Sandra Kübler, Wolfgang Maier

Über den Einfluss von Part-of-Speech-Tags auf Parsing-Ergebnisse

1 Einleitung

Lange Zeit konzentrierte sich die Forschung im datengetriebenen statistischen Konstituenzparsing auf die Entwicklung von Parsingmodellen für das Englische, genauer gesagt, für die Penn Treebank (Marcus et al., 1993). Einer der Gründe dafür, warum sich solche Modelle nicht ohne Weiteres auf andere Sprachen generalisieren lassen, ist die eher schwach ausgeprägte Morphologie des Englischen: Probleme, die sich bei Parsen einer morphologisch reichen Sprache wie z.B. Arabisch oder Deutsch stellen, stellen sich für das Englische nicht. Vor allem in den letzten Jahren erfuhr die Forschung zu Parsingproblemen, die sich auf komplexe Morphologie beziehen, ein gesteigertes Interesse (Kübler und Penn, 2008; Seddah et al., 2010, 2011; Apidianaki et al., 2012).

In einer Baumbank sind Wörter im allgemeinen Information annotiert, die Auskunft über die Wortart (Part-of-Speech, POS) und morphologischen Eigenschaften eines Wortes gibt. Wo, sofern vorhanden, die Trennlinie zwischen Wortart und morphologischer Information gezogen wird und wie detailliert annotiert wird, hängt von der Einzelsprache und dem Annotationsschema ab. In einigen Baumbanken gibt es keine separate morphologische Annotation (wie z.B. in der Penn Treebank), in anderen sind Part-of-Speech- und Morphologie-Tagsets getrennt (z.B. in den deutschen Baumbanken TiGer (Brants et al., 2002) und NeGra (Skut et al., 1997)), und in anderen ist wiederum nur ein Tagset vorhanden, das sowohl POS- als auch Morphologie-Information enthält (z.B. in der Szeged Treebank (Csendes et al., 2005)). Die Anzahl verschiedener Tags für Sprachen mit einer komplexen Morphologie kann in die Tausende gehen, so z.B. für Tschechisch (Hajič et al., 2000), während für die Modellierung der Wortarten von Sprachen mit wenig bis keiner Morphologie nur wenige Tags ausreichen, z.B. 33 Tags für die Penn Chinese Treebank (Xia, 2000). Wir schließen der Einfachheit halber alle Annotationstypen ein, wenn wir ab hier von *Part-of-Speech-Annotation* sprechen.

Die Part-of-Speech-Tags nehmen eine Schlüsselrolle beim Parsen ein als Schnittstelle zwischen lexikalischer Ebene und dem eigentlichen Syntax-Baum: Während des Parsingvorgangs wird der eigentliche Konstituenzbaum nicht direkt über den Wörtern, sondern über der Part-of-Speech-Annotation erstellt. Ein Part-of-Speech-Tag kann als eine Äquivalenzklasse von Wörtern mit ähnlichen distributionellen Charakteristika angesehen werden, die über die individuellen Wörter abstrahiert und damit die Anzahl der Parameter beschränkt, für die Wahrscheinlichkeiten gelernt werden müssen. Die eigentlichen Wörter finden bei lexikalisierten Parsern Eingang in das Wahrscheinlichkeitsmodell. Es ist offensichtlich, dass die Part-of-Speech-Annotation direkten Einfluss auf die Qualität des Parsebaums hat. Nicht nur die Qualität des Taggers spielt hierbei eine Rolle, sondern auch die Granularität des Tagsets an sich. Es muss ein Kompromiss

gefunden werden zwischen zu hoher Abstraktion, die wichtige Unterscheidungen unterdrückt, und zu hoher Detailgenauigkeit, die durch die Trainingsdaten nicht unbedingt abgedeckt ist.

In den letzten Jahren sind daher einige Arbeiten entstanden, die zum Ziel haben, sprachübergreifend die lexikalische Ebene zu einer möglichst sicheren Basis für die Erstellung eines Parsebaums zu machen. Diese reichen von der Einführung des Universal Tagsets (UTS) (Petrov et al., 2012), einem reduzierten, sprachübergreifenden Tagset, über Arbeiten zu lexikalischem Clustering, siehe z.B. Koo et al. (2008), bis zum Entwurf von faktorisierten Parsingmodellen, in denen versucht wird, Parsing und lexikalische Annotation separat voneinander zu modellieren, wie z.B. Chen und Kit (2011).

Wenig untersucht ist bis jetzt der Zusammenhang zwischen der Granularität des POS-Tagsets, verschiedenen Taggern und Parsingergebnissen innerhalb eines „Pipeline"-Ansatzes, in dem ein Parser die Ausgabe eines Taggers als Eingabe erhält. Im vorliegenden Artikel untersuchen wir diese Fragestellung anhand von verschiedenen Variationen des Stuttgart-Tübingen-Tagsets (STTS) (Schiller et al., 1995; Thielen und Schiller, 1994), das für die Annotation der beiden großen deutschen Baumbanken TiGer und TüBa-D/Z (Telljohann et al., 2012) verwendet wurde, und des Universal Tagsets (Petrov et al., 2012). Des Weiteren verwenden wir verschiedene POS-Tagger, namentlich TnT (Brants, 2000), SVMTool (Giménez und Màrquez, 2004), Mallet-CRF (McCallum, 2002) und Stanford-MaxEnt (Toutanova und Manning, 2000), sowie den Berkeley Parser (Petrov et al., 2006). In zwei Experimentgruppen betrachten wir den Einfluss der Granularität von STTS auf die Qualität von Tagger- und Parserausgaben. In zwei weiteren Experimentgruppen untersuchen wir verschiedene Untermengen von morphologischen Merkmalen des STTS, sowie den Einfluss der morphologischen Merkmalssets auf die Parsingqualität.

Der Artikel ist wie folgt strukturiert. Im folgenden Abschnitt stellen wir einen Ausschnitt der vorhandenen Literatur über POS-Tagging und Parsing vor. Abschnitt 3 ist der Vorstellung des Stuttgart-Tübingen-Tagsets und des Universal Tagsets gewidmet. In Abschnitt 4 erklären wir unseren Experimentaufbau, und in Abschnitt 5 analysieren wir die Ergebnisse der Experimente. Abschnitt 6 ordnet unsere Ergebnisse in den Stand der Forschung ein, und Abschnitt 7 schließt den Artikel.

2 Bisherige Arbeiten

In diesem Abschnitt stellen wir einen Ausschnitt der Literatur vor, die den Zusammenhang zwischen POS-Tagging und Parsing unter verschiedenen Aspekten untersucht. Während diese Übersicht keinen Anspruch auf Vollständigkeit erhebt, so sollte sie doch einen Eindruck über vorherige Arbeiten schaffen.

Eine Grundfrage, die sich die meisten Arbeiten stellen, ist, wie am besten (in Bezug auf Parsingergebnisse und Parsinggeschwindigkeit) zwischen mehreren Tags für ein einzelnes Wort disambiguiert werden kann.

Einige Arbeiten untersuchen, ähnlich wie wir, die Rolle des POS-Tagging für Parsing innerhalb eines „Pipeline"-Ansatzes, bei dem die Ausgabe eines POS-Taggers als

Eingabe für einen Parser fungiert. So diskutieren Charniak et al. (1996) die optimale Wahl eines Taggers für PCFG-Parsing. Sie kommen zu dem Hauptergebnis, dass Markov-Modell-basierte Tagger, die jedes Wort mit einem einzelnen Tag versehen (also komplett disambiguieren), am besten geeignet sind. Die Autoren zeigen, dass PCFG-Parser schlechter zwischen POS-Tags disambiguieren und außerdem einen höheren Rechenaufwand verursachen. Anders als Charniak et al. verwendet Maier (2006) keinen separaten Tagger, sondern Gold-POS-Tags in der Parsereingabe (sprich, eine perfekte Tag-Disambiguierung). Er führt PCFG-Parsingexperimente mit den zwei Baumbanken NeGra und TiGer durch und kommt zu einem vergleichbaren Ergebnis: Mit Gold-POS-Tags steigt die Ausgabequalität des Parsers und der Rechenaufwand verringert sich.

Andere Arbeiten untersuchen, ebenfalls innerhalb eines „Pipeline"-Ansatzes, Möglichkeiten zur Reduktion von Ambiguität über die Modifikation von Tagsets bzw. des Lexikons durch Tagset-Reduktion oder Wort-Clustering. Lakeland (2005) beschäftigt sich mit lexikalisiertem Parsing à la Collins (1999). Ähnlich der neueren Arbeiten z.B. von Koo et al. (2008) oder von Candito und Seddah (2010) geht er die Frage nach der für das Parsen optimalen Disambiguierung durch Clustering auf lexikalischer Ebene an. Ein Wort-Cluster wird hierbei als Äquivalenzklasse von Wörtern gesehen und übernimmt gewissermaßen die Funktion eines POS-Tags, kann aber den Trainingsdaten angepasst werden. Le Roux et al. (2012) beschäftigen sich mit Datenknappheit auf lexikalischer Ebene beim PCFG-Parsing der morphologisch reichen Sprache Spanisch. Die Autoren benutzen eine Neuimplementierung des Berkeley Parsers. Sie zeigen, dass Parsingergebnisse sowohl durch eine Vereinfachung des POS-Tagsets als auch durch Lemmatisierung verbessert werden können, da beide Vorgehensweisen die Datenknappheit reduzieren.

Wie schon erwähnt, kann ein POS-Tag als eine Äquivalenzklasse von Wörtern gesehen werden. Da im „Pipeline"-Ansatz der Parsebaum über den POS-Tags erstellt wird, ist es jedoch möglich, dass ein POS-Tagset zwar aus linguistischer Sicht optimal ist, sich jedoch in Bezug auf Parsingergebnisse nicht optimal verhält, da für den Parsebaum relevante lexikalische Information durch das POS-Tagset verdeckt wird. Während lexikalisches Clustering wie bei Koo et al. (2008) dieses Defizit dadurch überwindet, dass (semi-)-automatisch „bessere" Cluster gesucht werden, kopieren andere Arbeiten lexikalische Information durch Baumbanktransformationen in den eigentlichen Baum. Dafür wird auch die in einigen Baumbanken bereits vorhandene Annotation von grammatischen Funktionen benutzt. Derartige Transformationen werden z.B. von Versley (2005) und Versley und Rehbein (2009) beschrieben. Seeker und Kuhn (2013) stellen einen Ansatz vor, der das „Pipeline"-Modell (unter Benutzung eines Dependenz-Parsers (Bohnet, 2010)) um eine zusätzliche Komponente ergänzt, die Kasus-Information als Filter für den Parser verwendet. Sie erreichen Verbesserungen für Ungarisch, Deutsch und Tschechisch und stellen dabei fest, dass es verschiedene Arten von morphologischer Komplexität gibt, die beim Parsing unterschiedlich behandelt werden müssen.

