Inhaltsverzeichnis

0 Einleitung

In der Computerlinguistik ist seit einiger Zeit die Tendenz hin zu einer eher empirisch orientierten Arbeitsweise zu beobachten. Die in der natürlichen Sprachverarbeitung über lange Zeit entwickelten “Spielzeuggrammatiken” zeigten zwar, wie spezielle Phänomene eines eingeschränkten Anwendungsbereichs maschinell erfaßt werden können, für wirkliche Anwendungen jedoch sind diese Systeme aufgrund ihrer geringen Abdeckung kaum geeignet. Anstatt isolierte Einzelfälle auf sehr hohem Niveau zu analysieren, sollte das erste Ziel sein, alle Phänomene auf einem notwendig niedrigeren Niveau, aber dafür vollständig zu bearbeiten. Das heißt, zuerst einmal ist die observationelle Adäquatheit im Sinne von Chomsky zu gewährleisten.

Die Erfahrungen haben gezeigt, daß nur durch intensive empirische Arbeiten ein weiterer Fortschritt erzielt werden kann. Und so gewinnt die Verifizierung und Evaluierung linguistischer Theorien anhand umfangreicher Textkorpora eine immer größere Bedeutung. Hinzu kommt die (trügerische?) Hoffnung, durch geeignete Methoden der Korpusbearbeitung, neue linguistische Erkenntnisse gewinnen zu können. Als Folge dieser Tendenz (begünstigt durch die in zunehmendem Maße in maschinenlesbarer Form verfügbaren Textquellen) nahm die Aufbereitung umfangreicher maschinenlesbarer Korpora und deren maschinelle Auswertung zu. Als erstes umfangreiches maschinenlesbares Korpus wurde bereits Anfang der 80er Jahre das Brown-Korpus (Francis & Kucera 1982) erstellt und unter linguistischen Gesichtspunkten ausgewertet. In den letzten Jahren gab es im englischsprachigen Raum mehrere Projekte, deren Ziel die Annotation großer maschinenlesbarer Korpora mit linguistischer Information war und ist. Als Beispiel seien hier nur das SUSANNA-Corpus, ein mit syntaktischen Kategorien annotierter Ausschnitt des Brown-Korpus, (Garside/Leech/Sampson 1987) und die Penn-Treebank (Liberman 1991) genannt. Im Rahmen dieser Entwicklung gewinnt auch die maschinelle Lexikographie und die Computer-Morphologie zunehmend an Bedeutung, was sich unter anderem daran zeigt, daß in Deutschland die ersten Morpholympics (ein Wettbewerb, bei dem computerlinguistische Programme zur morphologischen Analyse und Synthese gegeneinander antraten) ausgetragen wurden.[1]

Eine wichtige Grundlage für die vollständige maschinelle Bearbeitung von beliebigen Texten, beispielsweise für Anwendungen in den Bereichen Wortklassentagging, Text-Alignment, Information Retrieval, automatische Indizierung, u.v.a., ist die morphologisch-lexikalische Analyse der Texte, um die es in dieser Arbeit gehen wird. Dazu gehört einerseits als Datengrundlage das Lexikon und andererseits ein Analyseverfahren, die Lemmatisierung.

Unter dem Begriff *Lemma* versteht man in der traditionellen Lexikographie schon seit jeher das Stichwort oder auch den Eintrag im Wörterbuch. Die Menge der in einem Wörterbuch verzeichneten Einheiten stellt den Lemmabestand dar. Dementsprechend

1. Die Ergebnisse und Anforderungen sind in LDV-Forum 11/1, 1994 veröffentlicht.

bedeutet *Lemmatisierung* in der traditionellen Lexikographie die Zuordnung eines Lemmas (oder auch Stichworts) zu einer (oder auch einer Menge von) Belegform(en). Der Begriff der *Lemmatisierung* ist in der klassischen Lexikographie also untrennbar mit dem Wörterbuch verknüpft. In der maschinellen Sprachverarbeitung versteht man unter *(automatischer) Lemmatisierung* die (automatische) Rückführung einer Wortform auf eine kanonische Grundform zusammen mit einer Annotierung von bestimmter linguistisch relevanter Information über die entsprechende Wortform. Die Grundform zusammen mit den weiteren Informationen bildet dann das *Lemma*. Die Rolle des Wörterbuches wurde - nicht zuletzt mangels verfügbarer umfangreicher maschinenlesbarer Wörterbücher - bei der automatischen Lemmatisierung in den Hintergrund gedrängt. Stattdessen wird in den meisten Systemen mit morphologischen Regeln und Heuristiken versucht, Hypothesen über die Grundform, die Wortklasse und die morphosyntaktischen Merkmale aufzustellen. Diese Hypothesen werden dann teilweise mit Hilfe eines relativ kleinen Lexikons zu verifizieren versucht. Es liegt auf der Hand, daß diese Vernachlässigung der lexikalischen Komponente bei der automatischen Lemmatisierung zu zahlreichen Problemen und Fehlanalysen führt. Schon im Abschlußbericht über das Lemmatisierungssystem SALEM wird die Bedeutung des Lexikons für die Lemmatisierung klar erkannt: “Gelingen und Nichtgelingen hängt eben zum großen Teil von dem Umfang des Wörterbuches und der Korrektheit der Einträge ab.” (Eggers 1980, S.13-14).

Es soll in dieser Arbeit gezeigt werden, daß der enge Zusammenhang zwischen Wörterbuch und Lemmatisierung auch im Rahmen der automatischen Sprachverarbeitung bewahrt werden muß und eine wichtige Grundvoraussetzung der automatischen Lemmatisierung ein vollständiges elektronisches Wörterbuch ist, das in dieser Form fürs Deutsche bislang noch nicht existierte. Ein vollständiges und korrektes elektronisches Wörterbuch findet darüberhinaus in nahezu allen Bereichen der Sprachverarbeitung Einsatz: als lexikalische Basis für natürlichsprachliche Systeme im Bereich der maschinellen Übersetzung, der natürlichsprachlichen Datenbankabfrage, der Erkennung gesprochener Sprache, bei der Rechtschreibkorrektur usw. Aus diesem Grund wurde das am CIS entwickelte Wörterbuchsystem (CISLEX) anwendungsunabhängig und theorieneutral konzipiert.

Die ausführliche Beschreibung elektronischer Wörterbücher stellt den ersten Teil dieser Arbeit dar. Es wird in diesem Kapitel zuerst um Aufgaben und Eigenschaften elektronischer Wörterbücher und den Stand der maschinellen Lexikographie in Deutschland gehen. Dann folgt eine Beschreibung des CISLEX-Wörterbuchsystems, bei der die Auswahlkriterien und die morphologische Kodierung der einfachen Wortformen des Deutschen im Vordergrund stehen werden. Abgeschlossen wird die Beschreibung des Wörterbuchs mit einigen Untersuchungen zum Formenbestand, die dann interessante Rückschlüsse auf Probleme bei der Lemmatisierung erlauben.

Das zweite Kapitel ist der Lemmatisierung gewidmet. Nach einem Überblick über verschiedene Lemmatisierungstechniken und einer Klärung des Lemmabegriffs wird dann das eigentliche Lemmatisierungsverfahren beschrieben. Es werden in diesem Kapitel die Schritte, die zur vollständigen Lemmatisierung von Texten notwendig

sind, anhand von zahlreichen Beispielen erläutert. Es geht dabei um den gesamten Lemmatisierungsprozeß von der korrekten Erfassung der relevanten Einheiten aus nicht aufbereitetem Fließtext (Tokenisierung) über die Wortformenanalyse bis hin zur Disambiguierung. Das Problem der Wortformenanalyse von Simplizia ist bei diesem Ansatz, der auf einem Vollformenlexikon aufbaut, zum größten Teil bereits im Lexikon gelöst. So kann in dieser Arbeit verstärkt auf Sonder- und Mischformen, Formen also, die außer Buchstaben noch weitere Zeichen enthalten, eingegangen werden, die in bisherigen System weitgehend zugunsten der morphologischen Analyse der Wortformen vernachlässigt wurden. Eine korrekte Identifikation dieser Formen ist jedoch für die meisten Anwendungen ebenso wichtig. Im Anschluß an die Untersuchung der zu behandelnden Fälle werden einige Ergebnisse, die die Lemmatisierung eines Testkorpus liefert, diskutiert. Das dritte Kapitel schließlich enthält eine Spezifikation des Lemmatisierungsprogramms mit seinen wichtigsten Funktionen. Ausführliches Beispielmaterial ist als Anhang beigefügt.

