merkmalsvektor). Von den übrigen 8340 NEs werden 4258 (51.1%) anhand des Wikipedia-Algorithmus und 4082 (48.9%) anhand der Listen klassifiziert (s. Tabelle 12).

NEs	**8899**	**100%**
Wikipedia	4258	47.8%
Listen	4082	45.9%
Keine Klass.	599	6.7%

Tabelle 12: Verteilung der Verfahren zur Named Entity-Klassifikation

Tabelle 13 gibt die Verteilung der Klassifikationen anhand von Listen wieder. Der hohe Anteil an *Companies* im Vergleich zu *Person* ergibt sich dadurch, dass Personen oft durch die Wikipedia-Klassifikation erfasst werden. Der Wikipedia-Algorithmus klassifiziert 1285 (30.2% aller Wikipedia-Klassifikationen) NEs als *Person*. Insgesamt ergeben sich daraus 1925 *Person*-Klassifikationen (23.1% aller klassifizierten NEs).

Listen	**4082**	**100%**
company	1464	35.9%
country	785	19.2%
person	640	15.7%
Letztes Nomen als Klasse	350	8.6%
capital	242	5.9%
month	234	5.7%
day	215	5.3%
continent	101	2.5%
nation	51	1.2%

Tabelle 13: Verteilung der Named Entitiy-Klassifikationen, die durch Listen vorgenommen wurden

5.1.1 Verteilung der semantischen Klassen

Nomen und NEs werden einer der vier übergeordneten semantischen Klassen (*Person, Organization, Location, Object*) zugeteilt (Vgl. Kapitel 4.3.2). Die Verteilung der Klassen zeigt Tabelle 14. Dabei ist festzuhalten, dass *Object* auch als „Auffang-Klasse“ für NPs gilt, die keiner anderen Klasse zugewiesen werden konnten.

Nomen + NEs	**37743**	**100%**
Object	24839	65.8%
Person	5570	14.8%
Organization	4342	11.5%
Location	2921	7.7%

Tabelle 14: Verteilung der semantischen Klassen

5.2 CEAF- und MUC-6-Resultate

Die Evaluation des vorliegenden Systems wird über allen Markables durchgeführt. Einige Verfahren setzten voraus, dass alle koreferenten Einheiten (True Mentions) bekannt sind und evaluieren nur über diesen. Dadurch verbessern sich die Resultate gerade bei der Auflösung nominaler Anaphora, da nur ein kleiner Teil der Nomen anaphorisch ist (in OntoNotes 20.1%, vgl. Tabelle 10, S. 68). Herauszufinden, ob ein Markable anaphorisch ist, ist eine schwierige Aufgabe (S. Kapitel 2.6.1). Da das vorliegende System in einem realen Setting über freien Texten verwendet werden können sollte, wird die Evaluation über allen Markables gemacht.

Für die Evaluation wird eine zehnfache Kreuzvalidierung durchgeführt. Dabei wird jeweils ein Zehntel der Artikel aus dem Goldstandard als Testkorpus und die anderen als Trainingskorpus verwendet. Dann wird entsprechend zehn Mal evaluiert und die Durchschnittswerte ermittelt. Die Resultate gibt Tabelle 15. Die besten Gain Ratios[55] erreichen gemäss Tabelle 16 die

	Recall	*Precision*	*F-Score*
CEAF	55.9%	61.9%	58.7%
MUC-6	49.3%	55.6%	52.2%

Tabelle 15: CEAF und MUC-6-Resultate der zehnfachen Kreuzvalidierung

Merkmale *Ident* und *Synonymie*. Aus Tabelle 10 zur Verteilung der Wortarten bezüglich der koreferen Einheiten geht hervor, dass die meisten True Mentions Nomen sind (69.2%). Folglich spielen Merkmale, die nominale Anaphora charakterisieren, die entscheidendste Rolle im Classifier. Schon der Entscheidungsbaum von Soon *et al.* (2001)[56] zeigte, dass String Matching-Verfahren einen Grossteil der nominalen Koreferenzen auflösen können. Versley (2007a) bestätigte, dass durch String Matching ein grosser Teil (fast 50%) der nominalen Anaphern aufgelöst werden. Auch die Einführung der Levenshtein-Distanz – ebenfalls ein Mass, das anhand von String Matching verfährt – zeigte bei Strube *et al.* (2002) eine enorme Verbesserung in der Auflösung nominaler Anaphora. Für die verbleibenden nominalen True Mentions zeigte z.B. Versley (2007a) auf, dass sie anhand von semantischen Merkmalen wie Hyponymie, Synonymie und Meronymie aufgelöst werden können.

Diese Befunde zeigen sich auch hier in der starken Gewichtung der semantischen Merkmale gegenüber den grammatikalischen, positionellen und syntaktischen Merkmalen. Dazu ist anzumerken, dass beispielsweise die Distanz in Sätzen als harter Filter verwendet wird und dass nur Paare betrachtet werden, die diesen Filter passieren. Würde dieser Filter nicht verwendet, würde das Distanzmerkmal zu einem stärkeren Indikator werden, da sich viele nicht koreferente Paare ergäben, die weit auseinander stünden.

Ansätze in regelbasierten Systemen zur Auflösung von pronominaler Anaphora zeigten, dass positionelle Merkmale und die grammatikalischen Rollen wichtige Indikatoren für mögliche Antezedenzien sind (Hobbs, 1976; Lappin & Leass, 1994; Mitkov, 1998). Diese Merkmale werden vom vorliegenden Classifier als relativ unbedeutend beurteilt. Erst wenn alle Paarvek-

[55] Die Gain Ratios sind Indikatoren dafür, wie nützlich die verschiedenen Merkmale für den Classifier sind (vgl. Kapitel 2.2.1).
[56] S. Abbildung 1, S. 10

Merkmal	*Anz. Werte*	*Information Gain*	*Gain Ratio*
Ident	4	0.10333	0.06045
Synonymie	4	0.10328	0.06021
In Glosse	4	0.07966	0.04406
Levenshtein-Distanz	20	0.06174	0.03924
Sem. Klasse j	6	0.05470	0.03780
Sem. Relation vorhanden	3	0.05672	0.03644
Meronymie	4	0.05435	0.03546
Hyponymie	4	0.05433	0.03497
Sem. Klassen identisch	3	0.05494	0.03472
PoS j	6	0.04474	0.03445
PoS i	6	0.04726	0.03290
Gloss Context Vectors	11	0.04959	0.02614
Jiang & Conrath	95	0.03120	0.02538
Resnik	13	0.04969	0.02465
Wu & Palmer	10	0.05266	0.02422
Lin	11	0.04611	0.02391
Sem. Klasse i	6	0.04697	0.02328
Path	16	0.05301	0.02142
Leacock & Chodorow	19	0.05302	0.02117
Lesk	601	0.04234	0.02094
Hirst & St-Onge	8	0.02085	0.01968
String i	5841	0.17957	0.01709
String j	2651	0.11099	0.01352
Gram. Rolle i	10	0.03317	0.01272
Parallelität gram. Rollen	3	0.01561	0.01178
Distanz in Sätzen	4	0.00954	0.00478
Gram. Rolle j	10	0.01095	0.00450
Distanz in Markables	50	0.01026	0.00332
Artikel/Definitheit j	3	0.00108	0.00114
Artikel/Definitheit i	4	0.00110	0.00098

Tabelle 16: Anzahl unterschiedlicher Werte, Information Gain und Gain Ratio der verschiedenen Merkmale in TiMBL (nach Gain Ratio sortiert)

toren aus dem Trainingskorpus gelöscht werden, die keine Pronomen enthalten, und dann ein Classifier trainiert wird, erhalten diese Merkmale eine stärkere Gewichtung. Tabelle 17 gibt die Gain Ratios bei der pronominalen Anaphernauflösung (inklusive Reflexiv- und Possessiv-Pronomen) im vorliegenden System wieder. Nebst der Wortart des Antezedens spielt dessen String eine wichtige Rolle. Das lässt sich auf Fälle zurückführen, in denen Antezedens und Anapher beides Pronomen sind und deren Strings identisch sind[57]. TiMBL bestimmt die Ähnlichkeit von nicht numerischen Merkmalen anhand eines adaptierten Levenshtein-Masses (s. Kapitel 2.2). Das String Matching fliesst so in den Entscheidungsprozess des Classifiers ein.