Der Zusammenhang von POS-Tagging und Parsing wurde nicht nur im Rahmen des baumbankbasierten Parsing untersucht, sondern auch im Rahmen des Parsing mit einer handgeschriebenen Grammatik und einer Disambiguierungskomponente.

Dalrymple (2006) untersucht die Rolle des POS-Tagging für das Parsing auf Basis des Systems von Riezler et al. (2002), das aus einer englischen *Lexical Functional Grammar*, einem Constraint-basierten Parser und einer Disambiguierungskomponente besteht. Die Autorin benutzt keinen separaten POS-Tagger. Sie bildet Äquivalenzklassen von Parserausgaben basierend auf deren Tag-Sequenzen. Aus der Zahl der gefundenen Äquivalenzklassen für einen einzelnen Satz schließt sie darauf, inwieweit perfektes Tagging für die Disambiguierung des Satzes helfen würde: Eine hohe Anzahl von Klassen lässt darauf schließen, dass die syntaktischen Analysen eines Satzes durch die POS-Tags differenzierbar sind, eine niedrige Anzahl deutet in die gegenteilige Richtung. Sie kommt zu dem Ergebnis, dass mit einem perfekten Tagger eine fünfzigprozentige Reduzierung der Parseambiguität zu erreichen wäre.

Watson (2006) stellt unter Benutzung des RASP-Systems (Briscoe und Carroll, 2002) weitergehende Untersuchungen zum Zusammenhang zwischen verschiedenen Modellen der Tagauswahl (einzelne/mehrere Tags pro Wort, Tagauswahl durch Parser oder Tagger) und Parsing an. Sie zeigt einerseits, ebenso wie andere Arbeiten, dass POS-Tagger eine bessere Tagauswahl treffen als Parser, und andererseits, dass es einen Trade-Off zwischen der Qualität von Parsingergebnissen und dem Zulassen multipler Tags pro Wort gibt. Prins und van Noord (2003) betrachten eine verwandte Frage für einen HPSG-Parser. Sie verwenden einen auf Parserausgaben trainierten Markov-Modell-POS-Tagger, um Parsereingaben vorzutaggen. Dies wirkt sich günstig sowohl auf Parsingergebnisse als auch auf die Parsingzeit aus. Curran et al. (2006) beschäftigen sich mit der Rolle von POS-Tagging für CCG- und TAG-Parsing. Sie berichten, dass eine zu frühe Disambiguierung von POS-Tags sich dann schlecht auf Parsingergebnisse auswirkt, wenn einzelne Tags sehr informativ sind (vgl. Supertagging gegenüber „normalem“ POS-Tagging (Bangalore und Joshi, 1999)). Die Arbeit von Yoshida et al. (2007) geht in dieselbe Richtung. Die Autoren verwenden einen HPSG-Parser mit vorgeschaltetem POS-Tagger und zeigen, dass das Zulassen von mehreren Tags pro Wort, d.h. das teilweise Übertragen der Tag-Disambiguierung an den Parser, sich unter bestimmten Bedingungen günstig auf Parsingergebnisse auswirkt.

Eine Reihe von Arbeiten entwirft Modelle für gleichzeitiges POS-Tagging, bzw. morphologische Segmentierung, und Parsing. Besonders interessant ist hier die Arbeit von Chen und Kit (2011). Sie gehen in gewisser Weise ebenfalls die Frage der Disambiguierung auf lexikalischer Ebene an. Basierend auf Arbeiten von Ratnaparkhi (1996) und Toutanova und Manning (2000) gehen die Autoren davon aus, dass lokale Merkmale für die Qualität von POS-Tagging entscheidend sind. Nichtsdestotrotz wird dies nicht berücksichtigt, wie sie bemerken, wenn auf ein „Pipeline“-Modell verzichtet wird, d.h. wenn dem Parser auch die Aufgabe des Tagging zufällt. Auf dieser Basis stellen sie ein erfolgreiches faktorisiertes Modell für das PCFG-Parsing vor, das das Parsing in ein diskriminatives lexikalisches Modell (mit lokalen Merkmalen) und das eigentliche Parsingmodell trennt. Die Modelle werden mittels eines *product-of-experts* (Hinton, 1999) kombiniert.

Kombinierte Modelle für gleichzeitiges POS-Tagging und Parsing lassen sich besonders in der Dependenzparsing-Literatur finden; hier zeigt sich vor allem eine Konzentration

auf Sprachen, die noch eine zusätzliche Segmentierung auf der Wortebene erfordern, so wie Chinesisch (Hatori et al., 2011) oder Hebräisch (Goldberg und Tsarfaty, 2008). Ein neuerer Ansatz von Bohnet und Nivre (2012) wurde auch auf dem Deutschen evaluiert. Ergebnisse zum POS-Tagging und Parsing des Deutschen mittels einer Constraint-Grammatik finden sich in Daum et al. (2003) sowie Foth et al. (2005). Da diese Arbeiten den Gegenstand unserer Arbeit jedoch nur am Rand berühren, verzichten wir auf einen weiteren Überblick.

3 Die Tagset-Varianten

In diesem Abschnitt werden die verschieden Tagset-Varianten beschrieben, die wir in unseren Experimenten verwenden. Wir beginnen mit dem Stuttgart-Tübingen-Tagset (STTS) (Schiller et al., 1995; Thielen und Schiller, 1994), das sich zum Standard POS-Tagset für das Deutsche entwickelt hat. Des Weiteren beschreiben wir die morphologische Erweiterung des STTS. Da die beiden hier verwendeten Baumbanken, TiGer und TüBa-D/Z (siehe Abschnitt 4.1), unterschiedliche Morphologie-Annotationen aufweisen, stellen wir beide Versionen der Morphologie vor. Die kleinste POS-Tagset-Variante ist das Universal Tagset (UTS) (Petrov et al., 2012).

Das UTS besteht aus 12 grundlegenden Tags, die in Tabelle 1 aufgeführt sind. Es wurde entwickelt, um als gemeinsames Tagset für eine große Anzahl von Sprachen verwendet zu werden, z.B. um sprachübergreifende POS-Tagger zu entwickeln, oder um Sprachen in Bezug auf das POS-Tagging zu vergleichen. Bei diesem Tagset fällt auf, dass nur sehr grundlegende Wortarten vertreten sind, es wird z.B. keine Unterscheidung zwischen verschieden Arten von Pronomen gemacht, und koordinierende und subordinierende Konjunktionen werden gemeinsam unter CONJ gruppiert. Dieses Tagset sollte eine hohe Qualität beim POS-Tagging garantieren, weil nur wenige, grobe Unterscheidungen getroffen werden. Jedoch stellt sich die Frage, inwieweit diese grobe Granularität genügend Information für eine syntaktische Analyse liefert.

Das STTS ist ein Tagset, das hauptsächlich basierend auf distributionellen Regularitäten des Deutschen entwickelt wurde. Es umfasst 54 Tags und modelliert damit wesentlich feinere Unterschiede als das UTS. Die STTS-Tags sind in den Tabellen 9 und 10 im Anhang aufgeführt.

Es fällt auf, dass das STTS nicht nur feinere Unterscheidungen von Wortarten macht, es modelliert auch die Finitheit bei Verben. Dies ist eine wichtige Unterscheidung für die syntaktische Analyse, aber es ist auch in bestimmten Fällen eine schwierige Aufgabe für den POS-Tagger, der nur einen sehr begrenzten lokalen Kontext verwendet: Deutsche Verbformen sind ambig in Bezug auf den Infinitiv und die Präsens-Plural-Form. Der Satz in (1) aus der TüBa-D/Z zeigt ein Beispiel hierfür: Das Verb `wirken` kann anhand des Kontexts von `wie Luken` nicht disambiguiert werden.

NOUN	Nomen
VERB	Verb
ADJ	Adjektiv
ADV	Adverb
PRON	Pronomen
DET	Determiner, Artikel
ADP	Preposition, Postposition
NUM	Numeral
CONJ	Konjunktion
PRT	Partikel
.	Interpunktion
X	alles andere

Tabelle 1: Die 12 Tags des Universal Tagsets.

ambig:	*
Genus:	maskulin (Masc), feminin (Fem), neutral (Neut)
Gradation:	Positiv (Pos), Komparativ (Comp), Superlativ (Sup)
Kasus:	Nominativ (Nom), Genitiv (Gen), Dativ (Dat), Akkusativ (Akk)
Modus:	Indikativ (Ind), Konjunktiv (Subj), Imperativ (Imp)
Numerus:	singular (Sg), plural (Pl)
Person:	1, 2, 3
Tempus:	Präsens (Pres), Präteritum (Past)

Tabelle 2: Die morphologischen Kategorien aus TiGer.

(1) es hat Passagen mit kleineren Fenstern , die aber nicht
PPER VAFIN NN APPR ADJA NN $, PRELS ADV PTKNEG
wie Luken wirken .
KOKOM NN VVFIN $.

Das STTS kann auch um eine morphologische Komponente erweitert werden. Dies ist bei den beiden Baumbanken TiGer und TüBa-D/Z geschehen, allerdings wurden unterschiedliche Entscheidungen getroffen. In der TiGer-Baumbank wird eine Menge von 585 verschiedenen morphologischen Merkmalskombinationen verwendet, die sich aus den in Tabelle 2 aufgelisteten Elementen zusammensetzen. Der Satz in (2) zeigt ein Beispiel der Kombination von STTS-Tags und -Morphologie, getrennt durch ein **%** Zeichen. Das Merkmal – bedeutet, dass keine morphologischen Merkmale vorliegen.

ambig:	*
Genus:	maskulin (m), feminin (f), neutral (n)
Kasus:	Nominativ (n), Genitiv (g), Dativ (d), Akkusativ (a)
Numerus:	singular (s), plural (p)
Person:	1, 2, 3
Tempus:	Präsens (s), Präteritum (t)
Modus:	Indikativ (i), Konjunktiv (k)

Tabelle 3: Die morphologischen Kategorien aus der TüBa-D/Z.