1 Elektronische Wörterbücher

In diesem Kapitel soll das Lexikon und die Problematik der Erstellung und Konzeption elektronischer Wörterbücher weitgehend unabhängig von der Lemmatisierung dargestellt werden. In diesem Zusammenhang werden zuerst die Anforderungen, denen ein elektronisches Wörterbuch genügen muß, formuliert. Danach wird eine Bestandsaufnahme der momentan in maschinenlesbarer Form verfügbaren deutschen Wörterbücher zeigen, daß diese Anforderungen bislang im deutschsprachigen Raum noch nicht erfüllt werden. Mit dem französischen DELA-System wird ein Modell für elektronische Wörterbücher vorgestellt, das den zuvor formulierten Anforderungen weitgehend entspricht.

1.1 Eigenschaften elektronischer Wörterbücher

Ziel dieses Abschnittes wird es sein, die Unterschiede, die zwischen herkömmlichen Wörterbüchern und elektronischen Wörterbüchern bestehen, herauszuarbeiten. Ausgehend von Eigenschaften bzw. Defekten herkömmlicher Wörterbücher soll gezeigt werden, was an elektronischen Wörterbüchern neu ist.

Die Eigenschaften und Mängel herkömmlicher Wörterbücher ergeben sich aus dem intendierten Einsatz. Traditionelle Lexika sind als Nachschlagewerke für einen menschlichen Benutzer konzipiert. Das bedeutet, daß ein gewisses Maß an Grundwissen beim Benutzer vorausgesetzt werden kann, so daß bestimmte Informationen nicht kodiert werden müssen. Andere Informationen brauchen nur implizit kodiert zu werden, da der Benutzer sie erschließen kann. Auch bezüglich der Darstellung und Kodierung unterschiedlicher Information kann mit dem Vorwissen des Benutzers gerechnet werden, so daß keine eindeutige Markierung verschiedener Typen von Information notwendig ist. Dies ist auch unter dem Gesichtspunkt der Übersichtlichkeit nicht unbedingt wünschenswert. Es kann also ganz allgemein gesagt werden, daß es bei traditionellen Lexika darum geht, einen sinnvollen Ausgleich zwischen dem Platzbedarf, dem Zeitaufwand, den ein Benutzer benötigt um eine gesuchte Information zu erhalten, und dem angestrebten Informationsgehalt zu finden. Aus diesem Problem resultieren im wesentlichen die folgenden Mängel herkömmlicher Lexika:[1]

- Unvollständige Kodierung der Information
- Inkonsistente Kodierung
- Unvollständigkeit bzgl. der Stichwörter

Dazu kommt, daß herkömmliche Lexika, bedingt durch das Medium Buch (bzw. in neuerer Zeit auch CD), eine statische Informationsquelle darstellen.

1. Eine weitere Ursache dieser Mängel ist natürlich auch die Art der Kodierung, die bei vielen Verlagen noch völlig unsystematisch per Hand erfolgt, und, damit verbunden, eine fehlende Verifizierung und Konsistenzüberprüfung.

Elektronische Wörterbücher sollen zwar auch für einen menschlichen Benutzer abfragbar sein, im Gegensatz zu den herkömmlichen Lexika dienen sie aber in erster Linie als Datenbasis für die verschiedensten sprachverarbeitenden Programme, wie Lemmatisierer, Rechtschreibkorrekturprogramme, linguistische Analysen und vieles mehr. Elektronische Wörterbücher müssen demnach den Anforderungen, die an Datenbanken zu stellen sind, genügen:

Überschaubare Strukturierung der Daten: Für den Benutzer (menschlicher Benutzer oder Anwendungsprogramm) ist der gezielte Zugriff auf bestimmte Informationen möglich. Dazu gehört auch die Vermeidung von Redundanz.

Trennung von Datenbestand und Anwendung: Die Daten sind in einem anwendungsunabhängigen Format gespeichert, was eine größere Flexibilität sowohl bei der Organisation der Daten als auch beim Anwendungsprogramm zur Folge hat.

Datenintegrität: Die Eingabe der Daten wird auf Konsistenz überprüft, die Daten sind gegen Verlust geschützt und schließlich durch die Vergabe von Zugriffsberechtigungen gegen Mißbrauch geschützt.

Dynamischer Charakter der Daten: Die Daten können laufend aktualisiert werden und sind über einen längeren Zeitraum hinweg nutzbar.

Das heißt für elektronische Wörterbücher, daß es zum einen eine Programmumgebung geben muß, die Eingabe und Wartung der Daten nach obengenannten Bedingungen ermöglicht. Zum anderen muß die Information auf andere Art und Weise als im traditionellen Wörterbuch kodiert werden. Während in herkömmlichen Wörterbüchern enzyklopädische, pragmatische, semantische und morphosyntaktische Angaben in den einzelnen Einträgen gebündelt werden, müssen diese Informationen im elektronischen Wörterbuch strukturell unterschieden werden und eventuell in verschiedenen Tabellen kodiert werden. Information, die im herkömmlichen Wörterbuch nur implizit kodiert ist oder die beim Benutzer vorausgesetzt wird, muß in allen genannten Bereichen explizit kodiert werden. Die Kodierung im elektronischen Wörterbuch muß nach einheitlichen (wenn möglich operationalisierbaren Kriterien) erfolgen. Es darf nicht zu einer inkonsistenten ad hoc Klassifikation kommen, wie das in vielen herkömmlichen Wörterbüchern etwa bei der Angabe der Wortarten der Fall ist.

Elektronische Wörterbücher unterscheiden sich in einer wichtigen zusätzlichen Anforderung von Datenbanken einerseits und herkömmlichen Lexika andererseits, nämlich hinsichtlich der Vollständigkeit. Ein Vergleich des Lemmabestands verschiedener herkömmlicher Lexika ergibt nicht nur erhebliche Unterschiede, sondern zeigt auch die Unvollständigkeit dieser Lexika. Da elektronische Wörterbücher als Datenbasis für sprachverarbeitende Programme dienen, die große Textmengen automatisch erfassen und verarbeiten (beispielsweise Lemmatisierer oder Rechtschreibkorrektur), muß es mit Hilfe des elektronischen Wörterbuchs möglich sein, jedes Element eines Textes zu identifizieren bzw. zu verarbeiten. Diese Forderung ist in Gross (1989) wie folgt als Arbeitsmaxime formuliert:

> "Given a text and a recognition procedure of words that uses the electronic dictionary, all the simple words of the text should be recognized, that is, there should not be any failure of the dictionary look up process."
>
> Gross (1989) S. 35

Zusammenfassend lassen sich elektronische Wörterbücher durch folgende Eigenschaften charakterisieren:

1. Vollständigkeit (sowohl hinsichtlich der Lemmaauswahl als auch der kodierten Information)
2. Konsistenz und Kohärenz der kodierten Information
3. Strukturierung der Information und Kodierung in einem anwendungsunabhängigen Format.

1.2 Maschinelle Lexikographie in Deutschland

In diesem Abschnitt soll eine Bestandsaufnahme der momentan verfügbaren Lexika und Wörterbücher des Deutschen, die in maschinenlesbarer Form vorliegen, gemacht werden. Diese Lexika und Wörterbücher können unterschiedlichen Ursprungs sein und zu verschiedenen Zwecken verwendet werden. Demnach lassen sich 3 Arten von Lexika unterscheiden:

1. Lexika oder Wörterbücher, die die maschinenlesbare Form eines gedruckten Wörterbuchs darstellen und sich in Struktur und Informationsgehalt nicht von der gedruckten Version unterscheiden. Es kann sich dabei um Satzbänder handeln oder um bereits für den Benutzer aufbereitetes Material, das auf CDs oder anderen Speichermedien verfügbar ist. Lexika und Wörterbücher dieser Art werden im folgenden als *maschinenlesbare Lexika* bezeichnet. Sie finden einerseits Anwendung im Bereich der maschinellen Lexikographie, wo versucht wird, die relevante Information automatisch aus den Dateien der Satzbänder zu extrahieren. Andererseits werden diese maschinenlesbaren Lexika auf geeigneten Speichermedien auch kommerziell vertrieben und stellen für den menschlichen Benutzer ein Nachschlagewerk in elektronischer Form dar. Zu den maschinenlesbaren Lexika zählen am Rande auch noch kommerzielle “Fun-Anwendungen”, etwa zum Lösen von Kreuzworträtseln oder für Spiele wie Scrabble, aber auch Vokabeltrainer und ähnliches. Auf diese Sonderformen maschinenlesbarer Lexika wird hier nicht weiter eingegangen.
2. Lexika oder Wörterbücher, die speziell für die Anwendung in natürlichsprachlichen Systemen entwickelt wurden. Diese Lexika werden im folgenden als *NLP-Lexika* bezeichnet. Sie decken in der Regel nur einen kleinen Sprachausschnitt ab, der für die jeweilige Anwendung relevant ist. Die Informationen sind in der Regel in einem speziellen Grammatikformalismus kodiert und damit sind diese NLP-Lexika meist nur in einem speziellen System einsetzbar.
3. Elektronische Wörterbücher im eigentlichen Sinn, wie in Abschnitt 1.1 definiert, die sowohl als Nachschlagewerk für menschliche Benutzer dienen können, als auch als Basis für diverse sprachverarbeitende Programme. Nur diese Lexika sollen im weiteren unter dem Begriff *elektronisches Wörterbuch* verstanden werden.