[57] Vgl. z.B. das Beispiel aus dem Goldstandard in Abbildung 3, S. 60

Merkmal	*Anz. Werte*	*Information Gain*	*Gain Ratio*
PoS i	6	0.07611	0.04471
String i	4648	0.33904	0.03092
Gram. Rolle i	9	0.08454	0.03379
Distanz in Sätzen	4	0.02757	0.01392
String j	19	0.03958	0.01371
Artikel/Definitheit i	4	0.00997	0.00771
Distanz in Markables	45	0.02326	0.00748
Parallelität gram. Rollen	3	0.00863	0.00663
PoS j	5	0.00537	0.00531
Gram. Rolle j	8	0.00095	0.00047

Tabelle 17: Anzahl unterschiedlicher Werte, Information Gain und Gain Ratio der verschiedenen Merkmale bei der Auflösung pronominaler Anaphora in TiMBL (nach Gain Ratio sortiert)

Die Distanz in Sätzen erhält im Classifier der pronominalen Koreferenz-Auflösung ein höheres Gewicht als im Classifier, der alle Formen von Anaphora auflöst. Die relativ schwache Gewichtung der grammatikalischen Parallelität ergibt sich daraus, dass nicht nur Personalpronomen, sondern auch Reflexiv- und Possessiv-Pronomen betrachtet werden. Da diese immer vom gleichen Verb abhängen wie ihr Antezedes und somit nicht die gleiche grammatikalische Rolle haben können, wird die Gewichtung der Parallelität der grammatikalischen Rollen abgeschwächt.

5.3 Der Einfluss von semantischen Merkmalen auf die Auflösung nominaler Anaphora

Wie diverse Arbeiten bestätigen (Vieira & Poesio, 2000; Soon *et al.*, 2001; Strube *et al.*, 2002; Versley, 2006, 2007a), ist die Auflösung nominaler Anaphern die schwierigste Aufgabe innerhalb der Koreferenz-Auflösung. In einer Reihe von Experimenten wird im Folgenden die Nützlichkeit diverser semantischer Merkmale für diese Aufgabe festgestellt. Weiter wird untersucht, wie semantische Eigenschaften und Beziehungen zwischen Markables als Filter benutzt werden können.

Für die Experimente werden im Trainingskorpus zuerst alle Vektoren entfernt, die pronominale Markables enthalten. Es bleiben von 98205 Trainingsvektoren 67872 (69.11%). Da die Nützlichkeit der semantischen Merkmale experimentell möglichst genau festgestellt werden soll, werden Trainingsvektoren (positive wie negative Beispiele), für die die semantischen Merkmalswerte nicht berechnet werden können (z.B. weil ein Nomen nicht in WordNet ist

oder eine NE nicht auf ein Nomen abgebildet werden konnte), aus dem Trainingskorpus entfernt. Dadurch wird eine Simplifikation vorgenommen, da die Abbildung von NEs auf Nomen aus WordNet, um semantische Merkmale zu berechnen, eine Aufgabe für jedes System zur Koreferenz-Auflösung darstellt. Mit der Absicht, die Nützlichkeit der semantischen Merkmale sinnvoll und anschaulich miteinander vergleichen zu können, wird hier diese Simplifikation vorgenommen. Durch das Löschen aller Trainingsvektoren, für die keine semantischen Merkmale berechnet werden konnten, reduziert sich das Trainingskorpus auf 41152 Trainingsvektoren (60.6% aller nominalen Paare).

Den Beobachtungen von Versley (2006) und Soon *et al.* (2001) folgend, werden aus dem verbleibenden Trainingskorpus alle Paare entfernt, die direkte nominale Anaphern enthalten und mit einfachem String Match aufgelöst werden können. Schliesslich bleiben 40656 Bridging Anaphern (59.9% aller nominalen Paare, bzw. 41.4% des gesamten Korpus) im Trainingskorpus übrig, über denen die Nützlichkeit der semantischen Merkmale evaluiert wird. Von den verbleibenden Vektoren sind nur 815 (2%) positive Instanzen. Das legt nahe, weitere Wege zu finden, um die Generierung von negativen Beispielen im Trainingskorpus zu verringern.

Für die Evaluation wird zuerst eine Baseline erstellt und das Feature Set fortlaufend um semantische Merkmale erweitert. Im Anschluss (Kapitel 5.3.5) werden die semantischen Merkmale als harte Filter verwendet. Die Verbesserungen werden in Bezug auf das CEAF-Mass diskutiert, MUC-6-Resultate werden der Vollständigkeit halber gegeben[58]. Relevante Verbesserungen, die statistisch signifikant sind[59], werden in den folgenden Tabellen herausgehoben. Dazu werden jeweils die entsprechenden p-Werte angegeben.

5.3.1 Baseline

Beim Trainieren des Baseline-Classifiers werden alle semantischen Merkmale ignoriert. Es bleibt für die jeweiligen Trainingsvektoren folgendes Feature Set.

- Distanz in Sätzen und Markables
- Grammatikalische Rollen und ob sie identisch sind
- Part-of-Speech-Information

[58] Für eine Diskussion der Vorteile des CEAF-Masses gegenüber dem MUC-6-Mass siehe Kapitel 3.3.
[59] Vgl. Kapitel 3.4

- Strings der Markables
- Determiner der Markables

Wird dieses Feature Set separat auf die ausgefilterten, direkten nominalen Anaphern angewandt, ergeben sich die Resultate in Tabelle 18. Die CEAF- und MUC-6-Resultate der Baseline für die Bridging-Koreferenzen gibt Tabelle 19 wieder.

	Recall	*Precision*	*F-Score*
CEAF	71.7%	65.7%	68.5%
MUC-6	70.9%	64.1%	63.7%

Tabelle 18: Evaluation des Baseline-Classifiers für direkte nominale Anaphern

	Recall	*Precision*	*F-Score*
CEAF	25.9%	48.2%	33.2%
MUC-6	24.4%	47.1%	32.2%

Tabelle 19: Evaluation des Baseline-Classifiers für Bridging Anaphern

Die unterschiedlichen Resultate für die direkten und die Bridging Anaphern zeigen das starke Gefälle in der Performanz der Baseline und die Notwendigkeit von zusätzlichen Ressourcen und Filtern bei der Auflösung von Bridging Anaphern auf. Für die folgenden Experimente werden die direkten Anaphern wieder vernachlässigt.

5.3.2 Semantische Relationen aus WordNet

Als erste Erweiterung der Baseline werden folgende semantische Relationen aus WordNet beim Trainieren des Classifiers als binäre Merkmale mitgegeben: Übereinstimmung der semantischen Klasse[60], Synonymie, Hyponymie, Glosse (ob ein Markable in der Glosse des anderen enthalten ist) und Meronymie. Ein weiteres binäres Merkmal hält fest, ob mindestens eine der semantische Relation vorhanden ist. Die Resultate dieses Classifiers gibt Tabelle 20.

[60] Hier sind nicht primär die Hauptklassen *Object, Person, Organization, Location* gemeint, sondern auch spezifischere Konzepte. Wenn beispielsweise „Carla Hills“ über Wikipedia auf „lawyer“ abgebildet wird und im Korpus mit „lawyer“ auf sie referenziert wird, stimmen die semantischen Klassen überein.

	Recall	*Precision*	*F-Score*
CEAF	**29.8%**	49.5%	**37.0%**
MUC-6	26.9%	44.7%	33.6%

Tabelle 20: Evaluation des Classifiers für Bridging Anaphern mit semantischen Relationen aus WordNet

Die semantischen Relationen als Merkmale führen v.a. zu einer Verbesserung des Recalls (3.9% in CEAF-Score, $p = 0.02$[61]) und zusammen mit der leichten Verbesserung der Precision zu einer Erhöhung des F-Scores.