(2) Konzernchefs lehnen den
NN%Nom.Pl.Masc VVFIN%3.Pl.Pres.Ind ART%Acc.Sg.Masc
Milliardär als US-Präsidenten ab /
NN%Acc.Sg.Masc APPR%– NN%Acc.Sg.Masc PTKVZ%– $(%–

Aus der Menge der möglichen Kombinationen morphologischer Merkmale sind 271 verschiedene Kombinationen in der TiGer-Baumbank belegt. Daraus ergeben sich insgesamt 783 mögliche Kombinationen von STTS-Tags und Morphologie. Von diesen sind 761 im Trainingsset vorhanden. Wegen der hohen Anzahl von möglichen Kombinationen ist jedoch zu erwarten, dass Kombinationen, die im Development- oder Testset vorhanden sind, nicht im Trainingsset vorkommen, d.h. dass mit Datenknappheit zu rechnen ist. Deswegen haben wir ermittelt, wie viele der Kombinationen, die in den Development- und Testdaten erscheinen, auch in Trainingsset vorhanden sind. Es stellt sich heraus, dass 25% bzw. 30% nicht in den Trainingsdaten vorkommen. Man beachte dabei, dass die Anzahl der Kombinationen in den Test- und Developmentsets wesentlich geringer sind als im Trainingsset.

In der TüBa-D/Z gibt es insgesamt 132 mögliche morphologische Merkmalskombinationen. Sie setzen sich aus den in Tabelle 3 aufgelisteten Elementen zusammen. Der Satz in (3) zeigt ein Beispiel der Kombination von STTS-Tags und -Morphologie.

(3) Aber Bremerhavens AfB fordert jetzt
KON%– NE%gsn NE%nsf VVFIN%3sis ADV%–
Untersuchungsausschuß
NN%asm

Aus der Menge dieser möglichen Kombinationen morphologischer Merkmale sind 105 verschiedene Kombinationen in der TüBa-D/Z Baumbank belegt. Daraus ergeben sich insgesamt 524 verschiedene Kombinationen von STTS-Tags und Morphologie. Von diesen erscheinen 513 im Trainingsset; von den Kombinationen, die in unseren Development- und Testdaten erscheinen, sind 16% bzw. 18% in den Trainingsdaten nicht vorhanden.

Die Tatsache, dass die Kombinations-POS-Tagsets für beide Baumbanken mehrere hundert verschiedene Labels aufweisen, in Kombination mit der mangelnden Abdeckung

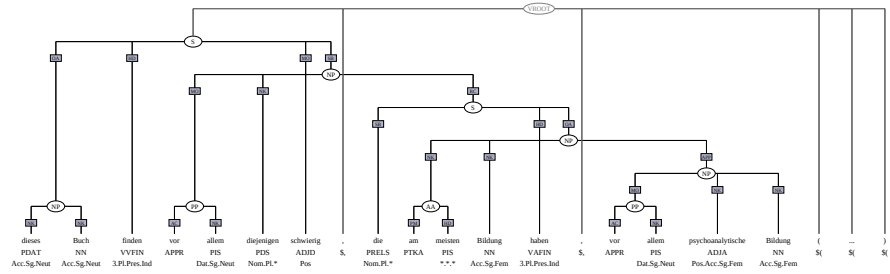

Abbildung 1: Ein Satz mit syntaktischer Annotation aus der TiGer-Baumbank.

des Trainingssets, lässt uns vermuten, dass die POS-Tagging-Ergebnisse für diese Variante weniger gut sein werden als für das Standard-STTS. Eine weitere Hypothese ist, dass das morphologische Tagset der TüBa-D/Z aufgrund seiner geringeren Größe besser zum Tagging geeignet ist als das TiGer-Tagset. Es ist jedoch offen, ob die morphologische Information beim Parsing gewinnbringend eingesetzt werden kann.

4 Aufbau der Experimente

4.1 Daten

Als Datensätze verwenden wir die zwei größten deutschen Baumbanken: TiGer (Brants et al., 2002) und die Tübinger Baumbank des Deutschen/Zeitungskorpus (TüBa-D/Z) (Telljohann et al., 2012). Beide Baumbanken basieren auf Zeitungstexten, die TiGer-Baumbank auf der Frankfurter Rundschau und die TüBa-D/Z auf der tageszeitung (taz). Beide Baumbanken verwenden das STTS mit minimalen Unterschieden. Aufbauend auf der POS-Ebene haben beide Baumbanken eine syntaktische Annotation bestehend aus einer Konstituentenstruktur, erweitert durch grammatische Funktionen. Die Baumbanken unterscheiden sich deutlich auf der Ebene der syntaktischen Annotationen, zum einen durch die unterschiedlichen Knoten- und Kantenlabels, zum anderen durch die Entscheidung in TiGer, kreuzende Kanten zu annotieren bzw. durch die Verwendung von topologischen Feldern (Höhle, 1986) in der TüBa-D/Z. Die Abbildungen 1 und 2 zeigen Beispielsätze aus den Baumbanken.

Für unsere Experimente verwenden wir Version 2.0 der TiGer-Baumbank, mit einem Umfang von 50 474 Sätzen, und Version 8.0 der TüBa-D/Z, mit einem Umfang von 75 408 Sätzen. Um unerwünschte Größeneffekte auszuschließen, verwenden wir die folgenden Größen für beide Baumbanken: die ersten 40 475 Sätze für das Training, die jeweils nächsten 5 000 Sätze für das Development- und das Testset. Dies entspricht der Aufteilung der TiGer-Baumbank durch Farkas und Schmid (2012).

Für beide Baumbanken müssen alle Interpunktionszeichen und alle anderen direkt an der Wurzel angehängte Elemente wie z.B. nicht angehängte Appositionen, in die

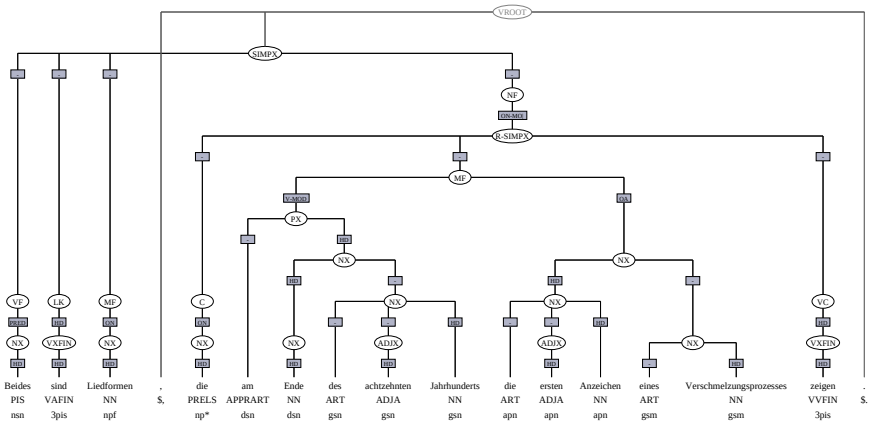

Abbildung 2: Ein Satz mit syntaktischer Annotation aus der TüBa-D/Z.

Konstituentenstruktur eingebaut werden. Hier folgen wir dem Ansatz von Maier et al. (2012).

Beide Baumbanken müssen außerdem in ein Format konvertiert werden, das von dem verwendeten Parser akzeptiert wird. Für die TiGer-Baumbank bedeutet dies, dass alle kreuzenden Kanten aufgelöst werden müssen, damit kontextfreie Regeln extrahiert werden können. Diese Auflösung wird in drei Schritten bewerkstelligt. In einem ersten Schritt wird für jede Phrase ein Kopf markiert. Dafür wird, soweit möglich, die vorhandene Kantenannotation („HD“) benutzt. Sollte ein Kopf so nicht zu bestimmen sein, kommen Heuristiken zum Einsatz. In einem zweiten Schritt wird die in Boyd (2007) beschriebene Transformation durchgeführt, d.h. für jeden kontinuierlichen Block einer diskontinuierlichen Konstituente wird ein separates Nichtterminal eingeführt; im gleichen Schritt wird aus der Menge bestehend aus dem ursprünglichen und den neu eingeführten Knoten derjenige Knoten markiert, unter dem letztendlich die als Kopf markierte Tochter verbleibt. In einem dritten Schritt erfolgt nun die eigentliche Auflösung der kreuzenden Kanten: Alle gesplitteten, aber nicht markierten Knoten werden entfernt und deren Töchter an den Mutterknoten des zu entfernenden Knotens gehängt. Für den Satz in Abbildung 1 bedeutet das, dass der Relativsatz S mit der grammatischen Funktion RC aus der NP `vor allem diejenigen` hochgereicht wird an die Mutter der NP, S.

4.2 POS-Tagger

Wir verwenden vier verschiedene POS-Tagger, die ein möglichst breites Spektrum an verschiedenen Ansätzen zum POS-Tagging abdecken, namentlich einen *Trigramm/Markov-*

Modell-Tagger, einen *Maximum-Entropy*-Tagger, einen *Conditional-Random-Field*-Tagger und einen *Support-Vector-Machine*-Tagger. Alle POS-Tagger werden mit den Standardeinstellungen verwendet.

TnT (Brants, 2000, 1998), kurz für *Trigrams and Tags*, ist ein Markov-Modell-POS-Tagger, der eine Interpolation zwischen Uni-, Bi- und Trigrammen als Wahrscheinlichkeitsmodell verwendet. Dieser POS-Tagger besitzt ein Modul zur Behandlung von unbekannten Wörtern, das einen *trie* aus Suffixen verwendet, der aus Hapax Legomena aus dem Trainingskorpus extrahiert wird. TnT ist auch 15 Jahre nach seiner Entwicklung das beste verfügbare Modul für diesen Zweck.

Der **Stanford log-linear POS Tagger** (Toutanova et al., 2003; Toutanova und Manning, 2000) basiert auf einem *Maximum-Entropy*-Modell. Dies bedeutet, dass dieser POS-Tagger, ebenso wie die beiden nachfolgend beschriebenen Tagger, im Hintergrund ein diskriminatives maschinelles Lernverfahren statt eines Markov-Modells verwendet. Dies hat den Vorteil, dass er mit Merkmalen arbeiten kann, die weit über den linken Kontext von 1-2 Wörtern hinausgehen. Der Nachteil des Verfahrens ist, dass keine globale Optimierung stattfindet. Wir verwenden den POS-Tagger mit dem bidirektionalen Modell basierend auf einem Kontext von 5 Wörtern.

Der dritte POS-Tagger basiert auf *Conditional Random Fields*. Diese sind zum Annotieren von Sequenzen besonders geeignet (Lafferty et al., 2001). Wir verwenden die Anwendung für *sequence tagging* in **MALLET** (McCallum, 2002).

Der letzte von uns verwendete Tagger, **SVMTool** (Giménez und Màrquez, 2004), basiert auf *Support Vector Machines* (SVMs), genauer gesagt auf der SVM-Implementierung SVM^{light} (Joachims, 1999). SVMs haben sich als extrem gut geeignet für Problemstellungen in der Computerlinguistik erwiesen. Wir verwenden das Modell, das einen Satz von links nach rechts in einem Durchgang analysiert, mit dem Standard-Merkmalsset, das Trigramme aus Wörtern und POS-Tags ebenso verwendet wie Wortlänge, Präfix- und Suffixinformation.