Die Aufstellung über die verfügbaren Wörterbücher ist nach der obengenannten Unterscheidung gegliedert.

1.2.1 Maschinenlesbare Lexika

Es ist an dieser Stelle nicht möglich, auf den ständig wachsenden Markt von Lexika und Wörterbüchern auf CD-ROM bzw. auf maschinelle Lexika im Taschenrechner- oder Scheckkartenformat einzugehen, die einem breiten Publikum zur Verfügung stehen. Die interne Struktur dieser Wörterbücher ist für den Benutzer nicht zugänglich. Andererseits ist es aufgrund des Urheberrechts in der Regel sehr schwierig, Lexika und Wörterbücher vollständig in maschinenlesbarer Form zu erhalten. Da diese Lexika

aber häufig Gegenstand von Forschungsprojekten auf dem Gebiet der maschinellen Lexikographie sind, sollen einige hier vorgestellt werden.

- Kletts 2-sprachige Wörterbücher:
  Das “Standardwörterbuch” von Klett gibt es in den Sprachpaaren Englisch-Deutsch/Deutsch-Englisch, Französisch-Deutsch/Deutsch-Französisch, Italienisch-Deutsch/Deutsch-Italienisch und Spanisch-Deutsch/Deutsch-Spanisch. Zur internen lexikographischen Bearbeitung und Editierung liegen diese Wörterbücher in maschinenlesbarer Form vor. Sie sind vom Informationsgehalt her identisch mit den gedruckten Versionen und mit einer hausinternen Mark-up-Sprache formatiert. Der Umfang beläuft sich auf ca. 45.000 Einträge pro Sprachpaar, wobei jedes Lexem durch eine Nennform als Eintrag repräsentiert wird (Kollokationen und Idiome haben keinen Lemmastatus), weitere Einträge ergeben sich durch unregelmäßige Formen. Innerhalb der Einträge werden folgende Informationen kodiert: phonetische Transkription, Kongruenzmerkmale, syntaktische Information (z.B. Transitivität bei Verben), semantische Kombinationsrestriktionen, pragmatische Information; Kollokationen werden als Beispiele aufgeführt. Information über Flexion und Derivation ist nicht kodiert, die Wortklasse ist nur implizit kodiert (z.B. durch Auftreten eines Genusmerkmals bei Nomen).
- Das DUDEN Universalwörterbuch (DUW)
  Die Satzbänder des DUW wurden in Teilen dem IKP in Bonn für ein Projekt zur automatischen Analyse der Einträge zur Verfügung gestellt. Die Satzbänder wurden dort aufbereitet und als relationale Datenbank organisiert. Die IBM Deutschland hatte ebenfalls die Möglichkeit, mit Satzbändern des DUW zu arbeiten, das Projekt wurde allerdings nicht weitergeführt.
  Im DUW sind neben der Definition und Beispielen zu den Lemmata morphologische Informationen kodiert, die Rückschlüsse auf die Wortart erlauben. Des weiteren sind Idiome, die das Lemma als zentrale Einheit enthalten, verzeichnet. Die pragmatischen Angaben innerhalb der Einträge beziehen sich auf Herkunft des Worts, Sprachebene und Zugehörigkeit zu Fachsprachen. Insgesamt enthält das DUW ca. 120.000 Stichwörter (Grundformen), darunter auch zahlreiche Eigennamen.
- Wahrig: Wörterbuch der deutschen Sprache
  Im Rahmen eines Forschungsprojekts zur EDV-unterstützten Untersuchung der semantischen Struktur natürlicher Sprache wurde in den 70-er Jahren unter Leitung von Prof. Wahrig an der Universität Mainz das dtv-Wörterbuch der deutschen Sprache unter Verwendung der Satzvorlagen in eine Datenbank transformiert. Der erfaßte Wortschatz entspricht samt Kodierung dem dtv-Wörterbuch und kann im Sinne Wahrigs (Wahrig 1966) als Grundwortschatz bezeichnet werden. Der erfaßte Wortschatz beläuft sich auf 16.000 Grundformen, die zum Teil innerhalb der Einträge in verschiedene Lesarten aufgesplittet werden. Es sind folgende Informationen optional kodiert: Silbentrennung, Aussprache (bei Sonderfällen), grammatische Angaben (Wortart, Flexion, Valenz usw.), Stilebene und Zugehörigkeit zu einer

Fach- bzw. Sondersprache (evtl. mit Hinweis auf die regionale Verbreitung) und evtl. Verweise auf zugrundeliegende Wörter (z.B. bei Abkürzungen), Synonyme und orthographische Varianten.

- Kandler: Wortanalytisches Wörterbuch
  Basierend auf Mackensens “Deutsches Wörterbuch” (Mackensen 1955) wurde unter Leitung von Prof. Kandler mit Unterbrechungen von 1960 bis 1989 am Wortanalytischen Wörterbuch gearbeitet. Der Lexembestand von Mackensen (1955) wurde auf Lochkarten übertragen und in Morpheme (bei Kandler: “Sinnelemente”) zerlegt (und wiederum auf Lochkarten abgespeichert). Das Lexikon ist nach Morphemen geordnet, was bedeutet, daß jedes Wort genauso oft aufgeführt wird, wie es verschiedene Morpheme enthält. Insgesamt wurden 117.370 Wortformen (Derivationen, Komposita und Flexionsformen) segmentiert und erfaßt. Außer der Segmentierung sind folgende Informationen kodiert: Wortart, Subkategorisierung, Kongruenzmerkmale, Zugehörigkeit zu bestimmten Fachsprachen, Dialekten und Sprachräumen (analog zur Beschreibung in Mackensen).

Auf die Verfügbarkeit einer maschinenlesbaren Version von zweisprachigen Lexika von ausländischen Verlagen mit Deutsch als einer Sprachkomponente soll hier nicht weiter eingegangen werden.

1.2.2 NLP-Lexika

Lexika als Komponenten natürlichsprachlicher Systeme gibt es nahezu so viele wie natürlichsprachliche Systeme selbst. Sie alle zu beschreiben ist weder sinnvoll, noch in diesem Rahmen aus Platzgründen möglich. Das Augenmerk soll hier vielmehr auf Lexikonkomponenten liegen, die einen repräsentativen Sprachausschnitt beschreiben. Diese Anforderung erfüllen in erster Linie die Lexikonkomponenten von maschinellen Übersetzungssystemen und größeren Experten- oder Datenbankabfragesystemen. Außer dem LEX-Lexikon stammen alle hier aufgeführten Lexika aus dem Bereich der maschinellen Übersetzung.

- SYSTRAN Wörterbuch
  SYSTRAN ist ein System, das weitgehend nur auf lexikalischer und syntaktischer Basis (das einzige “semantische” Kriterium ist Lesartenunterscheidung) Fachtexte maschinell übersetzt. Das System enthält für jedes Sprachpaar ein eigenes Wörterbuch. Die verschiedenen Lexika haben jeweils die drei Teile Hauptwörterbuch, Idiomwörterbuch und Wörterbuch mit Mehrwortausdrücken. Neben einem Grundwortschatz enthalten die Lexika noch fachsprachliche Einträge aus den verschiedenen Anwendungsgebieten. Das Hauptwörterbuch enthält Stämme, sofern die Flexion regulär ist, bei irregulärer Flexion auch Vollformen. Entsprechend der Anwendung ist vor allem morphologische und syntaktische Information kodiert. Der Umfang des Wörterbuchs für ein Sprachpaar umfaßt in etwa 40.000 Einträge.
- SADAW
  Im Rahmen des SFB 100 wurde an der Universität Saarbrücken von 1970 an das Saarbrücker deutsche Analysewörterbuch entwickelt, das unter anderem in Saar-

brücken als Lexikon des Lemmatisierungssystems SALEM (Eggers, 1980) und in der Analysekomponente des maschinellen Übersetzungsprojekts SUSY eingesetzt wurde. Am IdS in Mannheim wird eine dort überarbeitete Version von SADAW im dortigen Textlemmatisierungssystem COSMAS (Belica 1994) eingesetzt. Die Firma SIEMENS in München, das DFKI in Saarbrücken und die Universität Stuttgart versuchen seit Anfang 1993 gemeinsam dieses Lexikon für den Einsatz in natürlichsprachlichen Systemen nutzbar zu machen. Das SADAW enthält ca. 140.000 Einträge, was in diesem Fall Stämme bedeutet. Allomorphe Stämme eines Lexems erhalten einen eigenen Eintrag, so daß die Zahl von 140.000 sich auf die Anzahl allomorpher Stämme bezieht. SADAW wurde ausgehend von SDW, dem Saarbrücker deutschen Wörterbuch erstellt und umfaßt damit den Grundwortschatz nach Wahrig 1966. Die Stammeinträge enthalten sowohl morphologische Information (mögliche Flexionsendungen und Derivations- und Kompositionsmöglichkeiten) als auch syntaktische Information wie Wortart, Valenz- und Subkategorisierungseigenschaften sowie weitere Distributionseigenschaften der Lexeme. Zum Teil enthalten die Einträge auch noch eine Spezifizierung nach Fachsprachen.