5.3.3 Semantische Klassen

Die zweite Erweiterung der Baseline benutzt eine traditionelle Bestimmung von semantischen Hauptklassen (*Object, Person, Organization, Location*) anhand von WordNet und Listen als Merkmale (Vgl. Kapitel 4.3.2). Mit diesem Feature Set wird eine vergleichbare Verbesserung der Baseline erreicht (Tabelle 21, 4.5% Recall-Erhöhung, $p < 0.01$) wie mit demjenigen mit den semantischen Relationen aus WordNet.

	Recall	*Precision*	*F-Score*
CEAF	**30.4%**	49.8%	**37.4%**
MUC-6	28.2%	46.0%	35.0%

Tabelle 21: Evaluation des Classifiers für Bridging Anaphern mit semantischen Klassen

5.3.4 Semantische Ähnlichkeitsmasse

Als nächste Erweiterung der Baseline werden die semantischen Ähnlichkeitsmasse als Merkmale betrachtet. Die Ausgabe von WordNet::Similarity[62] ergibt zahlreiche Werte für die verschiedenen Ähnlichkeitsmasse. Wie in Kapitel 2.2.1 besprochen wurde, nützen zu viele unterschiedliche Werte für ein Merkmal bei der Klassifikation wenig. Sie führen zu hohen Information Gain- und kleinen Gain Ratio-Werten. Um das Gain Ratio der Merkmale zu verbessern, werden alle Ähnlichkeitsmasse, die Werte mit Nachkommastellen haben, so gerundet, dass sie nicht mehr als dreissig mögliche Werte haben. Dadurch wird

[61] Bei $p < 0.05$ gelten Verbesserungen als statistisch signifikant (vgl. Kapitel 3.4).
[62] s. Kapitel 2.4.4

grundsätzlich eine Verdoppelung der Gain Ratios erreicht. Trotzdem liegt die Performanz der Ähnlichkeitsmasse hinter den Verbesserungen der Baseline durch die semantischen Relationen und Klassen (Tabelle 22). Die Verbesserungen in Recall ($p = 0.06$) und F-Score ($p = 0.11$) sind ausserdem in Bezug auf das festgelegte Signifikanzniveau ($\alpha = 0.05$) statistisch gesehen nicht signifikant.

	Recall	*Precision*	*F-Score*
CEAF	29.7%	48.8%	36.5%
MUC-6	26.3%	44.6%	33.1%

Tabelle 22: Evaluation des Classifiers für Bridging Anaphern mit dem semantischen Ähnlichkeitsmassen

Aufgrund der Erkenntnis, dass das Gain Ratio eines Merkmals sich verbessert, je weniger verschiedene Werte es hat, wird das Prinzip des Rundens weitergeführt und die Ähnlichkeitsmasse in binäre Werte umgerechnet. Dazu wird für jedes Ähnlichkeitsmass einmal der Mindestwert und einmal der Durchschnitt aller Werte der im Trainingskorpus vorhandenen positiven Beispiele ermittelt. In jedem Vektor (positiv und negativ) wird dann für alle Ähnlichkeitsmasse überprüft, ob der jeweilige Merkmalswert über dem Mindest- respektive Durchschnittswert liegt und erhält als Wert 0 oder 1.

Wie die Evaluation zeigt (Tabelle 23), übertreffen beide Feature Sets die Performanz der ursprünglichen Ähnlichkeitsmasse mit mehreren Merkmalswerten. Das Feature Set mit den binären Werten, die anhand der Durchschnittswerte ermittelt wurden, schneidet am besten ab und übertrifft das ursprüngliche Ähnlichkeits-Feature Set auch bezüglich MUC-6-Score. Die Baseline wird dadurch bezüglich Recall um 5.1% ($p = 0.04$) und bezüglich F-Score um 4.8% ($p = 0.01$) verbessert. Damit erzielt dieses Feature Set das beste Resultat aller semantischen Merkmale.

Neben der weitgehend gebräuchlichen Verwendung von semantischen Merkmalen in Vektoren, soll nun im Folgenden evaluiert werden, inwiefern diese Merkmale auch als harte Filter – primär um negative Paare zu eliminieren – anwendbar sind.

5.3.5 Semantische Merkmale als harte Filter

Ein Verfahren ist Paare zu filtern, die bezüglich semantischer Klasse nicht übereinstimmen (*Person* kann z.B. nicht mit *Organization* koreferieren). Im Folgenden wird auch die Nützlich-

	Recall	*Precision*	*F-Score*
$CEAF_{binary-min}$	30.5%	48.6%	37.3%
$MUC6_{binary-min}$	26.5%	43.0%	32.8%
$CEAF_{binary-mean}$	**31.0**%	49.9%	**38.0**%
$MUC6_{binary-mean}$	28.1%	45.5%	34.7%

Tabelle 23: Evaluation des Classifiers für Bridging Anaphern mit den semantischen Ähnlichkeitsmassen als binäre Werte (ermittelt anhand der Minimalwerte ($CEAF/MUC6_{binary-min}$) oder dem Durchschnitt ($CEAF/MUC6_{binary-mean}$) der positiven Beispiele)

keit der semantischen Relationen und der Ähnlichkeitsmasse als harte Filter evaluiert. Bei der Berechnung der Evaluations-Resultate von Filterungen werden die Keys[63] des ungefilterten Goldstandards verwendet. Wenn also durch Filterung positive Instanzen in der Response gelöscht und dadurch nicht aufgelöst werden, bleiben sie im Key erhalten und fliessen in die Berechnung der Resultate mit ein.

Als erstes werden alle Instanzen im Korpus gelöscht, in denen die semantischen Klassen von Antezedens und Anapher nicht übereinstimmen. Dabei wird das Korpus auf 14900 (36.6% der Bridging Anaphern) Vektoren reduziert und enthält 594 positive Beispiele (4% des gefilterten Korpus). Dann wird der Baseline-Classifier ohne semantischen Merkmale im Feature Set über dem gefilterten Korpus trainiert und evaluiert. Die Filterung reduziert das Korpus stark, eliminiert gut die Hälfte der positiven Instanzen, verdoppelt aber ihren relativen Anteil. Dadurch wird v.a. die Precision (um 2.8%; Tabelle 24) des Baseline-Classifiers erhöht und der Rechenaufwand minimiert. Wenn die semantischen Klassen als Merkmale verwendet werden, erhöht sich dadurch v.a. der Recall des Baseline-Classifiers (s. Tabelle 21). Als Filter erhöhen die semantischen Klassen hingegen v.a. die Precision, wenn auch nicht statistisch signifikant (p = 0.06).

	Recall	*Precision*	*F-Score*
CEAF	26.5%	**51.5**%	34.6%
MUC-6	24.2%	51.0%	32.9%

Tabelle 24: Evaluation des Classifiers für Bridging Anaphern mit semantischen Klassen als Filter

[63] Die Keys entsprechen den manuell annotierten Koreferenzmengen im Goldstandard. Die Response bezeichnet die von einem System generierten Koreferenzmengen, die bei der Evaluation mit dem Key verglichen werden (s. Kapitel 3).

Bei der Filterung des Korpus anhand der semantischen Relationen aus WordNet wird das Vorhandensein einer solchen Relation als Filterkriterium verwendet. Paare, für die keine Relation bestimmt wurde, werden gelöscht. Das Korpus wird durch diesen Filter auf 1581 (3.9%) Vektoren verkleinert, wobei sich der Anteil der positiven Instanzen auf 21.7% (343) erhöht. Dadurch verbessert sich die Precision (Tabelle 25, CEAF/MUC-6 SemRel-Filt), um 13% ($p < 0.01$). Recall hingegen geht gegenüber der Baseline leicht zurück.

Wenn dem Classifier bei der Evaluation des gefilterten Korpus die semantischen Relationen zusätzlich als Merkmale mitgegeben werden, erhöht sich Recall leicht auf Kosten der Precision, und F-Score nimmt insgesamt zu (CEAF/MUC-6 SemRel-Filt+Feat).