4.3 Parser

Als Parser verwenden wir den **Berkeley Parser** (Petrov et al., 2006), einen Konstituentenparser, der seine Grammatik dadurch verfeinert, dass er die syntaktischen Labels in Unterklassen aufteilt, bzw. Labels verschmilzt, die in ähnlichen Umgebungen vorkommen. Die im Berkeley Parser implementierte Technik für das automatische Aufteilen und Verschmelzen von Labels ist der derzeitige Stand der Technik für das Parsing des Deutschen, basierend auf einem individuellen Parser ohne Reranking. Wir beschränken uns hier auf grundlegendes Konstituentenparsing, d.h. wir verwenden keine grammatischen Funktionen beim Parsing. Der Parser wurde in 6 Iterationen trainiert.

4.4 Experimente

In diesem Beitrag betrachten wir 4 Fragestellungen:

1. STTS-Variationen: In diesem Satz von Experimenten untersuchen wir, wie sich die Granularität von STTS auf die Qualität der Ausgaben der verschiedenen POS-Tagger auswirkt. Wir betrachten das Universal Tagset (UTS), das Standard-STTS und das durch Morphologie erweiterte STTS.

2. STTS-Variationen und Parsing: Hier untersuchen wir, wie sich die unterschiedliche Granularität von STTS auf die Parsingqualität auswirkt. Um von der unterschiedlichen Qualität der POS-Tagger zu abstrahieren, verwenden wir für dieses Experiment erst Gold-POS-Tags und wiederholen die Experimente dann mit automatisch getaggten Texten.

3. Morphologische Variationen: In diesem Experiment untersuchen wir verschiedene Untermengen von morphologischen Merkmalen. Hierfür verwenden wir nur TnT.

4. Morphologische Variationen und Parsing: Hier untersuchen wir den Einfluss der morphologischen Merkmalsets auf die Qualität des Parsers. Hierzu verwenden wir erst Gold-POS-Tags, dann von TnT erzeugte Tags.

Der erste Satz von Experimenten soll zum einen untersuchen, in welchem Ausmaß sich die Art des POS-Taggers auf die Qualität der Ergebnisse auswirkt. Zum anderen wollen wir feststellen, wie sich die unterschiedliche Granularität der STTS-Varianten auf die Tagger-Qualität auswirkt. Unsere Hypothese ist: Je mehr Informationen im Tagset vorhanden sind, desto schwieriger ist die Disambiguierung von verschiedenen Tags. Es ist außerdem anzunehmen, dass unterschiedliche POS-Tagger unterschiedlich gut mit den verschiedenen Tagset-Größen zurechtkommen.

Der zweite Satz von Experimenten dient dazu, den Einfluss der im Tagset vorhandenen Informationen auf die Parsingergebnisse zu bestimmen. Die Frage hier ist, ob es ausreicht, dem Parser wenige Informationen zu geben, oder ob er von der hohen Informationsdichte im morphologisch angereicherten Tagset ebenfalls profitieren kann. Hier ist anzumerken, dass der Berkeley Parser zu feine Unterscheidungen in den Tags in seinem Lernverfahren zusammenfassen kann.

Der dritte Satz von Experimenten soll bestimmen, ob sich eine bestimmte Unterkategorie von morphologischen Merkmalen zuverlässig mit einem POS-Tagger ermitteln lässt. Die ausgewählten Unterkategorien gründen auf der Intuition der Autoren, welche Art von Merkmalen potentiell hilfreich für das Parsing sein könnte. Die erste Unterkategorie (Kongruenz) beschränkt sich auf die nominalen Kongruenzmerkmale, Genus, Numerus und Kasus. Die zweite Unterkategorie besteht nur aus Kasusmerkmalen. Hier ist nicht zu erwarten, dass ein POS-Tagger, mit seinem sehr eingeschränkten lokalen Kontext und ohne morphologische Analyse, diese Unterscheidung immer erfolgreich treffen kann. Vor allem wegen des Synkretismus von Nominativ/Akkusativ bzw. Genitiv/Dativ im Deutschen kann eine solche Unterscheidung nur auf einer syntaktischen Analyse basierend getroffen werden. Die dritte Unterkategorie besteht aus den Numerusmerkmalen, die vierte kombiniert Numerus mit Person. In der letzten Unterkategorie werden alle verbalen Merkmale verwendet.

Der vierte Satz von Experimenten untersucht, ob sich die Trends innerhalb des POS-Tagging mit morphologischen Unterkategorien auch auf das Parsing übertragen lassen. Die Frage ist hier, anders ausgedrückt, ob es wichtiger ist, dem Parser wichtige, aber u.U. unzuverlässige Information zur Verfügung zu stellen, oder ob es besser ist, ein grobkörnigeres Tagset zu verwenden, das aber zuverlässig automatisch annotiert werden kann.

Zu beachten ist, dass unsere Untersuchung task-basiert ist, d.h. wir untersuchen, wie sich die einzelnen Tagsets in Bezug zur Aufgabe des Taggings verhalten. Dies stellt keinen direkten Vergleich zwischen einzelnen Tagsets dar.

4.5 Evaluierung

Die Evaluierung auf der POS-Ebene berechnet die Korrektheit des POS-Taggers in Bezug auf *accuracy*. Wir verwenden das Evaluierungsskript von TnT, da dieses uns erlaubt, auch die Korrektheit von bekannten und unbekannten Wörtern zu berechnen.

Für die Evaluierung des Parsers geben wir *precision*, *recall*, und *F-Score* an. Wir verwenden die Implementierung `evalb-lcfrs`.[1] Diese Implementierung verhält sich auf kontinuierlichen Konstituenten, wie hier vorhanden, ebenso wie die Standard-Software Evalb (ohne Parameterdatei).[2] Bei der Parserevaluierung werden die POS-Tags nicht berücksichtigt.

Da wir keine Optimierung der POS-Tagger und des Parsers vornehmen und da sich die Development- und Testsets sehr unterscheiden, geben wir die Ergebnisse jeweils für beide Datensätze an.

Alle Experimente werden auf einem 3,16 GHz Intel Xeon mit jeweils 32 GB maximalem Speicher pro Experiment durchgeführt.

5 Ergebnisse

5.1 Drei Varianten von STTS

Hier untersuchen wir, welchen Effekt die unterschiedliche Granularität des POS-Tagsets auf die Qualität von verschiedenen POS-Taggern hat. Die Ergebnisse sind in Tabelle 4 zusammengefasst.

Das erste Resultat dieser Experimente ist, dass sich die beiden Baumbanken kaum in den Ergebnissen unterscheiden. Daraus können wir schließen, dass beide Textquellen gleich schwierig für die POS-Tagger sind. Es ist jedoch auffällig, dass das Developmentset und das Testset jeweils unterschiedliche Ergebnisse aufweisen. Es gibt also innerhalb der jeweiligen Baumbanken deutliche Unterschiede.

Wenn man die verschiedenen POS-Tagger vergleicht, dann fällt auf, dass SVMTool die besten Ergebnisse für das minimalistische Universal Tagset (UTS) liefert.[3]Für das

[1] Siehe `https://github.com/wmaier/evalb-lcfrs`.

[2] Siehe `http://nlp.cs.nyu.edu/evalb/`.

[3] Hier muss allerdings darauf hingewiesen werden, dass ein Vergleich über die verschiedenen Tagset-Varianten hinweg nur mit Vorsicht zu genießen ist, da die unterschiedlichen Tagset-Größen

	TiGer		TüBa-D/Z	
POS-Tagger	Development	Test	Development	Test
	UTS			
MALLET	91,67%	90,22%	93,54%	94,01%
Stanford	97,88%	96,83%	97,11%	97,26%
SVMTool	**98,54%**	**98,01%**	**98,09%**	**98,28%**
TnT	97,94%	97,48%	97,72%	97,92%
	STTS			
MALLET	92,45%	90,29%	88,81%	89,12%
Stanford	96,26%	95,15%	95,63%	95,79%
SVMTool	97,06%	96,22%	96,46%	96,69%
TnT	**97,15%**	**96,29%**	**96,92%**	**97,00%**
	STTSmorph			
MALLET	–	–	–	–
Stanford	–	–	–	–
SVMTool	82,47%	79,53%	80,33%	81,31%
TnT	**85,77%**	**82,77%**	**84,67%**	**85,45%**

Tabelle 4: Die Ergebnisse für verschiedene POS-Tagger mit verschiedenen Tagset-Varianten: das Universal Tagset (UTS), das STTS und das STTS mit Morphologie (STTSmorph).

Standard-STTS und für die Variante mit morphologischer Information erweist sich TnT, der bei weitem älteste POS-Tagger, als der zuverlässigste. Bei der morphologischen Variante liegt der Unterschied zwischen SVMTool und TnT bei mehr als 3% für TiGer und bei mehr als 4% für die TüBa-D/Z. Eine mögliche Erklärung für die guten Ergebnisse von TnT auf der morphologischen Variante ist, dass Markov-Modelle weniger Trainingsdaten benötigen als die anderen Tagger, die auf diskriminativen Lernverfahren basieren. Dies ist ein wichtiger Vorteil in Situationen, in denen viele POS-Tags nur selten in den Trainingsdaten vorkommen.

Bei einer Betrachtung der Ergebnisse des Stanford Taggers und von MALLET fällt zunächst auf, dass wir keine Ergebnisse für die morphologische Variante haben. Das liegt daran, dass für das Training von MALLET zwei Wochen nicht ausreichten. Zu diesem Zeitpunkt haben wir die Experimente abgebrochen. Der Stanford Tagger brach das Tagging mit einer Fehlermeldung ab, die uns vermuten lässt, dass der Tagger die hohe Anzahl von Merkmalen nicht verarbeiten kann, die bei dieser Variante anfällt. Bei den anderen Varianten schneidet der Stanford Tagger geringfügig schlechter ab als der jeweils zweitplatzierte POS-Tagger. Im Gegensatz dazu schneidet MALLET bedeutend schlechter ab. Der Unterschied zwischen MALLET und dem bestplatzierten POS-Tagger beträgt 7-8% für TiGer und 4-5% für die TüBa-D/Z. Diese Differenz

auch einen Einfluss auf die Schwierigkeit der Aufgabe haben können. Unser Interesse in allen Experimenten beschränkt sich darauf, welcher Prozentsatz von Wörtern korrekt annotiert wurde.