- METAL-Lexikon
  METAL ist ein maschinelles Übersetzungssystem, das von der SIEMENS AG München entwickelt und vertrieben wird. Das System ist für die Sprachpaare Deutsch-Englisch, Deutsch-Spanisch, Englisch-Deutsch, Französisch-Holländisch und Französisch-Englisch verfügbar, wobei das Deutsch-Englische System am besten ausgearbeitet ist. Weitere Sprachpaare sind in der Entwicklung. Das METAL-System benützt ein einsprachiges Lexikon (20.000 - 50.000 Einträge, d.h. nach Lesarten unterschiedene Grundformen) zur Analyse und ein Transferlexikon zur Übersetzung (ca. 20.000-35.000 Grundformen oder Mehrwortlexeme). Die Lexika umfassen nur Simplizia. Transparente Derivationen und Komposita werden durch die Grammatik in ihre Bestandteile analysiert. Kodiert wurden folgende Informationen: Wortart, morphologische Klasse, syntaktische und morpho-syntaktische Eigenschaften (z.B. Genus und mass/count-Unterscheidung bei Nomen, Verbrahmen, usw.), elementare semantische Merkmale und terminologische Spezifizierung. Das Lexikon ist aufgeteilt nach Funktionswörtern, allgemeinem Vokabular und technischem Vokabular und liegt in Form einer relationalen Datenbank vor.

- ARIS-Lexikon
  Dieses Lexikon ist Teil eines Generierungsprogramms fürs Deutsche (von der ARIS Software GmbH Stuttgart entwickelt), das im Rahmen eines Deutsch-Japanischen maschinellen Übersetzungssystems eingesetzt wird. Das Lexikon umfaßt 65.000 Einträge (Stämme und Mehrwortlexeme) und ist in der Kodierung speziell auf das Generierungsprogramm zugeschnitten, was sich in einer linguistisch nicht motivierten Kodierung der morphologischen Eigenschaften niederschlägt, so daß das Lexikon nicht anderweitig genutzt werden kann. Auf der syntaktischen Ebene ist die Wortklasse und der Subkategorisierungsrahmen der Lexeme kodiert.

- LEX-Lexikon
  Das LEX-System ist ein von der IBM Deutschland GmbH entwickeltes juristisches Expertensystem mit einem natürlichsprachlichen Interface fürs Deutsche. Das Lexikon ist unabhängig vom System auch als lexikalische Datenbank (SQL-Datenbank) abfragbar und in anderen natürlichsprachlichen Systemen einsetzbar (der Zugriff erfolgt entweder über SQL oder den Grammatikformalismus USL[1]). Das Lexikon umfaßt ungefähr 20.000 Einträge (nur Stämme), die nach verschiedenen morphologischen, syntaktischen und semantischen Kriterien kodiert sind. Bei der Kodierung der syntaktischen und morphologischen Eigenschaften handelt es sich um sehr breitgefächerte und in der Regel durch operationale Tests bestimmbare Merkmale, die von der Wortart des Eintrags abhängen. Auf der morphologischen Ebene werden in erster Linie die Flexionseigenschaften (mit Information über allomorphe Stämme) kodiert. Auf der syntaktischen Ebene wird neben dem Subkategorisierungsrahmen auch noch eine Fülle weiterer Distributionseigenschaften der einzelnen Einträge beschrieben. Was die Semantik anbelangt, werden elementare semantische Merkmale vergeben.

1.2.3 Elektronische Wörterbücher

Unter dem Namen “elektronisches Wörterbuch” sollen hier Lexika aufgeführt werden, die unabhängig von einem traditionellen Lexikon in Buchform und unabhängig von einer speziellen Anwendung entstanden sind, und die zumindest in Teilen den Anforderungen an ein elektronisches Wörterbuch aus 1.1 zu entsprechen versuchen.

- LIMAS-Lexikon
  Das LIMAS-Lexikon entstand im Rahmen des Forschungsprojekts LIMAS (Linguistische und maschinelle Sprachverarbeitung) von 1965 - 1976 am IKP in Bonn. Das Lexikon wurde auf der Basis eines umfangreichen Korpus, des LIMAS-Totalkorpus, das ca. 3.000.000 Wörter enthält, erstellt. Es umfaßt 130.000 Einträge (Vollformen) mit Angaben zur Flexion und zur Wortart (nach Bergenholtz/Schaeder 1977).
- LEXDAT-D
  An der Universität Münster werden unter der Leitung von Prof. Paprotté verschiedene Lexikonprojekte durchgeführt. Das in diesem Fall interessanteste, LEXDAT-D, ist die Entwicklung einer einsprachigen lexikographischen Datenbank des Deutschen. Die möglichen Anwendungen der Datenbank sind maschinelle Übersetzung, Erstellung von Konkordanzen, als Datengrundlage für weitere linguistische Arbeiten und automatische Spracherkennung. Das Lexikon enthält 70.000 Einträge (hochfrequente Wörter, Flexionsformen, Derivationen, Komposita, Kollokationen und Morpheme, die aus bestehenden Wörterbüchern manuell extrahiert wurden). Die Einträge enthalten Information über die Silbentrennung, Wortklassenzugehörigkeit, Kongruenzmerkmale und Subkategorisierung. Des weiteren ist die Kodierung von semantischer, morphologischer und phonologischer Information geplant.

1. USL (User Specification Language) ist ein von der IBM entwickelter Grammatikformalismus.

- Lexika am IdS Mannheim
  Das IdS in Mannheim hat eine Reihe von maschinenlesbaren Lexika und elektronischen Wörterbüchern erstellt und diese kommerziell nutzbar gemacht: das Wörterbuch der schweren Wörter, ein Valenzlexikon der deutschen Verben (Valenz und morphosyntaktische Merkmale von 400 Verben), rückläufige Wortliste des Deutschen, die Bonner Wortdatenbank in Form einer SESAM-Datenbank (nähere Beschreibung siehe 1.2.4 auf Seite 12) und MOLEX, ein deutsches Morphemlexikon. Neben der Bonner Wortdatenbank, die an anderer Stelle ausführlich beschrieben wird, fällt lediglich MOLEX unter die Rubrik *elektronisches Wörterbuch*. Ursprünglich wurde MOLEX als morphologisches Lexikon zu einem ATN-Parser entwickelt. Die aktuelle Version von MOLEX soll den gesamten deutschen Wortschatz abdecken und wird am IdS zur Lemmatisierung und automatischen Morphemanalyse der dortigen Korpora verwendet. MOLEX umfaßt 1.500.000 Einträge (Vollformen, Grundformen, Komposita und Derivationen)[1], die aus verschiedenen Quellen zusammengestellt wurden, unter anderem aus der Bonner Wortdatenbank, der rückläufigen deutschen Wortliste und dem DUDEN. Die Einträge sind nach Wortklasse und morphosyntaktischen Merkmalen klassifiziert. Aus MOLEX ausgegliedert sind die Zahlwörter (Zahladverbien, Ordinal- und Kardinalzahlen, Brüche und reelle Zahlen), die in einer separaten Grammatik analysiert werden.

1.2.4 Die Bonner Wortdatenbank

Die am IKP in Bonn erstellte Bonner Wortdatenbank läßt sich schwer in eine der 3 Kategorien “maschinenlesbares Lexikon”, “NLP-Lexikon” und “elektronisches Wörterbuch” einordnen, da es sich bei der Bonner Wortdatenbank um den Versuch handelt, bestehende Lexika von unterschiedlichem Typ in einer großen Wortdatenbank zu integrieren. Die Bonner Wortdatenbank basiert auf 12 verschiedenen Quellen: die maschinenlesbaren Lexika stammen unter anderem von Mackensen, Wahrig und Klett; an NLP-Lexika wurden unter anderem das Lexikon des Dialogsystems CONDOR ([CONDOR] 1975) sowie das bereits beschriebene SADAW verwendet. MOLEX und das LIMAS-Lexikon sind elektronische Wörterbücher, die in die Wortdatenbank mitaufgenommen wurden. Das ergibt insgesamt einen Umfang von 300.000 Einträgen (Grund- und Vollformen, je nach Ausgangslexikon). Die Bonner Wortdatenbank ist als relationale Datenbank am IdS in Mannheim verfügbar.