	Recall	*Precision*	*F-Score*
CEAF SemRel-Filt	22.1%	**61.2**%	31.7%
MUC-6 SemRel-Filt	20.3%	54.9%	29.6%
CEAF SemRel-Filt+Feat	23.3%	**59.5**%	33.0%
MUC-6 SemRel-Filt+Feat	21.6%	55.6%	30.9%

Tabelle 25: Evaluation des Classifiers für Bridging Anaphern mit semantischen Relationen aus WordNet als Filter und Merkmale

Als nächstes werden die semantischen Ähnlichkeitsmasse betrachtet. Sie eignen sich insofern als Filter, als dass sie benutzt werden können, ohne dass dadurch positive Instanzen im Korpus gelöscht werden. In einem ersten Schritt werden dazu für alle Ähnlichkeitsmasse die jeweiligen Minimalwerte in den positiven Beispielen ermittelt. Dann werden alle generierten Paare, die diese Werte unterschreiten, gelöscht, wobei nur negative Instanzen eliminiert werden. Dadurch bleiben im Korpus 36891 (90.7%) Vektoren, wovon immer noch 815 (2.2%) positiv sind. Die Resultate einer solchen Filterung gibt Tabelle 26. Die Leistung des Baseline Classifers geht nach dieser Filterung des Korpus leicht zurück (CEAF/MUC-6 SemDist-Filt). Wenn dem Classifier die Ähnlichkeitsmasse zusätzlich als Merkmale mitgegeben werden (anhand der Durchschnittswerte gerundet), kann der Recall stark verbessert werden (CEAF/MUC-6 SemDist-Filt+Feat), um 6.3% in CEAF-Score ($p = 0.01$).
Dieses Feature Set mit der Filterung erreicht in der Evaluation den besten F-Score für die Bridging Anaphern. Wird das Korpus in diesem Setting zusätzlich durch die semantischen Klassen gefiltert, sinkt Recall leicht und die Precision verbessert sich nochmals deutlich (Tabelle 27).

	Recall	*Precision*	*F-Score*
CEAF SemDist-Filt	24.8%	48.5%	32.7%
MUC-6 SemDist-Filt	23.5%	47.4%	31.4%
CEAF SemDist-Filt+Feat	**31.1**%	50.3%	**38.2**%
MUC-6 SemDist-Filt+Feat	28.1%	46.2%	34.9%

Tabelle 26: Evaluation des Classifiers für Bridging Anaphern mit den semantischen Ähnlichkeitsmassen als Filter und binäre Merkmale

	Recall	*Precision*	*F-Score*
CEAF	29.2%	**55.6**%	37.8%
MUC-6	25.8%	51.3%	34.4%

Tabelle 27: Evaluation des Classifiers für Bridging Anaphern mit den semantischen Ähnlichkeitsmassen als binäre Werte (aufgrund der Durchschnittswerte); Korpus gefiltert anhand der Minimalwerte der Ähnlichkeitsmasse und semantischen Klassen

5.3.6 Diskussion der Resultate

Eine Übersicht der CEAF-Resultate gibt Tabelle 28. Darin sind relevante Verbesserungen herausgehoben. Die Evaluation der Feature Sets und Filter für die Klassifizierung von Bridging Anaphern hat gezeigt, dass semantische Merkmale als Merkmale in den Vektoren vor allem zu einer Verbesserung des Recalls führen (um bis zu 5.1% in CEAF-Score). Die Ähnlichkeitsmasse als binäre Merkmale erzielten in der Evaluation das beste Resultat bezüglich Recall, Precision und F-Score im Vergleich zu den anderen semantischen Merkmalen. Die Verwendung semantischer Merkmale als harte Filter bietet Möglichkeiten, das Korpus und damit den Rechenaufwand für die Klassifikation in unterschiedlichen Ausmassen zu verkleinern (um bis zu 96.1% mit 21.7% positiven Instanzen; Hendrickx *et al.* (2007) berichten eine Reduktion um bis zu 91.7% bei einem Anteil an positiven Instanzen von 17.7%). Dadurch ist eine starke Verbesserung der Precision beobachtbar (um bis zu 13% in CEAF-Score), wobei der Rückgang in Recall moderat bleibt (maximal 3.8%).

Direkte Vergleiche mit Systemen, die mit anderen Korpora, Evaluationsmassen, Feature Sets und Preprocessing verfahren, sind im Sinne von Mitkov (2000) kaum sinnvoll. Es soll dennoch angemerkt werden, dass z.B. Ponzetto & Strube (2006) und Versley (2007a), die ebenfalls mit semantischen Klassen, Relationen und Ähnlichkeitsmassen bzgl. Bridging Anaphora verfahren, Verbesserungen in verhältnismässig ähnlicher Grössenordnung berichten.

Feature Set	*Recall*	*Precision*	*F-Score*
Baseline	25.9%	48.2%	33.2%
+Sem. Relationen	29.8%	49.5%	37.0%
+Sem. Klassen	30.4%	49.8%	37.4%
+Sem. Ähnlichkeit	29.7%	48.8%	36.5%
+Sem. Ähnlichkeit binär (Minimalwert)	30.5%	48.6%	37.3%
+Sem. Ähnlichkeit binär (Durchschnitt)	31.0%	49.9%	38.0%
+Sem. Klasse als Filter	26.5%	51.5%	34.6%
+Sem. Relationen als Filter	22.1%	**61.2**%	31.7%
+Sem. Relationen als Filter & Merkmal	23.3%	59.5%	33.0%
+Sem. Ähnlichkeit als Filter (Minimalwert)	24.8%	48.5%	32.7%
+Sem. Ähnlichkeit als Filter (Minimalwert) & Merkmal (Durchschnitt)	**31.1**%	50.3%	**38.2**%
+Sem. Klassen & Ähnlichkeit als Filter (Minimalwert) & Ähnlichkeit als Merkmal (Durchschnitt)	29.2%	55.6%	37.8%

Tabelle 28: CEAF und MUC-6-Resultate der zehnfachen Kreuzvalidierung der verschiedenen Feature Sets und Filter im Vergleich zur Baseline

Die Verwendung der semantischen Ähnlichkeitsmasse als harte Filter wurde während der Recherche zur vorliegenden Arbeit in keinem bisherigen Ansatz entdeckt. Ein Vorteil von Ähnlichkeitsmassen gegenüber den semantischen Relationen oder Klassen als harte Filter ist, dass die Filterung nicht binär sein muss[64], sondern skaliert werden kann. In der vorangehenden Evaluation wurden zuerst die Minimalwerte aller positiven Instanzen ermittelt und alle Paare aus dem Korpus gefiltert, die diese unterschritten. Dadurch werden keine positiven Instanzen gelöscht. Bei der Filterung mit semantischen Relationen und Klassen werden positive Instanzen eliminiert, da das Ermitteln dieser Merkmale nicht fehlerfrei ist und das nicht Vorhandensein einer semantischen Relation nicht immer ein sicherer Hinweis darauf ist, dass zwei Markables nicht koreferent sind. Anzumerken bleibt, dass im Vergleich zu diesen Filtern die Ähnlichkeitsmasse mit den Minimalwerten als Filtergrenze das Korpus nicht sehr stark reduzieren (um 9.3%) und dass der Anteil der positiven Instanzen dadurch kaum (um 0.2%) erhöht wird. Werden die Durchschnittswerte genommen, reduziert sich das Korpus auf 20565 (50.6%) Vektoren, wovon 539 (2.6%) positiv sind. Gut vorstellbar ist das Ermitteln weiterer Referenzwerte (z.B. Minimal- und Maximal-Werte in negativen Beispielen, häufigste Werte, Mehrheiten von über- und unterdurchschnittlichen Werten etc.) für die Ähnlichkeitsmasse, anhand derer Instanzen aus dem Trainingskorpus gefiltert werden können.

[64] Semantische Klassen stimmen überein oder nicht; Semantische Relationen sind vorhanden oder nicht.

Die Evaluation hat gezeigt, dass Filterungen, auch wenn sie positive Instanzen löschen, den Recall meistens nicht deutlich reduzieren oder sogar leicht verbessern können. Das durch die Filterung bezüglich positiver und negativer Instanzen balanciertere Korpus führt dabei zu einer starken Verbesserung der Precision und somit des F-Scores.