	TiGer				TüBa-D/Z			
	Dev.		Test		Dev.		Test	
POS-Tag.	bek.	unbek.	bek.	unbek.	bek.	unbek.	bek.	unbek.
	UTS							
MALLET	92,83	77,66	92,09	73,30	95,47	74,84	95,75	75,53
Stanford	**99,05**	91,85	**98,78**	87,70	**98,94**	79,30	**98,92**	79,69
SVMTool	98,81	95,26	98,41	94,45	98,63	92,89	98,66	94,27
TnT	98,06	**96,50**	97,67	**95,74**	98,07	**94,28**	98,25	**95,25**
	STTS							
MALLET	94,66	65,70	93,40	62,25	91,44	63,27	91,67	62,14
Stanford	**98,16**	73,56	**97,75**	71,60	**97,96**	73,04	**97,97**	72,64
SVMTool	97,86	87,41	97,26	86,82	97,50	86,47	97,60	87,05
TnT	97,80	**89,25**	97,21	**87,95**	97,65	**89,78**	97,72	**89,33**
	STTSmorph							
SVMTool	84,67	55,89	82,40	53,58	82,87	55,81	83,61	57,01
TnT	**87,62**	**63,41**	**85,55**	**57,65**	**86,91**	**62,95**	**87,61**	**62,55**

Tabelle 5: Die Ergebnisse für verschiedene POS-Tagger für bekannte und unbekannte Wörter.

liegt zum Teil sicher daran, dass MALLET kein designierter POS-Tagger sondern ein generelles Sequenzlernverfahren ist, d.h. es existieren z.B. keine Strategien, wie unbekannte Wörter zu behandeln sind.

Bei einem Vergleich der Tagsetvarianten bestätigt sich unsere Hypothese: Je mehr Informationen vorhanden sind, desto schwieriger ist die Disambiguierung für die POS-Tagger. Das UTS kann am zuverlässigsten annotiert werden, hier liegen die Ergebnisse bei über 97% für TnT. Das Standard-STTS verzeichnet nur minimale Einbußen. Dies bedeutet, dass ein größeres Tagset nicht unbedingt schwieriger zu taggen ist, wenn die Tags die tatsächliche Distribution modellieren, die durch die in den Taggern vorhandene Information abgebildet ist.

Die morphologische Variante von STTS dagegen resultiert in einem wesentlich größeren Tagset und daher auch in einem wesentlich schwierigeren Problem. Dies zeigt sich in den Ergebnissen, die durchgehend 12-14% schlechter sind als dieselben Ergebnisse für das Standard-STTS. Unsere Hypothese, dass das morphologische Tagset der TüBa-D/Z aufgrund seiner geringeren Größe zu besseren Ergebnissen führt, ist nicht bestätigt: die Unterschiede zwischen dem Developmentset und dem Testset von TiGer sind größer als die Unterschiede zwischen den Baumbanken.

In einer weiteren Auswertung unterscheiden wir zwischen bekannten und unbekannten Wörtern im Development- und Testset. Bekannte Wörter definieren wir als diejenigen Wörter, die im Trainingsset erscheinen, unbekannte Wörter sind solche, die im Trainingsset nicht vorhanden sind. In TiGer sind 7,64% der Wörter im Developmentset unbekannt, im Testset 9,96%. In TüBa-D/Z sind 9,36% der Wörter im Developmentset

unbekannt, im Testset 8,64%. Die Auswertung für bekannte und unbekannte Wörter findet sich in Tabelle 5. Diese Ergebnisse zeigen, dass der Stanford Tagger für das UTS und das STTS für bekannte Wörter die besten Ergebnisse erreicht, während TnT durchgehend die besten Ergebnisse für unbekannte Wörter erreicht. Für das morphologisch erweiterte STTS erreicht TnT außerdem die besten Ergebnisse auch für bekannte Wörter. Wir erinnern daran, dass diese Variante mit MALLET und dem Stanford Tagger nicht getaggt werden konnte.

Die Ergebnisse zeigen außerdem, dass MALLET, wie erwartet, bei unbekannten Wörtern deutlich schlechter abschneidet als die anderen POS-Tagger, was darauf zurückzuführen ist, dass MALLET keine separate Strategie zur Verarbeitung unbekannter Wörter besitzt. Darüber hinaus sind jedoch MALLETs Ergebnisse für bekannte Wörter auch deutlich schlechter als die der anderen Tagger, was bedeutet, dass das Sequenzmodell für das POS-Tagging nicht optimal ist, zumindest nicht ohne Anpassung an die vorliegende Aufgabe.

5.2 Parsing mit unterschiedlicher Tagset-Granularität

In diesem Satz von Experimenten untersuchen wir, wie sich die verschiedenen Ergebnisse beim POS-Tagging mit unterschiedlicher Tagset-Granularität auf das Parsing auswirken. Für diese Experimente verwenden wir die syntaktischen Annotationen der beiden Baumbanken, allerdings ohne grammatische Funktionen. Wir extrahieren dieselben Trainings-, Development-, und Testsets, die für die vorhergehenden Experimente verwendet wurden. Wir trainieren den Parser mit Gold-POS-Tags. Um die Leistung des Tagging-Mechanismus des Berkeley Parsers zu untersuchen, führen wir ein Experiment durch, in dem wir den Parser POS-Tags erzeugen lassen. In allen anderen Experimenten geben wir Wort/POS-Tag-Paare in die Parsereingabe ein, d.h. der Parser wird mit einer Option gestartet, die ihn POS-Tags erwarten lässt. In unserem Fall sind dies zum einen die Gold-POS-Tags, und zum anderen die von SVMTool und von TnT erzeugten Tags. Es ist anzumerken, dass der Berkeley Parser diese vorgegebenen POS-Tags ändert, falls er, basierend auf den Eingabe-Tags, keine syntaktische Analyse findet.

Die Ergebnisse dieser Experimente sind in Tabelle 6 aufgelistet. Ein erster Blick auf die Ergebnisse zeigt, dass es beträchtliche Unterschiede zwischen der TiGer- und der TüBa-D/Z-Baumbank gibt: Bei der TiGer-Baumbank bewegen sich die F-Scores zwischen 78,00 und 87,06 mit Gold-POS-Tags; bei der TüBa-D/Z liegen die F-Scores zwischen 91,91 und 94,57, ebenfalls basierend auf Gold-Tags. Dies steht im Kontrast zu den POS-Tagging-Ergebnissen in Tabelle 4, die zeigen, dass die Unterschiede zwischen den beiden Baumbanken in Bezug auf das POS-Tagging wesentlich geringer sind. Dies ist ein bekanntes Phänomen (Kübler et al., 2006; Rehbein und van Genabith, 2007; Kübler et al., 2008), das sich durch die Unterschiede in den syntaktischen Annotationen der beiden Baumbanken erklären lässt. Es bleibt jedoch ungeklärt, ob die Unterschiede zum Großteil dadurch entstehen, dass die Evaluierungsmetrik Annotationen mit einer großen Anzahl von Knoten, wie in der TüBa-D/Z, bevorzugt, oder ob die hierarchischeren

Parser-Eingabe	TiGer Dev. prec.	TiGer Dev. rec.	TiGer Dev. F	TiGer Test prec.	TiGer Test rec.	TiGer Test F	TüBa-D/Z Dev. prec.	TüBa-D/Z Dev. rec.	TüBa-D/Z Dev. F	TüBa-D/Z Test prec.	TüBa-D/Z Test rec.	TüBa-D/Z Test F
	UTS											
gold	87,43	85,85	86,63	83,62	81,42	82,51	92,24	91,59	91,91	92,63	91,92	92,27
SVM	85,74	84,47	85,10	81,06	79,26	80,15	89,13	88,86	89,00	89,69	89,42	89,55
TnT	84,48	83,39	83,93	79,50	78,05	78,77	88,55	88,11	88,32	89,19	88,69	88,94
Berkeley	82,99	81,83	82,41	78,87	77,09	77,97	90,51	89,72	90,11	91,08	90,25	90,67
	STTS											
gold	87,57	86,55	87,06	83,30	82,01	82,65	94,28	94,00	94,14	94,77	94,38	94,57
SVM	83,56	83,54	83,55	77,81	77,77	77,79	88,63	89,68	89,15	89,53	90,41	89,97
TnT	84,09	83,83	83,96	78,69	78,43	78,56	90,26	90,53	90,40	90,66	90,94	90,80
Berkeley	86,48	85,47	**85,97**	81,34	79,94	**80,64**	92,37	91,92	**92,15**	92,94	92,52	**92,73**
	STTSmorph											
gold	82,47	83,14	82,80	77,64	78,37	78,00	93,21	93,44	93,33	93,87	93,87	93,87
SVM	72,02	77,45	74,63	65,62	71,32	68,35	83,36	86,61	84,95	84,55	87,51	86,00
TnT	75,90	79,67	77,74	69,89	73,61	71,70	85,25	87,61	86,41	86,32	88,39	87,34
Berkeley	80,41	79,97	80,19	75,13	74,57	74,85	91,16	91,05	91,11	91,78	91,60	91,69

Tabelle 6: Die Ergebnisse des Berkeley Parsers auf verschiedenen Tagset-Varianten.

Annotationen in der TüBa-D/Z eine größere Generalisierung der Regeln und damit ein zuverlässigeres Parsing ermöglichen.

Ein Vergleich der Ergebnisse zwischen den Tags von SVMTool und TnT zeigt, dass die Tendenzen in beiden Baumbanken erhalten bleiben: Während SVMTool bei der Verwendung des Universal Tagsets die besten Ergebnisse erreicht (TiGer F-Scores: 85,10 und 80,15; TüBa-D/Z F-Scores: 89,00 und 89,55), erreicht TnT bessere Ergebnisse basierend auf STTS (TiGer F-Scores: 83,96 und 78,56; TüBa-D/Z F-Scores: 90,40 und 90,80) und auf STTSmorph (TiGer F-Scores: 77,74 und 71,70; TüBa-D/Z F-Scores: 86,41 und 87,34). Interessanterweise zeigt sich, dass die vom Parser selbst verteilten Tags zu besseren Parsingergebnissen führen (außer für TiGer mit UTS). Die entsprechenden F-Scores liegen i.d.R. zwischen den F-Scores für Gold-POS-Tags und den F-Scores für TnT, was zeigt, dass das Tagging von den Informationen im Split/Merge-Modell des Berkeley Parser profitiert – die Teilung und Verschmelzung von Tags hängt auch vom syntaktischen Kontext ab. Im Vergleich zu dem Experiment, in dem wir Gold-Standard POS-Tags verwenden zeigt sich, dass die Fehlerraten vom POS-Tagging zum Parsing größtenteils konstant bleiben. D.h. wenn der POS-Tagger eine Fehlerrate von 2% hat, verschlechtern sich die Parsingergebnisse von Gold-Tags zu automatischen Tags um etwa dieselbe Größenordnung. Das bedeutet, dass Fehler im POS-Tagging zwar einen negativen Einfluss auf die Qualität der Parses haben, dass aber nur in minimalem Umfang Nachfolgefehler auftreten. Beim morphologischen Tagset, STTSmorph, ist die Differenz zwischen den Parser-Ergebnissen mit Gold- und TnT-Tags sogar deutlich geringer als die Fehlerrate beim POS-Tagging, was bedeutet, dass Fehler in der Morphologie z.T. keine Auswirkung auf die Baumstruktur haben.