Am IKP wurde ein kumulativer Ansatz verfolgt, d. h. die in den einzelnen Lexika verzeichnete Information wurde ungefiltert in ein einheitliches Format überführt, Inkonsistenzen in der Kodierung einzelner Lemmata in verschiedenen Ausgangslexika wurden nicht bereinigt, ferner haben die verschiedenen Lemmata der Wortdatenbank einen unterschiedlichen Informationsgrad, je nachdem, welche Information im Ausgangslexikon kodiert war. Die Wortdatenbank weist folgende Struktur auf:

1. Worteintrag mit Herkunftsangabe

1. Stand 1987

2. Wortklasse
3. Morphologie
4. Syntaktische Merkmale (Distributionsmerkmale)
5. Semantik: Klassifizierung der Wortbedeutung
6. Pragmatische Angaben: spezifische Verwendung eines Wortes, Fachbereichszugehörigkeit, Häufigkeit
7. Phonologische Angaben (geplant)

Die angegebenen Felder stellen den Maximalrahmen für Einträge dar. Je nach aus den Quellen verfügbarer Information bleiben einzelne Felder unbelegt.

1.3 Formen elektronischer Wörterbücher

Nachdem im vorangegangenen Abschnitt die prinzipiellen Typen von Lexika und Wörterbüchern in maschinenlesbarer Form besprochen wurden, soll es nun in diesem Kapitel um die interne Struktur, insbesondere um die Frage nach den Beschreibungseinheiten in elektronischen Wörterbüchern gehen, d.h. um die Art und Weise, in der Lexeme repräsentiert werden können. Traditionelle Wörterbücher und damit auch maschinenlesbare Lexika im Sinne von 1.2.1 verzeichnen als Einträge die Grundformen der Lexeme (DUDEN, Wahrig) oder bei zweisprachigen Wörterbüchern zum Teil auch die Vollformen (Kletts zweisprachige Wörterbücher). NLP-Lexika sind entweder als Vollformenlexikon (SYSTRAN), als Stammlexikon (zum Teil SADAW) oder als Grundformenlexikon (LEX, METAL) organisiert. Bei den besprochenen elektronischen Wörterbüchern gibt es mit MOLEX ein morphembasiertes Wörterbuch, während es sich bei LEXDAT-D und der Bonner Wortdatendank um gemischte Formen handelt. Es soll hier untersucht werden, welche Repräsentationsform die Anforderungen, die an ein elektronisches Wörterbuch gestellt werden, am besten erfüllt.

1.3.1 Grundformenwörterbuch

Das Grundformenwörterbuch ist die kanonische Darstellungsform eines Lexikons, da traditionelle Wörterbücher in diesem Format organisiert sind. Die Grundform ist eine ausgezeichnete Form des Paradigmas, die per Konvention festgelegt ist (in der Regel bei Nomen die Nominativ-Singular-Form, bei Verben der Infinitiv und bei Adjektiven die Prädikativform). Es wird vom Benutzer erwartet, daß er ein fragliches Wort auf seine Grundform reduzieren kann. Bei mehrsprachigen Lexika kann nicht davon ausgegangen werden, daß der Benutzer ein Wort auf seine Grundform reduzieren kann, deshalb werden in diesem Fall teilweise auch repräsentative Flexionsformen mit Verweis auf die entsprechende Grundform verzeichnet. Bei elektronischen Lexika, die im Rahmen anderer Programme beispielsweise zur Rechtschreibkorrektur oder zur Lemmatisierung verwendet werden sollen, kann ebenfalls nicht davon ausgegangen werden, daß der “Benutzer” (in diesem Fall das Anwendungsprogramm) Wortformen auf ihre Grundform reduzieren kann. Wenn ein elektronisches Wörterbuch nur als Grundformenwörterbuch vorliegt, so ist eine separate Morphologiekomponente unerläßlich, die diese Lemmatisierung leistet. Die Effizienz und die Korrektheit der lexikalischen Anwendung hängt damit wesentlich von der Qualität und Effizienz der Morphologiekomponente ab. Für die Morphologiekomponente würde sich die Finite-State- oder 2-Ebenen-Morphologie (Koskenniemi 1983) anbieten. In diesem Rahmen wurden schon zahlreiche sehr effiziente Morphologiekomponenten für verschiedene Sprachen entwickelt. Ferner sind auch regelgesteuerte Lemmatisierer denkbar. Der Unterschied zwischen herkömmlichen, regelbasierten Lemmatisierern und Morphologiekomponenten, wie sie hauptsächlich in natürlichsprachlichen Systemen eingesetzt werden, besteht in der Verwendung eines umfangreichen Lexikons bei Morphologiekomponenten. Im Gegensatz dazu versuchen regelbasierte Lemmatisierer aufgrund eines relativen kleinen Lexikons die Wörter zu analysieren. Läßt sich eine Wortform nicht auf einen Lexikoneintrag zurückführen, wird mit reinen Formkriterien und Heuristiken

eine mögliche Grundform konstruiert. Ein regelbasierter Lemmatisierer wird deshalb in der Regel zu jedem Wort eines Textes auch eine Analyse liefern, im Gegensatz zu Morphologiekomponenten, die nur dann eine Analyse liefern, wenn das Wort oder seine Bestandteile auch als Lexikoneintrag existieren, bzw. die Wortform über morphologische Regeln von einem Eintrag abgeleitet werden kann. Allerdings ist von einem regelbasierten Lemmatisierer keine morphologische Korrektheit zu erwarten.

1.3.2 Stammwörterbuch

Der Begriff *Grundform* hat unter morphologischen Gesichtspunkten wenig Aussagekraft. Bei der Annahme einer bestimmten Form des Paradigmas als Grundform handelt es sich um eine relativ willkürliche Konvention. Es hat sich jedoch eingebürgert, existierende Wortformen des Paradigmas als Grundformen anzunehmen und damit den Begriff der Grundform vom Begriff des *Stamms* zu unterscheiden, da ein Stamm in der Regel nicht Bestandteil des Paradigmas ist, sondern lediglich die Basis für die Affigierung mit Flexiven und für Wortbildungsprozesse darstellt. In diesem Sinne unterscheidet sich ein Stammlexikon nur insofern von einem Grundformenlexikon, als es zu einem Lemma mehrere allomorphe Stämme geben kann.

Im Stammlexikon sind die Stämme der entsprechenden Grundformen aufgelistet zusammen mit den möglichen morphosyntaktischen Merkmalen, deren Formen mit diesen Stämmen gebildet werden können. Dieses Vorgehen bedeutet eine explizite Trennung zwischen konkatenativer und nicht-konkatenativer Morphologie. Die nichtkonkatenativen Prozesse werden nicht über eine Regelkomponente ausgeführt sondern explizit als allomorphe Stämme aufgelistet. Die Flexionsaffixe werden dann durch Regeln mit dem Stamm konkateniert. Dieses Verfahren hat gegenüber dem Grundformenlexikon den Vorteil, daß bei einem verhältnismäßig geringen Speicher-Mehraufwand die Morphologiekomponente wesentlich effizienter gestaltet werden kann. Es stellt sich jedoch sowohl beim Grundformenwörterbuch wie auch beim Stammwörterbuch das Problem der Darstellung von komplexen Wörtern, d.h. ob komplexe Wörter explizit als Grundformen bzw. die entsprechenden Stämme aufgelistet werden oder ob die komplexen Wörter über Regeln von den entsprechenden Grundelementen im Lexikon abgeleitet werden. Die traditionellen Lexika beziehen zu dieser Frage nicht eindeutig Stellung sondern verzeichnen eine willkürliche Auswahl komplexer Wörter entweder als Kopfeinträge oder als Untereinträge beim jeweiligen Basislexem. Für die anderen Fälle wird beim Benutzer die Kenntnis der entsprechenden Wortbildungsregularitäten vorausgesetzt.

1.3.3 Morphemwörterbuch

Während das Stammwörterbuch nur solche Morpheme enthält, die den Stamm eines Wortes bilden, werden in einem Morphemwörterbuch alle Morpheme als Einträge betrachtet. Die Morpheme werden unterschieden in Flexionsmorpheme, Derivationsmorpheme und freie bzw. Stamm-Morpheme, die bestimmten Wortklassen angehören. Mit Hilfe eines Morphemwörterbuch und entsprechenden Kombinationsregeln kann

eine gegebene Wortform in ihre elementaren Bestandteile, die Morpheme, zerlegt werden. Aus dieser Zerlegung wiederum kann das entsprechende Lemma abgeleitet werden. Ein Morphemwörterbuch fürs Deutsche in maschinenlesbarer Form ist das MOLEX vom IdS[1]. Aufgrund der sehr aufwendigen Segmentierung sind reine Morphemwörterbücher zur Lemmatisierung großer Textmengen kaum geeignet.