5.4 Fehleranalyse

Mitkov *et al.* (2002) zeigten auf, dass durch Fehler im Preprocessing – v.a. durch das Parsen – die Leistung eines jeden Anaphernresolutions-Systems eingeschränkt wird. Einige Systeme verwenden als Eingabe die Ausgabe aus manuell revidierten Baumbanken, die ein (fast) perfektes Parsing liefern, was beim vorliegenden System nicht der Fall ist. Der folgende Abschnitt soll einige Fehlerquellen aufzeigen, die die Leistung des Classifiers beeinträchtigen. Die Fehlklassifikationen des Classifiers ergeben sich aus Majoritätsentscheiden anhand der Nearest Neighbors aus dem Trainingskorpus. Wesentlich sind – nebst der ungleichen Anzahl positiver und negativer Beispiele (s. Kapitel 2.3) – also die Fehler, die im Preprocessing entstehen und zu einer fehlerhaften Repräsentation im Trainingskorpus führen.

5.4.1 Segmentierungsfehler

Ein Problem ist die Segmentierung der Texte in Sätze, Chunks und Markables. Wie in Kapitel 4.1 aufgezeigt, treten in TTT2 bei der Satzsegmentierung Probleme durch die Interpunktion auf (Interpunktionen in Abkürzungen wie „Calif.“ werden als Satzgrenze interpretiert). Dadurch werden 195 Artikel (32.66%) des Goldstandards unbrauchbar. Da diese Artikel ausgeschlossen werden, entsteht dadurch keine Beeinträchtigung des Classifiers.

Segmentierungsfehler treten auch bei Named Entities auf. Gewisse NEs enthalten Koordinationen, die zur NE gehören. Durch die Heuristik bei der Identifikation der Markables werden sie aber als Segmentierungsgrenze erkannt: „St. Michael and All Angels“ ist der Name einer texanischen Kirche. Durch die Segmentierungsheuristik wird die NE in „St. Michael“ und „All Angels“ unterteilt. Der Wikipedia-Algorithmus bildet „St. Michael“ eigentlich korrekt auf „archangel“ (als belebte, männliche Person) und „All Angels“ auf eine britische „pop group“ ab, anstatt auf eine Kirche. Das gleiche Problem entsteht bei Mehrwort-Ausdrücken, die aus Nomen bestehen und koordiniert sind, wie „Sunday morning and evening services“. Der Ausdruck wird in „Sunday morning“ und „evening services“ zerlegt.

Ein weiteres Segmentierungsproblem besteht bei Verben mit indirekten Objekten, wenn Objekt und indirektes Objekt NEs sind und nicht durch einen Artikel getrennt werden. In Konstruktionen wie „Peter gave Mary Rowdy.“ wird „Mary Rowdy“ als ein Chunk identifiziert.

Auch werden aufgrund der statistischen Ressourcen des PoS-Taggers strukturell identische NE-Koordinationen je nach den auftretenden NEs unterschiedlich segmentiert. In „Peter and I go to his house.“ werden „Peter“, „and“ und „I“ als einzelne Chunks segmentiert. „I“ wird als Subjekt von „go“ bestimmt und „Peter“ erhält keine Verb-Dependenz. Im Gegensatz dazu wird in „Mark and Peter go to his house.“ der Chunk „Mark and Peter“ als solcher segmentiert und vom Preprocessing nicht weiter zerlegt. „Mark“ wird für „go“ über den Kopf des Chunks („Peter“) als Subjekt ermittelt.

5.4.2 Fehler im PoS-Tagging

Das PoS-Tagging produziert in gewissen Konstellationen folgende Fehler. In Konstruktionen wie „I do matter.“ oder „I do like math.“ klassifiziert der CC-Tagger „matter“ und „math“ als Nomen und „like“ als Präposition. Dadurch wird beispielsweise „math“ ein Präpositional-Objekt von „do“ und „matter“ als nominales Mention in die Weiterverarbeitung übergeben. Auch entstehen Fehler wenn Possessivpronomen an Satzenden stehen. In „I saw a friend on mine.“ wird „mine“ als Nomen und PP-Attachment von „friend“ interpretiert.

Unter den Preprocessing-Fehlern spielen v.a. die Segmentierungsambiguitäten bei koordinierten NPs eine zentrale Rolle. Wie in Kapitel 4.3.1 beschrieben, gestaltet sich ihre Verarbeitung schwierig und wird durch Fehler im Preprocessing kompliziert. Gerade die Bestimmung des Verbs, von dem die Markables solcher NPs anhängen, wird dadurch oft verunmöglicht. Angesichts dessen, dass die Verb-Dependenz als harter Filter bei der Paargenerierung verwendet wird, ist dieser Punkt wesentlich.

5.4.3 Fehler in der Bestimmung der Verb-Dependenzen

Die Bestimmung der Verben, von denen die Markables abhängen, erfolgt über den Dependenz-Prädikaten der Pro3Gres-Ausgabe. Folgende Dependenz-Relationen werden dabei berücksichtigt: Subjekt, Objekt, Präpositional-Objekt, indirektes Objekt, Apposition, modale Beziehungen und Bridging (in dieser Reihenfolge). Da die Markables, Verben und Argumente in den jeweiligen Prädikaten nicht immer an der selben Argument-Position der Prolog-Prädikate

stehen, kann es vorkommen, dass anstatt der Verben die Verb-Argumente extrahiert werden (da die PoS-Information in den Dependenz-Prädikaten nicht enthalten ist, kann sie nicht als Indikator verwendet werden). Auch können vom Parser nicht für alle Sätze die entsprechenden Dependenzen ermittelt werden. Gerade in Sätzen, in denen zwischen dem Subjekt und dem Verb lange Einschübe – z.B. in Form von Appositionen mit Aufzählungen – vorhanden sind, schlägt dies fehl.

> „Those countries – including Japan , Italy , Canada , Greece and Spain – are still of some concern to the U.S. but are deemed to pose less-serious problems for American patent and copyright owners than those on the "priority" list.“

Verb-Dependenzen nach Pro3Gres:

```
modpp('concern#8', 'u.s.#10', 'to#9', '(->)', 13).
pobj('be#5', 'concern#8', 'of#7', '(->)', 13).
iobj2('be#5', 'concern#8', 'pp:iobj', 13).
obj('pose#13', 'problem#14', 'owner#16', '(->)', 13).
sentobj('deem#12', 'pose#13', 'problem#14', '(->)', 13).
bridge('be#5', 'deem#12', 'but#11', '(->)', 13).
```

Die Subjekt-Relation zwischen „countries“ und „are“ wird nicht erkannt. Bei den insgesamt 41679 Markables schlägt die Bestimmung der Verb-Dependenz bei 3035 (7.3%) fehl. Der entsprechende Filter, der die Verb-Dependenzen überprüft und allenfalls die Vektor-Generierung verhindert, lässt Paare passieren, bei denen ein oder beide Markables ohne Verb-Dependenz sind. Dadurch entstehen im Trainingskorpus 15426 (15.7%) Vektoren, bei denen der Filter nicht angewendet werden kann.

5.4.4 Inkonsistente Annotationen im Goldstandard

OntoNotes (s. Kapitel 4.1) strebt ein Inter Annotator Agreement von 90% an. Dennoch entstehen in der Annotation Fehler. Beispielsweise fehlen True Mentions in Koreferenzmengen.

> „The oldest bell-ringing group in the **country**, the Ancient Society of College Youths, founded in 1637, remains male-only, a fact that’s particularly galling to women because the group is the sole source of ringers for **Britain**’s most prestigious churches, St. Paul’s Cathedral and Westminster Abbey.”