Vergleicht man die Performanz des Parsers basierend auf den verschiedenen Varianten des Tagsets, zeigen die beiden Baumbanken unterschiedliche Tendenzen: In TiGer sind die Unterschiede zwischen UTS und STTS minimal, d.h. TiGer kann nicht von der zusätzlichen Information im STTS profitieren. In der TüBa-D/Z dagegen bewirkt der Wechsel von UTS zu STTS eine Verbesserung von ca. 2%, d.h. die feinkörnigere Information im STTS ist hilfreich für die Disambiguierung von Konstituenten. Ein Grund dafür ist in der Tatsache zu finden, dass die TüBa-D/Z strukturelle Unterschiede zwischen Haupt- und untergeordneten Sätzen macht: nebenordnende Konjunktionen (KON) werden unter das Feld KOORD gruppiert, unterordnende Konjunktionen (KOUS und KOUI) unter C. Außerdem werden Relativsätze mit einem eigenen Knotenlabel (R-SIMPX) gekennzeichnet (siehe Abbildung 2 für ein Beispiel). Aus diesem Grund ist die Unterscheidung zwischen nebenordnenden und unterordnenden Konjunktionen wichtig. Diese werden im UTS unter einem gemeinsamen Tag (CONJ) gruppiert. In der TiGer-Baumbank dagegen werden alle Sätze unter S gruppiert (siehe Abbildung 1), da die Funktionsinformation beim Parsing nicht verwendet wurde. D.h. diese Unterscheidung im STTS ist nicht von Bedeutung. Eine andere für die TüBa-D/Z wichtige Unterscheidung ist diejenige zwischen finiten und infiniten Verben, da diese sich in der Verbalphrase widerspiegelt.

Die morphologischen Informationen in STTSmorph haben einen negativen Einfluss auf die Parsingergebnisse: Der F-Score für TiGer fällt um ca. 4,5% bei Gold-Tags und

	TiGer		TüBa-D/Z	
Morphologie	Dev.	Test	Dev.	Test
STTS	**97,15%**	**96,29%**	**96,92%**	**97,00%**
STTSmorph	85,77%	82,77%	84,67%	85,45%
Kongruenz	86,04%	83,08%	84,96%	85,77%
Kasus	88,10%	86,47%	87,48%	87,91%
Numerus	95,60%	94,19%	95,24%	95,41%
Numerus + Person	95,55%	94,11%	95,18%	95,24%
Verb-Merkmale	97,03%	96,02%	96,55%	96,44%

Tabelle 7: Die Ergebnisse für TnT mit verschiedenen morphologischen Varianten.

um ca. 6% bei POS-Tags von TnT. Bei der TüBa-D/Z ist die Differenz etwas geringer, sie liegt um 1% bei Gold-Tags und zwischen 2,5% und 3,5% bei TnT Tags. Daraus können wir schließen, dass der Parser die morphologische Information nicht sinnvoll verwenden kann, selbst wenn Gold-POS-Tags vorhanden sind.

Die insgesamt besten Parsingergebnisse erreichen wir bei TiGer und TüBa-D/Z mit dem internen POS-Tagging des Berkeley Parsers. Im „Pipeline“-Modell mit automatischen POS-Tags erreichen wir die besten Ergebnisse für TiGer mit der Kombination von SVMTool und STTS und für TüBa-D/Z mit TnT und STTS. Diese Ergebnisse bestätigen unsere Vermutung, dass das morphologische Tagset zu viel Information enthält, die automatisch nicht hochwertig genug annotiert werden kann, um für das Parsing hilfreich zu sein. Daraus ergibt sich die Frage, ob es Untermengen von morphologischen Merkmalen gibt, die zuverlässig mittels eines POS-Taggers annotiert werden können und die für das Parsing hilfreich sind. Diese Frage wird in den nächsten Abschnitten untersucht.

5.3 Morphologische Varianten

In diesem Satz von Experimenten untersuchen wir, ob es syntaktisch relevante Untermengen von morphologischen Merkmalen gibt, die sich zuverlässiger annotieren lassen als die komplette Menge morphologischer Merkmale. Diese Experimente wurden mit TnT durchgeführt, weil er sich in den vorhergehenden Experimenten als der zuverlässigste POS-Tagger erwiesen hat. Die Ergebnisse dieser Experimente sind in Tabelle 7 aufgelistet. Zu Vergleichszwecken wiederholen wir die Ergebnisse für das STTS und die STTS-Variante mit kompletter Morphologie (STTSmorph) aus Tabelle 4.

Die Ergebnisse zeigen, dass es morphologische Untermengen gibt, die zuverlässige Ergebnisse ermöglichen: Wenn nur Verb-Merkmale verwendet werden, erreichen wir Ergebnisse, die nur leicht unter denen der Standard-STTS-Variante liegen. Bei Numerus + Person und bei Numerus alleine beträgt die Differenz ca. 2%. Die Varianten, die nur Kasus oder alle Kongruenzmerkmale verwenden, sind leicht (Kasus) bzw. deutlich (Kongruenz) schlechter als die komplette Morphologie. Daraus ergibt sich die Frage,

inwieweit sich diese Ergebnisse auf das Parsing übertragen lassen. Dies wird im nächsten Abschnitt untersucht.

Es ist weiterhin auffällig, dass in den meisten Fällen die Unterschiede zwischen dem Development- und dem Testset in TiGer größer sind als die Unterschiede zwischen TiGer und der TüBa-D/Z. Ein Grund dafür ist sicher die hohe Anzahl der morphologischen Tags in TiGer, die zu einem hohen Maß von Tags rührt, die im Development- und Testset vorkommen, aber nicht im Trainingsset.

5.4 Parsing mit morphologischen Varianten

In diesem Satz von Experimenten untersuchen wir, wie sich die morphologischen Varianten des STTS aus den vorhergehenden Experimenten auf das Parsing auswirken. Wir verwenden dasselbe Setup wie in Abschnitt 5.2, d.h. wir verwenden keine grammatischen Funktionen beim Parsing, wir verwenden Gold-POS-Tags in der jeweiligen morphologischen Kombination fürs Training, und jeweils die Wörter in Kombination mit den Gold-POS-Tags oder den Ausgaben von TnT als Eingabe für den Parser.

Die Ergebnisse dieser Experimente finden sich in Tabelle 8. Zu Vergleichszwecken wiederholen wir die Ergebnisse für die STTS-Varianten mit und ohne komplette Morphologie (STTSmorph und STTS) aus Tabelle 6.

Die Ergebnisse zeigen, dass alle morphologischen Variationen basierend auf Gold-Tags bessere Parsingergebnisse erzielen als die Varianten mit kompletter Morphologie. Dies gilt für beide Baumbanken. Wenn die POS-Tags auf TnT-Ausgaben basieren, verbessern sich die Ergebnisse in allen Fällen, bis auf die Kombination von Kongruenzmerkmalen für die TiGer-Baumbank. In diesem Fall sind die Ergebnisse für die komplette Morphologie um ca. 2% besser. Des Weiteren fällt auf, dass die Ergebnisse für TüBa-D/Z mit kompletter Morphologie und mit Kongruenzmerkmalen extrem ähnlich (für das Developmentset) oder identisch (für das Testset) sind. Eine weitere Analyse zeigt, dass die Parser-Ergebnisse nur minimale Unterschiede aufweisen, obwohl die POS-Tagging-Ergebnisse sich deutlicher unterscheiden. Daraus können wir schließen, dass die Kongruenzmerkmale einen Großteil der Merkmale in der TüBa-D/Z ausmachen, die schwierig für POS-Tagger sind, die jedoch einen Einfluss auf den Parsebaum haben. D.h. wenn diese Merkmale mit größerer Genauigkeit annotiert werden könnten, hätten sie wohl einen positiven Einfluss auf die Parsequalität.

Bei einer näheren Betrachtung der Ergebnisse wird deutlich, dass für die TüBa-D/Z die Ergebnisse für die Merkmalskombinationen Numerus + Person und Verb-Merkmale mit den Ergebnissen für das Standard-STTS vergleichbar sind: Für das STTS erreicht der Parser einen F-Score von 90,40 bzw. 90,80 basierend auf TnT POS-Tags. Für Numerus + Person erreicht der Parser 90,03 und 90,37 und für Verb-Merkmale 90,41 und 90,88. Dies bedeutet, dass wir diese morphologischen Merkmale zuverlässig genug annotieren können, dass sie aber für das Konstituenzparsing nicht aussagekräftig genug sind. Im Gegensatz dazu sind die Ergebnisse für die TiGer-Baumbank durchgehend und deutlich schlechter als die Ergebnisse basierend auf dem UTS. D.h., in der TiGer-

Parser-Eingabe	TiGer Dev. prec.	TiGer Dev. rec.	TiGer Dev. F	TiGer Test prec.	TiGer Test rec.	TiGer Test F	TüBa-D/Z Dev. prec.	TüBa-D/Z Dev. rec.	TüBa-D/Z Dev. F	TüBa-D/Z Test prec.	TüBa-D/Z Test rec.	TüBa-D/Z Test F
	STTS											
gold	87,57	86,55	87,06	83,30	82,01	82,65	94,28	94,00	94,14	94,77	94,38	94,57
TnT	84,09	83,83	**83,96**	78,69	78,43	**78,56**	90,26	90,53	90,40	90,66	90,94	90,80
	STTSmorph											
gold	82,47	83,14	82,80	77,64	78,37	78,00	93,21	93,44	93,33	93,87	93,87	93,87
TnT	75,90	79,67	77,74	69,89	73,61	71,70	85,25	87,61	86,41	86,32	88,39	87,34
	Kongruenz											
gold	82,94	83,36	83,15	78,56	78,73	78,64	93,41	93,68	93,55	93,91	94,00	93,96
TnT	73,24	78,13	75,60	67,12	72,31	69,62	85,56	87,97	86,75	86,32	88,38	87,34
	Kasus											
gold	86,36	85,50	85,93	82,68	81,72	82,20	94,28	94,09	94,19	94,73	94,35	94,54
TnT	78,50	80,39	79,43	72,47	74,83	73,63	87,46	88,74	88,09	88,13	89,19	88,66
	Numerus											
gold	86,22	85,41	85,81	81,94	80,94	81,43	94,06	93,90	93,98	94,60	94,33	94,47
TnT	82,41	82,59	82,50	76,66	76,99	76,82	89,81	90,42	90,12	90,10	90,73	90,42
	Numerus + Person											
gold	86,25	85,52	85,88	81,95	80,93	81,44	94,09	93,94	94,01	94,48	94,17	94,33
TnT	82,58	82,79	82,68	76,59	76,90	76,74	89,73	90,34	90,03	90,11	90,64	90,37
	Verb-Merkmale											
gold	86,46	85,42	85,94	81,98	80,81	81,39	94,23	94,07	94,15	94,69	94,41	94,55
TnT	82,98	82,74	82,86	77,32	77,08	77,20	90,25	90,57	**90,41**	90,71	91,04	**90,88**

Tabelle 8: Die Ergebnisse des Berkeley Parsers auf verschiedenen morphologischen Varianten.