1.3.4 Vollformenwörterbuch

Im Gegensatz zu den obengenannten Wörterbüchern werden im Vollformenwörterbuch alle möglichen Formen eines Lexems aufgeführt. Dies hat den Vorteil, daß beim Look-up einer Wortform keine morphologische Analyse durchgeführt werden muß. Zur Lemmatisierung können elementare Suchalgorithmen verwendet werden. Jedes Paar (Flexionsform, Grundform) im Vollformenlexikon ist assoziiert mit weiterer Information wie Wortklassenzugehörigkeit und realisierte morphosyntaktische Merkmale.

1.3.5 Bewertung

Das Vollformenlexikon ist konzeptionell am einfachsten, der Zeitaufwand des lexical look-up läßt sich direkt berechnen, die Analyse hängt nicht von einer Morphologiekomponente ab. Trotzdem wird das Vollformenlexikon in der Regel aus folgenden Gründen abgelehnt:

- mangelnde linguistische Adäquatheit
- keine Möglichkeit zur Formulierung von Generalisierungen
- zu großer Speicherplatzbedarf
- mangelnde Effizienz beim lexical look-up in großen Lexika
- größerer Aufwand bei der Erstellung und Erweiterung gegenüber einem Grundformen- oder Stammlexikon

Das Argument der mangelnden linguistischen Adäquatheit und damit im Zusammenhang auch die mangelnde Möglichkeit zur Formulierung von Generalisierungen ist nur von oberflächlicher Bedeutung. Es beruht auf dem Mißverständnis, daß Vollformenlexikon bedeutet, daß die Formen zusammen mit ihren Merkmalen explizit eingegeben würden. Dieses Verfahren mag zwar bei kleinen Testlexika für natürlichsprachliche Systeme noch praktikabel sein, die Erstellung eines umfangreichen Vollformenlexikons auf diese Art und Weise jedoch wäre zu aufwendig. Aus diesem Grund kann man davon ausgehen, daß als Grundlage eines Vollformenlexikons ein Grundformen- oder Stammlexikon zu erstellen ist, auf dessen Basis das Vollformenlexikon erstellt wird. Damit ist klar, daß auch beim Vollformenlexikon eine Morphologiekomponente, in der

1. Die maschinelle Aufbereitung von Augst (1975) durch das IKS in Bonn erfüllt nicht die Ansprüche an ein elektronisches Wörterbuch und wird daher hier nicht weiter berücksichtigt.

die relevanten linguistischen Generalisierungen enthalten sind, definiert werden muß. Es ist also nicht das Problem, daß Regeln formuliert oder nicht formuliert werden sondern lediglich, wie und wann diese Regeln angewendet werden. Im einen Fall werden die Wortformen auf der Basis von Grundformen- oder Stammlexikon zur Laufzeit von der Morphologiekomponente analysiert, im Fall des Vollformenlexikons werden die Formen entweder einmalig oder jedesmal beim Starten des Systems durch die Morphologiekomponente aus den Grundformen bzw. Stämmen generiert. In jedem Fall hängt es von Regeln selbst und nicht von der Richtung ihrer Anwendung ab, ob es sich um eine linguistisch adäquate Beschreibung handelt oder nicht. Durch die zugrundeliegende Regelkomponente entfällt auch das Argument der aufwendigeren Wartung und Erstellung.

Die Effizienzargumente gegen ein Vollformenlexikon lassen sich ebenso wie die Adäquatheitsargumente leicht entkräften. Denn durch die Entwicklung moderner Speichermedien mit enormer Kapazität und sehr schnellen Zugriffszeiten wird sowohl das Platzproblem als auch das Problem des effizienten Zugriffs auf große Datenmengen gelöst. Durch die Anwendung effizienter Suchalgorithmen und effizienter Datenstrukturen kann die Suchzeit stark reduziert werden. Selbst bei Standardsuchverfahren wie TRIES (Gonnet 1984) ist die Komplexität der Erkennung linear in der Wortlänge. Betrachtet man demgegenüber, daß selbst ein in der Praxis sehr effizientes Verfahren wie die Two-Level-Morphologie prinzipiell NP-vollständig ist (Barton, 1987), so kann das Argument der Ineffizienz eines Vollformenlexikons kaum aufrechterhalten werden.

Während für die Lemmatisierung aus den obengenannten Gründen ein Vollformenlexikon vorzuziehen ist, ist für andere Anwendungen, wie Rechtschreibkorrektur[1] das Morphemlexikon sinnvoll. Ein wirklich multifunktionales Lexikonsystem kann damit nicht allein aus einem der obenangeführten Lexikontypen bestehen, sondern muß sowohl ein Grundformen- oder Stammlexikon sowie ein Vollformen- und Morphemlexikon umfassen, um je nach intendierter Anwendung auf die am besten geeignete Datenstruktur zugreifen zu können. Bei der Realisierung eines solchen multifunktionalen Lexikonsystems fürs Deutsche diente das französische DELA-System als Vorbild. Lexika in diesem Format existieren in unterschiedlichem Umfang bereits für die Sprachen Englisch, Italienisch, Portugiesisch, Griechisch, usw. Um diese Systeme gemeinsam für die maschinelle Übersetzung nutzen zu können, wurde beim CISLEX-Kernlexikon versucht, die Kodierung soweit wie möglich kompatibel zur Kodierung der DELA-Wörterbücher zu halten.

1. Bei der Rechtschreibkorrektur ist ein Lexikon, das möglichst kleine Einheiten verwendet besser geeignet, um die korrekten Bestandteile eines Worters zu identifzieren und auf diese Art, den Fehler zu lokalisieren. Dafür sind die Morpheme als minimale Beschreibungseinheiten vorzuziehen.

1.4 Das französische DELA-System

Das DELA-System umfaßt eine Reihe elektronischer Wörterbücher des Französischen, die am LADL (Laboratoire d'Automatique Documentaire et Linguistique der Universität Paris) unter Leitung von Prof. Maurice Gross entwickelt wurden. Der Name "DELA" steht für Dictionnaire électronique du LADL. Die DELA-Lexika stellen die Basis eines umfangreichen Forschungsprogramms zur automatischen Analyse natürlicher Sprache, das vom LADL verfolgt wird, dar. Das Hauptziel dieses seit 20 Jahren laufenden Projekts ist die vollständige Beschreibung des gesamten Vokabulars hinsichtlich Orthographie, Phonologie, Morphologie und Syntax. Die Lexika beinhalten die orthographische, phonologische und morphologische Beschreibung von Lexemen. Die syntaktische Beschreibung geschieht mittels einer sogenannten Lexikongrammatik (lexique grammaire). Die Lexikongrammatik enthält eine möglichst vollständige Beschreibung der syntaktischen Distribution der im Lexikon verzeichneten Elemente. Die Lexikongrammatik ist in Form von Tabellen organisiert, in denen jedes Lexem bezüglich einer Reihe von Distributionseigenschaften klassifiziert wird. Derartige Beschreibungen existieren bereits für folgende Bereiche: einfache Verben (Boons, Gross, Guillet, Leclère, 1976), Adverbien (Gross, 1990), Idiome (Gross, 1988) und verschiedene Stützverbkonstruktionen (Labelle, 1978; Meunier, 1981; Giry-Schneider, 1978 & 1988). Eine semantische und pragmatische Beschreibung der Lexeme ist für die Zukunft geplant.

Die DELA-Lexika können aber auch unabhängig von den lokalen Grammatiken eingesetzt werden, und zwar zur Rechtschreibkorrektur, zur automatischen Indexierung von Texten (in diesem Fall werden mit Hilfe des Lexikons die Texte lemmatisiert) und zur Bearbeitung von Korpora (auch in diesem Fall wird zuerst lemmatisiert, um dann Konkordanzen auf der Basis der Lemmata zu erstellen). Die Bearbeitung der Korpora dient sowohl der automatischen Dokumentation als auch der Vervollständigung des Lexikons selbst.

1.4.1 Die DELA-Teillexika

Das DELA-System besteht aus einem Lexikon der einfachen Formen (DELAS), dem Lexikon der komplexen Formen (DELAC) dem Lexikon der flektierten einfachen Formen (DELAF), dem phonologischen Lexikon (DELAP) und einem Lexikon der Radikale (DELAR). DELAS und DELAF werden in 1.4.2 auf Seite 19 und 1.4.3 auf Seite 24 ausführlich beschrieben. In dem phonologische Lexikon DELAP ist für alle Formen, die im DELAF kodiert sind, zusätzlich zu den morphosyntaktischen Merkmalen die phonetische Transkription gegeben. Das DELAP ist die Datenbasis für ein Programm zur Phonetisierung von Texten.

Beispiel: (N:ms steht für Nomen: maskulin-singular)

vacher,[vaSe],N:ms.