Im Goldstandard fehlt „country“ in der Koreferenzmenge von „Britain“. Ein weiteres Problem ergibt sich aus der Unterteilung der Koreferenz-Relationen in IDENT- und APPOS-Relationen (Äquivalenz und Apposition) in OntoNotes und der konzeptionellen Entscheidung, Appositions-Relationen aus dem Goldstandard im vorliegenden System nicht ins Trainingskorpus aufzunehmen (vgl. Kapitel 4.1). Anaphern, die in Appositionen auftreten, werden in der OntoNotes-Annotation nicht zusätzlich in die IDENT-Klasse des Antezedens aufgenommen.

```
CHAIN: IDENT@72@wsj/00/wsj_0089@wsj@en@on
LINKS:
 IDENT@38:4:15@IDENT@72@wsj/00/wsj_0089@wsj@en@on: the Rev. Jeremy Hummerstone ,
 vicar of Great Torrington , Devon
 IDENT@38:29:29@IDENT@72@wsj/00/wsj_0089@wsj@en@on: he
 IDENT@39:19:21@IDENT@72@wsj/00/wsj_0089@wsj@en@on: the Vicar Hummerstone

CHAIN: APPOS@145@wsj/00/wsj_0089@wsj@en@on
LINKS:
 HEAD@38:4:7@APPOS@145@wsj/00/wsj_0089@wsj@en@on: the Rev. Jeremy Hummerstone
 ATTRIBUTE@38:9:14@APPOS@145@wsj/00/wsj_0089@wsj@en@on: vicar of Great Torrington , Devon
```

Das Appositions-Attribut „vicar of Great Torrington, Devon“ wird nicht separat in die Koreferenzmenge (IDENT@72) des Antezedens „the Rev. Jeremy Hummerstone“ aufgenommen, sondern der Instanz der IDENT-Relation angefügt. Findet das System dennoch solche Koreferenz-Relationen, werden sie als False Positives interpretiert.

6 Schlussbemerkungen

In der vorliegenden Arbeit wurde die Entwicklung von Machine Learning-basierter Systeme zur Koreferenz- und Anaphernresolution aufgezeigt. Dabei wurden relevante Merkmale und Filtertechniken beleuchtet. Ein ML-System zur Koreferenz-Auflösung wurde implementiert und v.a. die Performanz bezüglich der Auflösung von Bridging Anaphern ausführlich evaluiert. Die Recherche in der Forschungsliteratur und die in der vorliegenden Arbeit durchgeführten Experimente haben ergeben, dass semantische Merkmale die Verarbeitung nominaler Anaphora – wie hypothetisch angenommen – verbessern. In den Experimenten hat sich die Verwendung semantischer Merkmale als harte Filter als zusätzliche Verbesserung erwiesen. Durch Filterung wird nicht nur die Leistung von Classifiern gesteigert, sondern auch die Rechenzeit verringert, die für die Koreferenz-Auflösung benötigt wird, da weniger Instanzen gelernt und klassifiziert werden müssen.

Die Experimente mit den Ähnlichkeitsmassen in der vorliegender Arbeit haben gezeigt, dass auch sie sich als Filter eignen. Als einer ihrer Vorteile gegenüber den semantischen Relationen oder Klassen als Filter wurde ihre Skalierbarkeit hervorgehoben. Mit verschiedenen Referenzwerten lässt sich ein Trainingskorpus graduell und unterschiedlich filtern. Durch die Kombination der Verwendung der Ähnlichkeitsmasse als Merkmale und Filter wurden die besten Resultate in der Evaluation des implementierten Systems zur Auflösung von Bridging Anaphoa erreicht. Die Verwendung der Ähnlichkeitsmasse als Filter stellt ein Novum dar.

Die Untersuchung des verwendeten Korpus zeigte, dass der grösste Teil der möglichen koreferenten Einheiten Eigennamen und Nomen sind. Der Vergleich der Classifier für direkte und indirekte nominale Anaphora im implementierten System wies auf, dass die Leistung des Letzteren deutlich hinter derjenigen des Ersten zurückliegt (35% in CEAF-Score), auch noch nach der bestmöglichen Verbesserung durch semantische Filter und Merkmale (30% Rückstand in CEAF-Score). Da die indirekten nominalen Anaphern einen bedeutenden Teil (60% im verwendeten Korpus) aller nominaler Anaphern ausmachen, liegt in der Verbesserung ihrer Auflösung immer noch ein grosses Potenzial zur allgemeinen Verbesserung der Leistung von Koreferenz-Auflösungssystemen.

Auch hat die Evaluation gezeigt hat, dass Merkmale bei der Auflösung von verschiedenen Arten von Anaphora unterschiedliche Gewichtungen erhalten. Deutlich erkennbar wurde dies bei der Analyse der Merkmalsgewichtung des pronominalen Classifiers im Vergleich

mit derjenigen des Classifiers, der alle Arten von Anaphora auflöst. Die unterschiedliche Gewichtung legt Nahe, verschiedene Feature Sets und Classifier für verschiedene Arten von Anaphora zu erstellen und zu trainieren. Geschieht dies nicht, müssen gewisse Merkmale in gewissen Konstellationen mit Platzhalter-Werten belegt werden, da sie nicht berechnet werden können oder sollen (z.B. Levenshtein-Distanz zwischen Nomen und Pronomen). TiMBL beispielsweise kann nicht signalisiert werden, dass ein solcher Platzhalter-Wert vorhanden ist, der ohne Aussagekraft bleibt. Platzhalter werden also als konkrete Merkmalswerte interpretiert und aus ihnen gelernt, wodurch die eigentliche Aussagekraft dieser Merkmale verfälscht wird.

Eine Demoversion des implementierten Systems, das pronominale und nominale Koreferenz berechnet, kann im Internet[65] benutzt werden. Die Auswahl der semantischen Merkmale im Feature Set wurden in der Demoversion im Hinblick auf sinnvolle Berechnungszeiten auf die semantischen Hauptkategorien beschränkt.

[65] http://www.cl.uzh.ch/kitt/coref/ (Stand 5.8.2009)

7 Literaturverzeichnis

Literatur

Ailloud, Etienne, & Klenner, Manfred. 2009. Towards More Linguistically Constrained Coreference Resolution. *In: TALN (Traitement Automatique des Langues Naturelles) (to appear)*.

Aone, Chinatsu, & Bennett, Scott. 1996. Applying Machine Learning to Anaphora Resolution. *S. 302–314 in: Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*. London, UK: Springer-Verlag.

Baayen, Harald R., Piepenbrock, Richard, & Gulikers, Leon. 1995. *The CELEX Lexical Database. Release 2 (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania.

Bagga, Amit, & Baldwin, Breck. 1998. Algorithms for Scoring Coreference Chains. *S. 563–566 in: The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*.

Baldwin, Breck. 1997. CogNIAC: High Precision Coreference with Limited Knowledge and Linguistic Resources. *S. 38–45 in: Proceedings of the ACL'97/EACL'97 Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*.

Banerjee, Satanjeev, & Pedersen, Ted. 2002. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. *S. 136–145 in: CICLing '02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*. London, UK: Springer-Verlag.

Bikel, Daniel M., Schwartz, Richard, & Weischedel, Ralph M. 1999. An Algorithm that Learns What's in a Name. *Machine Learning*, **34**(1), 211–231.

Bontcheva, Kalina, Dimitrov, Marin, Maynard, Diana, Tablan, Valentin, & Cunningham, Hamish. 2002 (June). Shallow Methods for Named Entity Coreference Resolution. *In: TALN 2002*.

Bunescu, Razvan, & Pasca, Marius. 2006. Using Encyclopedic Knowledge for Named Entity Disambiguation. *S. 9–16 in: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06), Trento, Italy*.

Cardie, Claire. 1996. Automating Feature Set Selection for Case-Based Learning of Linguistic Knowledge. *S. 113–126 in: Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing, EMNLP*.

Cardie, Claire, & Wagsta, Kiri. 1999. Noun Phrase Coreference as Clustering. *S. 82–89 in: Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Chomsky, Noam. 1956. Three Models for the Description of Language. *Information Theory, IRE Transactions on Information Theory IT*, **2**(3), 113–124.

Cover, Thomas, & Hart, Peter E. 1967. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, **13**(1), 21–27.