Baumbank können die morphologischen Merkmale keine nützlichen Informationen fürs Konstituenzparsing zur Verfügung stellen.

6 Diskussion

POS-Tags bzw. morphologische Merkmale bilden eine Schnittstelle zwischen der lexikalischen Ebene in einer Baumbank und den eigentlichen syntaktischen Bäumen. Wie schon eingangs erwähnt, kann ein einzelnes POS-Tag als eine Äquivalenzklasse von Wörtern gesehen werden (z.B. über alle finiten Verben mit einer bestimmten Kombination von Person und Numerus oder über alle Verben insgesamt). Ein Parser konstruiert nun den eigentlichen Syntax-Baum, ausgehend von den POS-Tags, nicht direkt auf den Wörtern. Die Haupterkenntnis, die wir aus unseren Experimenten mitnehmen, ist, dass der Erfolg einer Kombination von POS-Tagging und Parsing durch ein Zusammenspiel verschiedener Faktoren bestimmt wird. Faktoren sind die verwendeten Ansätze zur Kombination bzw. zur Disambiguierung von einzelnen Tags, die Charakteristika der Syntax-Bäume und die Granularität des POS-Tagsets.

Wir haben uns bei unserer Arbeit auf den „Pipeline“-Ansatz beschränkt, bei dem die Ausgabe eines POS-Taggers als Eingabe des Parsers dient. Wie in anderen Arbeiten (siehe Abschnitt 2) scheint auch bei uns die Qualität der Parserausgabe stark von der Qualität der Tag-Eingabe abzuhängen. Je komplexer das Tagset, desto mehr Informationen können dem Parser zur Verfügung gestellt werden. Allerdings sind nicht alle Informationen hilfreich. Und je komplexer das Tagset ist, desto schlechter können die Tagger zwischen einzelnen Tags für ein Wort disambiguieren. Fehler beim Tagging führen dann beim Parsen zu qualitativ schlechten Bäumen. Die „richtige“ Granularität des POS-Tagsets hängt wiederum vom syntaktischen Annotationsschema ab. Während für die eher flach annotierte TiGer-Baumbank das Universal Tagset am besten abschneidet, so liegt bei der TüBa-D/Z das STTS vorne.

Es ist zu erwarten, dass sich ein anderes Bild ergäbe, würde man mit grammatischen Funktionen parsen und auswerten. In diesem Falle könnte der Parser von dem Parallelismus zwischen Kasus und grammatischer Funktion von Konstituenten profitieren. Für gute Ergebnisse sollte hier ein feineres POS-Tagset mit morphologischer Information effektiver sein. Allerdings muss diese morphologische Annotation mit einem hohen Korrektheitsgrad annotiert werden. Ansonsten leidet die Qualität des Parsers. Daher sollte erwogen werden, die morphologische Disambiguierung nicht durch einen POS-Tagger sondern durch eine spezialisierte morphologische Komponente, wie z.B. *Morfette* (Chrupala et al., 2008), vorzunehmen.

Ein für das Parsing ideales POS-Tagset hätte die Eigenschaft, dass es die einzelnen Wörter in Äquivalenzklassen einteilt, abhängig vom Potential einer Äquivalenzklasse, bei der Unterscheidung von Parse(unter)bäumen zu helfen. Diese Äquivalenzklassen müssen nicht notwendigerweise mit linguistisch motivierten Äquivalenzklassen zusammenfallen – ideal wäre ein Maß, mit dem dieser Informationsgehalt eines einzelnen Tags, bzw. einer Wort-Äquivalenzklasse, ausgedrückt werden könnte. Abhängig von der betrachteten Einzelsprache kommen verschiedene Techniken für die automatische Ausbildung solcher

Äquivalenzklassen durch Clustering in Betracht. Sollte man auf die Reproduktion eines bestimmten anderen POS-Tagsets angewiesen sein, so können diese Äquivalenzklassen auch nach dem Parsing durch die Tags aus dem entsprechenden POS-Tagset ersetzt werden.

Abgesehen davon scheint der „Pipeline"-Ansatz nicht unbedingt die ideale Kombination der beiden Aufgaben von Tagging und Parsing zu sein. Im „Pipeline"-Ansatz werden die POS-Tags bereits *vor* dem Parsing komplett disambiguiert. Dies hat den Vorteil, dass die Aufgabe des Parsers vereinfacht wird, was i.a. zu einer höheren Parsing-Geschwindigkeit führt. Allerdings ist der Parser an diese Entscheidungen gebunden, auch wenn sich herausstellen sollte, dass die POS-Sequenz nicht dem syntaktischen Modell des Parsers entspricht. Da bereits mehrere erfolgreiche kombinierte Modelle für gleichzeitige Tag-Disambiguierung und Dependenzparsing (oder kürzlich auch fürs Konstituenzparsing) vorgestellt wurden, sollten derartige Ansätze in Zukunft weiter verfolgt werden. Hierbei muss, wie schon vorher erwähnt, ein Maß gefunden werden, wie viel Information bzw. welcher Ambiguitätsgrad in den POS-Tags beim Parsing zu optimalen Ergebnissen führt. Des Weiteren muss abgeklärt werden, bis zu welcher morphologischen Granularität ein kombiniertes Modell erfolgreich arbeiten kann. In jedem Fall bleibt in kombinierten Modellen eine weitere Schwierigkeit bestehen. Für eine ideale Modellierung der Einzelkomponenten können verschiedene Modelle erforderlich sein, die schwierig zu kombinieren sind. So ist u.U. die Finite-State-Technologie am besten für eine morphologische Komponente geeignet, Markov-Modelle für das Part-of-Speech-Tagging und ein bestimmter Grammatikformalismus für das eigentliche Parsing.

7 Schlussbetrachtung

In unserer Arbeit haben wir den Einfluss von Part-of-Speech-Tagsets auf Parsingergebnisse untersucht. Dies geschah unter Benutzung des Stuttgart-Tübingen-Tagsets, morphologischer Varianten desselben, und des Universal Tagsets. Unsere Experimente haben wir in einem „Pipeline"-Ansatz durchgeführt, bei dem die Ausgabe eines Taggers als Eingabe des Parsers fungiert. Es kamen mehrere Tagger zum Einsatz, basierend auf einem Markov-Modell, auf einem Maximum-Entropy-Modell, auf Conditional Random Fields, und auf Support Vector Machines. Als Parser wurde der Berkeley Parser verwendet.

Unsere Ergebnisse zeigen, dass es deutliche Unterschiede in Bezug auf die Qualität der POS-Tagger gibt und dass die Auswahl des POS-Taggers auch von der Granularität des Tagsets abhängig gemacht werden muss. Für TiGer besteht die beste Kombination aus dem Universal Tagset und SVMTool, während für die TüBa-D/Z die Kombination aus STTS und TnT bessere Ergebnisse erbringt. Morphologische Information erweist sich beim Konstituentenparsing als wenig hilfreich, selbst wenn diese Information vollständig korrekt ist. Weitere Forschung ist nötig, um eine geeignete Repräsentation der morphologischen Information zu finden, die es dem Parser erlaubt, sie gewinnbringend einzusetzen.

Literatur

Apidianaki, M., Dagan, I., Foster, J., Marton, Y., Seddah, D. und Tsarfaty, R. (Hgg.) (2012). *Proceedings of the ACL 2012 Joint Workshop on Statistical Parsing and Semantic Processing of Morphologically Rich Languages*. Association for Computational Linguistics, Jeju, Republik Korea.

Bangalore, S. und Joshi, A. (1999). Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–265.

Bohnet, B. (2010). Very high accuracy and fast dependency parsing is not a contradiction. In: *Proceedings of the 23rd International Conference on Computational Linguistics (IJCNLP)*, Seiten 89–97, Peking, China.

Bohnet, B. und Nivre, J. (2012). A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In: *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Seiten 1455–1465, Jeju, Republik Korea.

Boyd, A. (2007). Discontinuity revisited: An improved conversion to context-free representations. In: *Proceedings of The Linguistic Annotation Workshop (LAW) at ACL 2007*, Seiten 41–44, Prag, Tschechische Republik.

Brants, S., Dipper, S., Hansen, S., Lezius, W. und Smith, G. (2002). The TIGER treebank. In: *Proceedings of the First Workshop on Treebanks and Linguistic Theories (TLT)*, Seiten 24–41, Sozopol, Bulgarien.

Brants, T. (1998). *TnT–A Statistical Part-of-Speech Tagger*. Universität des Saarlandes, Computational Linguistics, Saarbrücken, Deutschland.

Brants, T. (2000). TnT–a statistical part-of-speech tagger. In: *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics and the 6th Conference on Applied Natural Language Processing (ANLP/NAACL)*, Seiten 224–231, Seattle, WA, USA.

Briscoe, T. und Carroll, J. (2002). Robust accurate statistical annotation of general text. In: *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*, Seiten 1499–1504, Las Palmas, Spanien.

Candito, M. und Seddah, D. (2010). Parsing word clusters. In: *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, Seiten 76–84, Los Angeles, CA, USA.

Charniak, E., Carroll, G., Adcock, J., Cassandra, A., Gotoh, Y., Katz, J., Littman, M. und McCann, J. (1996). Taggers for parsers. *Artificial Intelligence*, 85(1-2):45–57.

Chen, X. und Kit, C. (2011). Improving part-of-speech tagging for context-free parsing. In: *Proceedings of 5th International Joint Conference on Natural Language Processing (IJCNLP)*, Seiten 1260–1268, Chiang Mai, Thailand.

Chrupala, G., Dinu, G. und van Genabith, J. (2008). Learning morphology with Morfette. In: *Proceedings the Fifth International Conference on Language Resources and Evaluation (LREC)*, Marrakesch, Marokko.

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. Doktorarbeit, University of Pennsylvania, Philadelphia, PA, USA.

Csendes, D., Csirik, J., Gyimóthy, T. und Kocsor, A. (2005). The Szeged Treebank. In: Matoušek, V., Mautner, P. und Pavelka, T. (Hgg.), *Text, Speech and Dialogue: Proceedings of TSD 2005*, Seiten 123–131. Springer.