Alle komplexen Wörter, auch Mehrwortlexeme und Idiome, sind im DELAC kodiert. Komposita sind nach ihrem Bildungstyp und der Art der morphologischen Markierung klassifiziert. Die morphologischen Eigenschaften der Bestandteile der Komposita sind im DELAS kodiert, das DELAC verweist nur auf die dortigen Einträge. Mit Hilfe des DELAC werden unter anderem technische Dokumente automatisch indiziert, da technisches Vokabular häufig komplex ist. Beispiel zur Kodierung komplexer Nomen vom Typ NAN (Nomen-à-Nomen) im DELAC:

accumulateur/à/le/plomb un/N1/ms;-+

asperges/à/la/crème des/fp;--

Die Einträge geben Auskunft über die möglichen Artikel (un: indefinit, des: indefinit-plural) bei Komposita, die verschiedene Flexionsformen bilden, die morphologische Klasse des flektierenden Elements (N1), Genus und Numerus (ms: maskulin-singular, fp: feminin-plural), sowie über die Stelle, an der Flexion sichtbar wird.

Im DELAR schließlich sind alle Stämme mit den möglichen Flexions- oder Derivationsendungen aufgeführt. Dieses Lexikon dient lediglich Dokumentationszwecken und wird nicht zur maschinellen Sprachverarbeitung eingesetzt.

1.4.2 Das DELAS - F

Im DELAS sind alle einfachen Formen des Französischen beschrieben. Der Begriff *einfache Form* ist orthographisch definiert als alle Sequenzen von Buchstaben, die zwischen Separatoren stehen können. Als Separatoren werden Leerzeichen, Bindestriche, Apostrophen und andere Sonderzeichen betrachtet. Das heißt, daß auch Teile von Bindestrich-Komposita, die isoliert kein sinnvolles Lexem darstellen und kontrahierte Formen, die durch Apostroph abgetrennt werden, im DELAS aufgeführt werden müssen. Ausgeschlossen werden Eigennamen und Abkürzungen, die im Französischen durch Großschreibung gekennzeichnet sind. Die Einträge sind mit einem Kode versehen, der aus einem Kürzel für die Wortart (siehe die Definition der entsprechenden Kodes in 1.4.2.1 auf Seite 20) besteht. Im Falle von flektierenden Wortklassen wird der Wortartkode ergänzt durch eine bis zu 3-stellige Zahl, die die Zugehörigkeit zu einer morphologischen Subklasse angibt. Das Beispiel zeigt den Eintrag für *consacrant*, das sowohl als Nomen vom morphologischen Typ 32 als auch als Adjektiv vom morphologischen Typ 32 auftreten kann:

consacrant,.N32.A32

Es findet auf dieser Beschreibungsebene keine Lesartenunterscheidung statt. Homographe Wörter mit denselben morphologischen Eigenschaften werden nicht unterschieden. Haben Homographen unterschiedliche morphologische Eigenschaften, so werden innerhalb eines Eintrags alle möglichen Wortart- und morphologischen Kodes aufgelistet. Das DELAS enthält momentan ungefähr 80.000 orthographisch verschiedene Einträge.

Die Einträge wurden aus den großen gängigen französischen Wörterbüchern (Petit Robert, Petit Larousse Illustré, Larousse LEXIS, Grand Dictionnaire Encyclopédique Larousse und Grand Robert) übernommen. Wörter, die in keinem dieser Wörterbücher verzeichnet waren aber in Texten belegt sind und allem Anschein nach sinnvolle französische Wörter darstellten, wurden ebenfalls aufgenommen.

1.4.2.1 Die Wortklassen des DELAS

Im DELAS werden folgende Wortklassen durch unterschiedliche Kodes beschrieben:

Nomen: N
Adjektive: A
Verben: V
Adverben: ADV
koordinierende Konjunktionen: CNJC
subordinierende Konjunktionen: CNJS
Determinatoren: Det
Interjektionen: INTJ
Präpositionen: PREP
Pronomen: Pron
Präfixe: PRF
die Restkategorie XINC.

Wörter, die Teile festgefügter Wortgruppen sind, werden mit KAT* bezeichnet, wobei KAT eine der oben aufgeführten Kategorien ist.

Die Klassen der Adverben, Determinatoren und Pronomen sind nach syntaktischer bzw. semantischer Funktion weiter subklassifiziert. Für jede flektierende Wortart ist das zugehörige Paradigma durch ein sogenanntes Flexionsmodell beschrieben. Das Flexionsmodell beschreibt alle zum Paradigma gehörenden Elemente mit den realisierbaren morphosyntaktischen Merkmalen. Die Beschreibungen der morphologischen Subklassen müssen für jedes Element des Flexionsmodells angeben, auf welche Art und Weise die morphosyntaktischen Merkmale realisiert werden. Die Flexionsmodelle sind zur Unterscheidung von den Beschreibungen der konkreten morphologischen Subklassen durch das Kategorienmerkmal und die Pseudoklasse 00 bezeichnet.

1.4.2.2 Nomen und Adjektive

Nomen und Adjektive besitzen dasselbe Flexionsmodell. Bei beiden Wortarten können die Merkmale Genus und Numerus mit den Werten **f** für *feminin* und **m** für *maskulin* sowie **s** für *singular* und **p** für *plural* markiert werden.

Modell für Nomen und Adjektive:

N/A00 = :ms,:fs,:mp,:fp,

Die verschiedenen morphologischen Typen unterscheiden sich durch die unterschiedliche Füllung dieser vier Slots. Mögliche Füllungen dieser Slots sind Flexionsendungen und möglicherweise notwendige Operationen oder bei irregulären Formen die vollständigen Formen. Bei Nomen, die nur in einem Genus vorkommen können, bleiben die entsprechenden anderen Slots ungefüllt. Das bedeutet, daß in diesem Rahmen auch Derivationsprozesse, die Genuswechsel bewirken, beschrieben werden. Ebenso werden Nomen, die unterschiedliche Formen für beide Genera haben, unter einem Eintrag repräsentiert. Daneben gibt es für Sonderfälle wie Plurale oder Singulare Tantum folgende Modelle:

N/A00S = :ms,:fs,-,-,

N/A00MS = :ms, -, -, -,

N/A00FS = -,:fs, -, -,

N/A00P = -, -,:mp,:fp,

N/A00MP = -, -,:mp, -,

N/A00FP = -, -, -,:fp,

N/A00M = :ms, -,:mp, -,

N/A00F = -,:fs, -,:fp,

N/A00MF = :ms:fs,:ms:fs,:mp:fp,:mp:fp (maskulin oder feminin)

1.4.2.3 Verben

Das Modell für Verben enthält Slots für sämtliche finiten und infiniten Verbformen. In der Modellbeschreibung sind die Formen nach den verschiedenen Tempus/Modus-Kombinationen aufgeführt.[1] Die morphosyntaktischen Merkmale sind durch folgende Kürzel kodiert:

INF = Infinitiv, **VANT** = Partizip-Präsens (-ant Form des Verbs) die Merkmale der finiten Formen sind eine Folge aus Modusmerkmal-Tempusmerkmal-Person-Numerus. Das Modusmerkmal ist **I** für Indikativ, **C** für Konditional oder **s** für Konjunktiv (subjonctif), das Tempusmerkmal ist **PR** für Präsens, **IM** für Imperfekt, **PA** für passé simple und **FU** für Futur, die Person ist durch **1**, **2** oder **3** markiert und Numerus durch **s** für Singular und **p** für Plural.

1. inf/pr ist der Infinitiv-Präsens, part/pr das Partizip Präsens, part/pa die Partizip-Perfekt Formen, ind/pr die Indikativ-Präsens Formen, ind/imp die Indikativ-Imperfekt Formen, ind/pass/r die Indikativ Formen des passé simple, ind/fut/s die Indikativ-Futur Formen, subj/pr die Konjunktiv-Präsens Formen, subj/imp die Konjunktiv-Imperfekt Formen, impérat die Imperativformen und cond/pr sind die Konditional-Präsens Formen.

Modell für Verben:

V00 =inf/pr(:INF)
part/pr(:VANT)
part/pa(:PPms,:PPfs,:PPmp,:PPfp)
ind/pr(:IPR1s,:IPR2s,:IPR3s,:IPR1p,:IPR2p,:IPR3p)
ind/imp(:IIM1s,:IIM2s,:IIM3s,:IIM1p,:IIM2p,:IIM3p)
ind/pass/r(:IPA1s,:IPA2s,:IPA3s,:IPA1p,:IPA2p,:IPA3p)
ind/fut/s(:IFU1s,:IFU2s,:IFU3s,:IFU1p,:IFU2p,:IFU3p)
subj/pr(:SPR1s,:SPR2s,:SPR3s,:SPR1p,:SPR2p,:SPR3p)
subj/imp(:SIM1s,:SIM2s,:SIM3s,:SIM1p,:SIM2p,:SIM3p)
imperat(:IMP1s,:IMP2s,:IMP3s,:IMP1p,:IMP2p,:IMP3p)
cond/pr(:CPR1s,:CPR2s,:CPR3s,:CPR1p,:CPR2p,:CPR3p)

Daneben gibt es noch Muster für unpersönliche Verben und andere defektive Paradigmen.