Curran, James R., & Clark, Stephen. 2003. Investigating GIS and Smoothing for Maximum Entropy Taggers. *S. 91–98 in: EACL '03: Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Daelemans, Walter, Zavrel, Jakub, van der Sloot, Ko, & van den Bosch, Antal. 2007. *TiMBL: Tilburg Memory-Based Learner*. Tech. Rept. Induction of Linguistic Knowledge, Tilburg University and CNTS Research Group, University of Antwerp.

De Meulder, Fien, & Daelemans, Walter. 2003. Memory-based Named Entity Recognition using Unannotated Data. *S. 208–211 in: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*. Morristown, NJ, USA: Association for Computational Linguistics.

Denis, Pascal, & Baldridge, Jason. 2009. Global Joint Models for Coreference Resolution and Named Entity Classification. *S. 87–96 in: Procesamiento del Lenguaje Natural 42*.

Evans, Richard, & Orăsan, Constantin. 2000 (16 – 18 November). Improving Anaphora Resolution by Identifying Animate Entities in Texts. *S. 154 – 162 in: Proceedings of the Discourse Anaphora and Reference Resolution Conference (DAARC2000)*.

Fellbaum, Christiane. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

Finkel, Jenny Rose, & Manning, Christopher D. 2008. Enforcing Transitivity in Coreference Resolution. *S. 45–48 in: HLT '08: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*. Morristown, NJ, USA: Association for Computational Linguistics.

Francis, W. Nelson, & Kucera, Henry. 1979. *Brown Corpus Manual*. Department of Linguistics, Brown University.

Garera, Nikesh, & Yarowsky, David. 2006. Resolving and Generating Definite Anaphora by Modeling Hypernymy Using Unlabeled Corpora. *S. 37–44 in: CoNLL-X '06: Proceedings of the Tenth Conference on Computational Natural Language Learning*. Morristown, NJ, USA: Association for Computational Linguistics.

Gasperin, Caroline, & Vieira, Renata. 2004. Using Word Similarity Lists for Resolving Indirect Anaphora. *S. 40–46 in:* Harabagiu, Sanda, & Farwell, David (Hrsg.), *ACL 2004: Workshop on Reference Resolution and its Applications*. Barcelona, Spain: Association for Computational Linguistics.

Gosset, William Sealy. 1908. The Probable Error of a Mean. *Biometrika*, **6**(1), 1–25.

Grover, Claire. 2008 (July). *LT-TTT2 Example Pipelines Documentation*. Tech. Rept. Edinburgh Language Technology Group, University of Edinburgh.

Hamp, Birgit, & Feldweg, Helmut. 1997. GermaNet – a Lexical-Semantic Net for German. *S. 9–15 in: In Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*.

Harabagiu, Sanda M., Bunescu, Razvan C., & Maiorano, Steven J. 2001. Text and Knowledge Mining for Coreference Resolution. *S. 1–8 in: NAACL '01: Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001*. Morristown, NJ, USA: Association for Computational Linguistics.

Hendrickx, Iris, Hoste, Veronique, & Daelemans, Walter. 2007. Evaluating Hybrid versus Data-driven Coreference Resolution. *S. 137–150 in: Anaphora: Analysis, Algorithms and Applications (LNAI)*. London, UK: Springer-Verlag.

Hirst, Graeme, & St-Onge, David. 1997. *WordNet: An Electronic Lexical Database*. The MIT Press. Kap. Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms, s. 305–332.

Hobbs, Jerry R. 1976. *Pronoun Resolution*. Tech. Rept. 76-1. Research Report, Department of Computer Sciences, City College, City University of New York.

Hoste, Veronique. 2005. *Optimization Issues in Machine Learning of Coreference Resolution*. Ph.D. Thesis, Antwerp University.

Hovy, Eduard, Marcus, Mitchell, Palmer, Martha, Ramshaw, Lance, & Weischedel, Ralph. 2006. OntoNotes: The 90% Solution. *S. 57–60 in: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. New York City, USA: Association for Computational Linguistics.

Jiang, Jay J., & Conrath, David W. 1997. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. *S. 9008–9023 in: International Conference Research on Computational Linguistics (ROCLING X)*.

Kasami, Tadao. 1965. *An Efficient Recognition and Syntax-analysis Algorithm for Context-free Languages*. Tech. Rept. Air Force Cambridge Research Lab.

Kazama, Jun'ichi, & Torisawa, Kentaro. 2007. Exploiting Wikipedia as External Knowledge for Named Entity Recognition. *S. 698–707 in: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Klenner, Manfred. 2007. Enforcing Consistency on Coreference Sets. *S. 323–328 in: In Recent Advances in Natural Language Processing (RANLP)*.

Lappin, Shalom, & Leass, Herbert J. 1994. An Algorithm for Pronominal Anaphora Resolution. *Computational Linguistics*, **20**, 535–561.

Leacock, Claudia, & Chodorow, Martin. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press. Kap. Combining Local Context and WordNet Similarity for Word Sense Identification, s. 265–283.

Lesk, Michael. 1986. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. *S. 24–26 in: SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation*. New York, NY, USA: ACM.

Levenshtein, Vladimir I. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, **10**(8), 707–710.

Lin, Dekang. 1998. An Information-Theoretic Definition of Similarity. *S. 296–304 in: ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Linke, Angelika, Nussbaumer, Markus, & Portmann, Paul R. 2001. *Studienbuch Linguistik.* 4 Aufl. Tübingen: Niemeyer.

Luo, Xiaoqiang. 2005. On Coreference Resolution Performance Metrics. *S. 25–32 in: HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Morristown, NJ, USA: Association for Computational Linguistics.

Magnini, Bernardo, Negri, Matteo, Prevete, Roberto, & Tanev, Hristo. 2002. A WordNet-based Approach to Named Entities Recognition. *S. 1–7 in: COLING-02 on SEMANET*. Morristown, NJ, USA: Association for Computational Linguistics.

Marcus, Mitchell P., Santorini, Beatrice, Marcinkiewicz, Mary Ann, & Taylor, Ann. 1999. *Treebank-3*. Tech. Rept. Linguistic Data Consortium, Philadelphia.

Markert, Katja, & Nissim, Malvina. 2005. Comparing Knowledge Sources for Nominal Anaphora Resolution. *Computational Linguistics*, **31**(3), 367–402.

McCarthy, Joseph F., & Lehnert, Wendy G. 1995. Using Decision Trees for Coreference Resolution. *S. 1050–1055 in: Proceedings of the Fourteenth International Conference on Artificial Intelligence*.

Miller, George A., & Charles, Walter G. 1991. Contextual Correlates of Semantic Similarity. *Language and Cognitive Processes*, **6**(1), 1–28.

Minnen, Guido, Carroll, John, & Pearce, David. 2001. Applied Morphological Processing of English. *Journal for Natural Language Engineering*, **7**(3), 207–223.

Mitchell, Alexis, Strassel, Stephanie, Przybocki, Mark, Davis, JK, Doddington, George, Grishman, Ralph, Meyers, Adam, Brunstein, Ada, Ferro, Lisa, & Sundheim, Beth. 2003. *ACE-2 Version 1.0*. Tech. Rept. Linguistic Data Consortium, Philadelphia.

Mitkov, Ruslan. 1997. Factors in Anaphora Resolution: They are not the only things that matter. A case study based on two different approaches. *S. 14–21 in: Proceedings of the ACL'97/EACL'97 Workshop on Operational Factors in Practical, Robust Anaphora Resolution*. Morristown, NJ, USA: Association for Computational Linguistics.

Mitkov, Ruslan. 1998. Robust Pronoun Resolution with Limited Knowledge. *S. 869–875 in: ACL-36: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Mitkov, Ruslan. 2000. Towards a more Consistent and Comprehensive Evaluation of Anaphora Resolution Algorithms and Systems. *In: Proceedings of the Discourse, Anaphora and Reference Resolution Conference (DAARC2000)*.

Mitkov, Ruslan, Evans, Richard, & Orasan, Constantin. 2002. A New, Fully Automatic Version of Mitkov's Knowledge-Poor Pronoun Resolution Method. *S. 168–186 in: CICLing '02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*. London, UK: Springer-Verlag.