Curran, J. R., Clark, S. und Vadas, D. (2006). Multi-tagging for lexicalized-grammar parsing. In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL)*, Seiten 697–704, Sydney, Australien.

Dalrymple, M. (2006). How much can part-of-speech tagging help parsing? *Natural Language Engineering*, 12(4):373–389.

Daum, M., Foth, K. und Menzel, W. (2003). Constraint based integration of deep and shallow parsing techniques. In: *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Budapest, Ungarn.

Farkas, R. und Schmid, H. (2012). Forest reranking through subtree ranking. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL)*, Seiten 1038–1047, Jeju, Republik Korea.

Foth, K., Daum, M. und Menzel, W. (2005). Parsing unrestricted German text with defeasible constraints. In: Christiansen, H., Skadhauge, P. R. und Villadsen, J. (Hgg.), *Constraint Solving and Language Processing*, Seiten 140–157. Springer.

Giménez, J. und Màrquez, L. (2004). SVMTool: A general POS tagger generator based on Support Vector Machines. In: *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, Seiten 43–46, Lissabon, Portugal.

Goldberg, Y. und Tsarfaty, R. (2008). A single generative model for joint morphological segmentation and syntactic parsing. In: *Proceedings of The 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL:HLT)*, Seiten 371–379, Columbus, OH, USA.

Hajič, J., Böhmová, A., Hajičová, E. und Vidová-Hladká, B. (2000). The Prague Dependency Treebank: A three-level annotation scenario. In: Abeillé, A. (Hg.), *Treebanks: Building and Using Parsed Corpora*, Seiten 103–127. Kluwer, Amsterdam.

Hatori, J., Matsuzaki, T., Miyao, Y. und Tsujii, J. (2011). Incremental joint POS tagging and dependency parsing in Chinese. In: *Proceedings of 5th International Joint Conference on Natural Language Processing (IJCNLP)*, Seiten 1216–1224, Chiang Mai, Thailand.

Hinton, G. (1999). Products of experts. In: *Proceedings of the Ninth International Conference on Artificial Neural Networks*, Seiten 1–6, Stockholm, Schweden.

Höhle, T. (1986). Der Begriff "Mittelfeld", Anmerkungen über die Theorie der topologischen Felder. In: *Akten des Siebten Internationalen Germanistenkongresses 1985*, Seiten 329–340, Göttingen, Deutschland.

Joachims, T. (1999). Making large-scale SVM learning practical. In: Schölkopf, B., Burges, C. und Smola, A. (Hgg.), *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

Koo, T., Carreras, X. und Collins, M. (2008). Simple semi-supervised dependency parsing. In: *Proceedings of The 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL:HLT)*, Seiten 595–603, Columbus, OH, USA.

Kübler, S., Hinrichs, E. W. und Maier, W. (2006). Is it really that difficult to parse German? In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Seiten 111–119, Sydney, Australien.

Kübler, S., Maier, W., Rehbein, I. und Versley, Y. (2008). How to compare treebanks. In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*, Seiten 2322–2329, Marrakesch, Marokko.

Kübler, S. und Penn, G. (Hgg.) (2008). *Proceedings of the Workshop on Parsing German at ACL-08*. Association for Computational Linguistics, Columbus, OH, USA.

Lafferty, J. D., McCallum, A. und Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, Seiten 282–289, San Francisco, CA, USA.

Lakeland, C. (2005). *Lexical Approaches to Backoff in Statistical Parsing*. Doktorarbeit, University of Otago, Neuseeland.

Le Roux, J., Sagot, B. und Seddah, D. (2012). Statistical parsing of Spanish and data driven lemmatization. In: *Proceedings of the ACL 2012 Joint Workshop on Statistical Parsing and Semantic Processing of Morphologically Rich Languages*, Seiten 55–61, Jeju, Republik Korea.

Maier, W. (2006). Annotation schemes and their influence on parsing results. In: *Proceedings of the COLING/ACL 2006 Student Research Workshop*, Seiten 19–24, Sydney, Australien.

Maier, W., Kaeshammer, M. und Kallmeyer, L. (2012). Data-driven PLCFRS parsing revisited: Restricting the fan-out to two. In: *Proceedings of the Eleventh International Conference on Tree Adjoining Grammars and Related Formalisms (TAG+11)*, Paris, Frankreich.

Marcus, M. P., Santorini, B. und Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330. Special Issue on Using Large Corpora: II.

McCallum, A. (2002). *MALLET: A Machine Learning for Language Toolkit*. `http://mallet.cs.umass.edu`.

Petrov, S., Barrett, L., Thibaux, R. und Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL)*, Seiten 433–440, Sydney, Australien.

Petrov, S., Das, D. und McDonald, R. (2012). A universal part-of-speech tagset. In: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Türkei.

Prins, R. und van Noord, G. (2003). Reinforcing parser preferences through tagging. *Traitement Automatique des Langues, Special Issue on Evolutions in Parsing*, 44(3):121–139.

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In: *Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP)*, Seiten 133–142, Philadelphia, PA, USA.

Rehbein, I. und van Genabith, J. (2007). Treebank annotation schemes and parser evaluation for German. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Seiten 630–639, Prag, Tschechische Republik.

Riezler, S., King, T. H., Kaplan, R. M., Crouch, R., Maxwell, J. T. und Johnson, M. (2002). Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Seiten 271–278, Philadephia, PA, USA.

Schiller, A., Teufel, S., Stöckert, C. und Thielen, C. (1995). Vorläufige Guidelines für das Tagging deutscher Textcorpora mit STTS. Technischer Bericht, Universität Stuttgart, Institut für maschinelle Sprachverarbeitung, und Seminar für Sprachwissenschaft, Universität Tübingen.

Seddah, D., Kübler, S. und Tsarfaty, R. (Hgg.) (2010). *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*. Association for Computational Linguistics, Los Angeles, CA, USA.

Seddah, D., Tsarfaty, R. und Foster, J. (Hgg.) (2011). *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*. Association for Computational Linguistics, Dublin, Irland.

Seeker, W. und Kuhn, J. (2013). Morphological and syntactic case in statistical dependency parsing. *Computational Linguistics*, 39(1):23–55.

Skut, W., Krenn, B., Brants, T. und Uszkoreit, H. (1997). An annotation scheme for free word order languages. In: *Proceedings of the 5th Applied Natural Language Processing Conference (ANLP)*, Seiten 88–95, Washington, DC, USA.

Telljohann, H., Hinrichs, E. W., Kübler, S., Zinsmeister, H. und Beck, K. (2012). *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)*. Seminar für Sprachwissenschaft, Universität Tübingen, Deutschland.

Thielen, C. und Schiller, A. (1994). Ein kleines und erweitertes Tagset fürs Deutsche. In: Feldweg, H. und Hinrichs, E. (Hgg.), *Lexikon & Text*, Seiten 215–226. Niemeyer, Tübingen.

Toutanova, K., Klein, D., Manning, C. und Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In: *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, Seiten 252–259, Edmonton, Kanada.

Toutanova, K. und Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC)*, Hong Kong.

Versley, Y. (2005). Parser evaluation across text types. In: *Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*, Barcelona, Spanien.

Versley, Y. und Rehbein, I. (2009). Scalable discriminative parsing for German. In: *Proceedings of the 11th International Conference on Parsing Technologies (IWPT)*, Seiten 134–137, Paris, Frankreich.

Watson, R. (2006). Part-of-speech tagging models for parsing. In: *Proceedings of the 9th Annual CLUK Colloquium*, Milton Keynes, UK.

Xia, F. (2000). The part-of-speech tagging guidelines for the Penn Chinese Treebank (3.0). IRCS Technical Report IRCS-00-07, University of Pennsylvania, Philadelphia, PA, USA.

Yoshida, K., Tsuruoka, Y., Miyao, Y. und Tsujii, J. (2007). Ambiguous part-of-speech tagging for improving accuracy and domain portability of syntactic parsers. In: *IJCAI'07: Proceedings of the 20th International Joint Conference on Artifical Intelligence*, Seiten 1783–1788, Hyderabad, Indien.

Anhang: Das STTS-Tagset

ADJA	attributives Adjektiv
ADJD	adverbiales oder prädikatives Adjektiv
ADV	Adverb
APPR	Präposition; Zirkumposition links
APPRART	Präposition mit Artikel
APPO	Postposition
APZR	Zirkumposition rechts
ART	Bestimmter oder unbestimmter Artikel
CARD	Kardinalzahl
FM	Fremdsprachliches Material
ITJ	Interjektion
KOUI	Unterordnende Konjunktion mit zu und Infinitiv
KOUS	Unterordnende Konjunktion mit Satz
KON	Nebenordnende Konjunktion
KOKOM	Vergleichspartikel, ohne Satz
NN	Normales Nomen
NE	Eigennamen
PDS	Substituierendes Demonstrativpronomen
PDAT	Attribuierendes Demonstrativpronomen
PIS	Substituierendes Indefinitpronomen
PIAT	Attribuierendes Indefinitpronomen
PPER	Ersetzbares Personalpronomen
PPOSS	Substituierendes Possessivpronomen
PPOSAT	Attribuierendes Possessivpronomen
PRELS	Substituierendes Relativpronomen
PRELAT	Attribuierendes Relativpronomen
PRF	Reflexives Personalpronomen
PWS	Substituierendes Interrogativpronomen
PWAT	Attribuierendes Interrogativpronomen

Tabelle 9: Die 54 Tags des STTS (Teil 1).

PWAV	Adverbiales Interrogativ- oder Relativpronomen
PROAV/PAV	Pronominaladverb
PTKZU	`zu` vor Infinitiv
PTKNEG	Negationspartikel
PTKVZ	Abgetrennter Verbzusatz
PTKANT	Antwortpartikel
PTKA	Partikel bei Adjektiv oder Adverb
TRUNC	Kompositions-Erstglied
VVFIN	Finites Verb, voll
VVIMP	Imperativ, voll
VVINF	Infinitiv, voll
VVIZU	Infinitiv mit `zu`, voll
VVPP	Partizip Perfekt, voll
VAFIN	Finites Verb, aux.
VAIMP	Imperativ, aux.
VAINF	Infinitiv, aux.
VAPP	Partizip Perfekt, aux.
VMFIN	Finites Verb, modal
VMINF	Infinitiv, modal
VMPP	Partizip Perfekt, modal
XY	Nichtwort, Sonderzeichen
$,	Komma
$.	Satzbeendende Interpunktion
$(	Sonstige Satzzeichen; satzintern
NNE	Kombination aus Nomen und Eigenname

Tabelle 10: Die 54 Tags des STTS (Teil 2).