Ng, Vincent. 2004. Learning Noun Phrase Anaphoricity to Improve Coreference Resolution: Issues in Representation and Optimization. *In: ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Ng, Vincent. 2007. Semantic Class Induction and Coreference Resolution. *S. 536–543 in: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics.

Ng, Vincent, & Cardie, Claire. 2002a. Combining Sample Selection and Error-driven Pruning for Machine Learning of Coreference Rules. *S. 55–62 in: EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*. Morristown, NJ, USA: Association for Computational Linguistics.

Ng, Vincent, & Cardie, Claire. 2002b. Identifying Anaphoric and Non-anaphoric Noun Phrases to Improve Coreference Resolution. *S. 1–7 in: Proceedings of the 19th International Conference on Computational linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Ng, Vincent, & Cardie, Claire. 2002c. Improving Machine Learning Approaches to Coreference Resolution. *S. 104–111 in: ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Patwardhan, Siddharth. 2003 (August). *Incorporating Dictionary and Corpus Information into a Context Vector Measure of Semantic Relatedness*. M.Phil. Thesis, University of Minnesota.

Pedersen, Ted, Patwardhan, Siddharth, & Michelizzi, Jason. 2004. WordNet::Similarity: measuring the relatedness of concepts. *S. 38–41 in: HLT-NAACL '04: Demonstration Papers at HLT-NAACL 2004 on XX*. Morristown, NJ, USA: Association for Computational Linguistics.

Phan, Hieu X., Nguye, Minh L., Horiguchi, S., Ho, Bao T., & Inoguchi, Y. 2005. Classification with Maximum Entropy Modeling of Predictive Association Rules. *S. 682–689 in: Machine Learning: ECML 2005*. Heidelberg: Springer Berlin.

Poesio, Massimo. 2004. Discourse Annotation and Semantic Annotation in the GNOME Corpus. *S. 72–79 in: Proceedings of the ACL Workshop on Discourse Annotation*. Barcelona: Association for Computational Linguistics.

Poesio, Massimo, Uryupina, Olga, Vieira, Renata, Alexandrov-Kabadjov, Mijail, & Goulart, Rodrigo. 2004. Discourse-New Detectors for Definite Description Resolution: A Survey and a Preliminary Proposal. *S. 47–54 in: ACL 2004: Workshop on Reference Resolution and its Applications*. Barcelona: Association for Computational Linguistics.

Ponzetto, Simone P., & Strube, Michael. 2006. Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution. *S. 192–199 in: Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. Association for Computational Linguistics Morristown, NJ, USA.

Pradhan, Samer, Ward, Wayne, Hacioglu, Kadri, Martin, James H., & Jurafsky, Dan. 2004 (May). Shallow Semantic parsing using support vector machines. *In: Proceedings of the Human Language Technology Conference/North American chapter of the Association for Computational Linguistic annual meeting*. Association for Computational Linguistics, Boston, MA.

Prince, Ellen F. 1992. The ZPG Letter: Subjects, Definiteness, and Information-status. *S. 295–325 in: Discourse Description: Diverse Analyses of a Fund-raising Text*. Philadelphia: John Benjamins.

Quinlan, J. Ross. 1993. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Resnik, Philip. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. *S. 448–453 in: Proceedings of the 14th International Joint Conference on Artificial Intelligence*.

Roth, Dan, & Yih, Wen-Tau. 2004. A Linear Programming Formulation for Global Inference in Natural Language Tasks. *S. 1–8 in: Proceedings of the 2004 Conference on Computational Natural Language Learning (CoNLL-2004)*.

Schneider, Gerold. 2007. *Hybrid Long-distance Functional Dependency Parsing*. Ph.D. Thesis, University of Zurich.

Schütze, Hinrich. 1998. Automatic Word Sense Discrimination. *Computational Linguistics*, **24**(1), 97–123.

Soon, Wee M., Ng, Hwee T., & Daniel. 2001. A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, **27**(4), 521–544.

Strube, Michael, & Ponzetto, Simone Paolo. 2006. WikiRelate! Computing Semantic Relatedness Using Wikipedia. *S. 1419–1424 in: AAAI'06: proceedings of the 21st national conference on Artificial intelligence*. AAAI Press.

Strube, Michael, Rapp, Stefan, & Müller, Christoph. 2002. The Influence of Minimum Edit Distance on Reference Resolution. *S. 312–319 in: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, 6-7 July 2002*, vol. 10. Philadelphia, USA: Association for Computational Linguistics.

Sundheim, Beth (Hrsg.). 1993. *Proceedings of the 5th Conference on Message Understanding*. Baltimore, Maryland, USA: Morgan Kaufmann Publishers Inc.

Tapanainen, Pasi, & Järvinen, Timo. 1997. A Non-projective Dependency Parser. *S. 64–71 in: Proceedings of the Fifth Conference on Applied Natural Language Processing*. Morristown, NJ, USA: Association for Computational Linguistics.

Telljohann, Heike, Hinrichs, Erhard, & Kübler, Sandra. 2004. The TüBa-D/Z Treebank - Annotating German with a Context-Free Backbone. *S. 2229–2235 in: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*.

Tjong Kim Sang, Erik F., & De Meulder, Fien. 2003. Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition. *S. 142–147 in: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*. Morristown, NJ, USA: Association for Computational Linguistics.

Uryupina, Olga. 2003. High-precision Identification of Discourse New and Unique Noun Phrases. *S. 80–86 in: ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Versley, Yannick. 2006. A Constraint-based Approach to Noun Phrase Coreference Resolution in German Newspaper Text. *In: Konferenz zur Verarbeitung Natürlicher Sprache (KONVENS '06)*.

Versley, Yannick. 2007a. Antecedent Selection Techniques for High-Recall Coreference Resolution. *S. 496–505 in: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics.

Versley, Yannick. 2007b. Using the Web to Resolve Coreferent Bridging in German Newspaper Text. *S. 253—-261 in: Proceedings of GLDV-Frühjahrstagung 2007*. Tübingen: Gunter Narr Verlag.

Vieira, Renata, & Poesio, Massimo. 2000. An Empirically Based System for Processing Definite Descriptions. *Comput. Linguist.*, **26**(4), 539–593.

Vilain, Marc, Burger, John, Aberdeen, John, Connolly, Dennis, & Hirschman, Lynette. 1995. A Model-theoretic Coreference Scoring Scheme. *S. 45–52 in: MUC6 '95: Proceedings of the 6th conference on Message understanding*. Morristown, NJ, USA: Association for Computational Linguistics.

Weischedel, Ralph, & Brunstein, Ada. 2005. *BBN Pronoun Coreference and Entity Type Corpus*. Tech. Rept. Linguistic Data Consortium, Philadelphia.

Wu, Zhibiao, & Palmer, Martha. 1994. Verb Semantics And Lexical Selection. *S. 133–138 in: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*. Las Cruces, New Mexico: Association for Computational Linguistics.

Ševciková, Renata. 2005. *Schwierigkeiten in der automatischen Verarbeitung von Anaphora*. M.Phil. Thesis, Institut für Computerlinguistik, Universität Zürich.

A Lebenslauf

Persönliche Angaben

Don Basil Tuggener
Rötelstr. 19
8006 Zürich

Email: don.tuggener@gmail.com

Geboren am 7.9.1979
Heimatort: Zürich

Ausbildung

1996-2000	Kantonsschule Stadelhofen, Typus L
2001-2003	Ausbildung in digitaler Klangsynthese (Csound) an der Hochschule für Musik und Theater Zürich
2003-2010	Studium der Deutschen Sprach- und Literaturwissenschaft, Computerlinguistik und Sozial- und Wirtschaftsgeschichte an der Universität Zürich

Berufliche und nebenberufliche Tätigkeiten

2000-2001	Redaktionsmitglied beim E-Zine ZDNet.ch
2001-2003	Webpublisher bei Credit-Suisse
seit 2004	Freiberuflicher Übersetzer und Filmuntertitler
seit 2005	Vorstandsmitglied des Fachvereins für Computerlinguistik-Studierende ClinZ/CH

Freizeit

Alpinismus
Skateboarding
Wellenreiten
Reisen