1.4.2.4 Determinatoren und Pronomen

Die Determinatoren und Pronomen sind gemäß “La syntaxe du nom” von Maurice Gross (1986) klassifiziert. Im Rahmen dieser beiden Sammelklassen werden zahlreiche Subklassen nach syntaktischen oder morphologischen Kriterien unterschieden. Determinatoren und unpersönliche Pronomen können Genus- und Numerusmarkierung tragen und werden durch dasselbe Modell wie Adjektive und Nomen beschrieben:

Dét/Pron = :ms,:fs,:mp,:fp

Personalpronomen folgen aufgrund der Personenmarkierung einem anderen Modell (die Ziffern 1, 2, 3 geben zusätzlich zu Genus und Numerus die entsprechende Person an):

Pron = :1s,:2s,:3ms,:3fs,:1p,:2p,:3mp,:3fp

Die Possessiva, obwohl sie eine Personalmarkierung haben, werden gemäß dem Schema für Determinatoren kodiert, wobei unterschiedliche Personenmarkierung durch verschiedene Einträge repräsentiert werden und nicht als mögliche Ausprägung eines Eintrags.

1.4.2.4.1 Determinatoren

Bei den Determinatoren werden folgende Subklassen unterschieden:

- Dind: indefiniter Artikel (un, de)
- Ddéf: definiter Artikel (la, le)
- Dadj: adjektivische Determinatoren (quelque, tout, différent, ...)
- Dadv: adverbiale Determinatoren (assez, autant, trop, ...)
- Dnom: nominale Determinatoren (combien, moins, quantité, ...)
- Dnum: Numerale
- Ddém: Demonstrativa (ce)
- Poss: Possessiva (mon, son, ...)
- Préd: Prädeterminatoren (uniquement, même, plutôt, surtout, ...)
- Prépdét: Präposition mit Determinator (au, aux, du, des)

Die Possessiva bilden aus morphologischer Sicht eine Ausnahme, insofern sie neben den Genus- und Numerus-Merkmalen auch das Personenmerkmal realisieren. Um bei den Determinatoren verschiedene Modelle zu vermeiden, stellen die Possessiva der einzelnen Personen wie bereits erwähnt unterschiedliche Einträge dar. Für die veränderlichen Determinatoren gilt das Modell:

Dxxx00 = :ms,:fs,:mp,:fp

Zu dieser Gruppe gehören Dind, Ddéf, Dadj, Dnom zum Teil, Dnum, Ddém, Préd, Poss und Prépdét. Die Klasse Dadv ist wie ein Teil von Dnom unveränderlich.

1.4.2.4.2 Pronomen

Die Pronomen sind wie die Determinatoren nach Morphologie, syntaktischer und semantischer Funktion weiter unterschieden. Im Rahmen des DELAS werden folgende Teilklassen der Pronomen kodiert:

- Ppv: präverbale Pronomen (en, y, Personalpronomen, Reflexivpronomen)
- Ppov: postverbale Pronomen (le, Personalpronomen)
- Pdém: Demonstrativpronomen (ce, celui, icelui)
- Pposs: Possessivpronomen
- PronQ: Relativ- und Interrogativpronomen (dont, que, qui, quoi)
- Pron: andere Pronomen (soi, quel, lequel, personne, quiconque, cézigue)
- PrépPron: Präposition mit Pronomen (auquel, duquel, laquelle)

Das Personalpronomen wird nicht als ein Lexem betrachtet; die unterschiedlichen Kasus (im Französischen die Verwendung als Subjekt, direktes oder indirektes Objekt) dienen ebenso wie die unterschiedliche Stellung in bezug aufs Verb einer weiteren Differenzierung.

In bezug auf die Morphologie bilden die Pronomen eine inhomogene Gruppe: die Personalpronomen und Possessivpronomen können die Merkmale Person, Genus und Numerus realisieren, die Demonstrativpronomen und Präpositionen mit Pronomen realisieren nur Genus und Numerus und ein Teil der sonstigen Pronomen, die Pronomen *y* und *en*, die Interrogativ- und Reflexivpronomen schließlich sind völlig invariant.

Modell für Pronomen mit Personenmerkmal:

Ppv/Ppov/Pron/Pposs00 =p3(:3ms,:3fs,:3mp,:3fp)
p1(:1ms,:1fs,:1mp,:1fp)
p2(:2ms,:2fs,:2mp,:2fp)

Zusammenfassend läßt sich also feststellen, daß die Klassifikation der Determinatoren und Pronomen nicht in erster Linie auf morphologischen Eigenschaften beruht sondern auf syntaktisch-semantischen Kriterien. Dies ergibt jedoch in bezug auf die Morphologie zum Teil wenig konsistente Klassen.

1.4.2.5 Unveränderliche Wortklassen

Bei den nicht-flektierenden Wortklassen sind im DELAS die Standardklassen kodiert:

- ADV für Adverbien
- INTJ für Interjektionen
- PREP für Präpositionen
- CNJC für koordinierende Konjunktionen
- CNJS für subordinierende Konjunktionen
- PFX für Präfixe
- XINC für nicht-kategorisierbare Wörter

Die Kategorie PFX ist für solche Elemente gedacht, die regelmäßig als Präfix zu anderen Wörtern vorkommen und in der Regel durch Bindestriche abgetrennt sind, deshalb also im DELAS als Eintrag behandelt werden müssen. Im Gegensatz zu diesen regelmäßig auftretenden Präfixen werden Teile von Bindestrichkomposita, die nur in speziellen Zusammensetzungen auftreten, wie etwa “tohu” in “tohu-bohu” als nicht-kategorisierbar, d.h. XINC, klassifiziert.

1.4.3 Das DELAF - F

Das DELAF beinhaltet alle flektierten bzw. konjugierten Formen von Einträgen des DELAS zusammen mit der Kodierung der realisierten morpho-syntaktischen Merkmale. Das DELAF ist damit die Datenbasis für die Lemmatisierungs- und Rechtschreibkorrekturprogramme. Die Formen werden automatisch durch ein Programm

generiert, das aus den Beschreibungen der morphologischen Klassen und den Einträgen des DELAS die Vollformen generiert. Die Beschreibung der morphologischen Klassen beinhaltet eine Spezifikation der allomorphen Stämme (nur bei unregelmäßigen Verben relevant) und der jeweiligen Endungen. Die allomorphen Stämme sind explizit angegeben, sie werden nicht durch Regeln generiert.

Ein Eintrag des DELAF besteht entweder aus dem Wort allein (bei nicht-flektierenden Wortarten) oder aus einer Folge aus der flektierten Form, der zugehörigen Grundform zusammen mit ihrem Kode und einer Liste der verschiedenen Möglichkeiten von realisierten morpho-syntaktischen Merkmalen. Das folgende Beispiel zeigt den Eintrag für *maison* (Haus) im DELAF (Nfs steht für Nomen-feminin-singular, Nfp für Nomen-feminin-plural):

%maison,maison.N21:Nfs

%maisons,maison.N21:Nfp

Insgesamt ergeben sich auf der Basis der 80.000 Einträge des DELAS 700.000 flektierte Formen im DELAF, was einer Dateigröße von 10 Megabyte entspricht. Durch Kompression der Daten, in diesem Fall durch Transformation des Lexikons in die Form eines endlichen Automaten, kann der Umfang auf 1 Megabyte reduziert werden. Mit dieser komprimierten Form des DELAF können auf PS/2-Systemen ca. 100.000 Wortformen pro Minute nachgeschlagen werden.

2 Das CISLEX

Bei dem CISLEX-Projekt handelt es sich um ein Lexikonprojekt, dessen Ziel darin besteht, den Wortbestand des Deutschen im Rahmen eines elektronischen Wörterbuchsystems systematisch zu erfassen. Das Ziel ist es, mit Hilfe des Lexikonsystems jede Wortform (d.h., Buchstabenfolge, die in Texten zwischen Separatoren auftritt) mit Hilfe des Lexikons zu identifizieren. Nach einem Überblick über die verschiedenen Module des Wörterbuchsystems wird das deutsche Kernlexikon mit seiner morphologischen Kodierung ausführlich dargestellt. Eine wichtige Rolle in der Beschreibung des CISLEX spielt auch die Frage, was die relevanten Beschreibungseinheiten und die angemessenen morphologischen und morphosyntaktischen Kategorien für ein derartiges Wörterbuch sind. Ein weiterer Schwerpunkt ist die Darstellung der im Flexionsbereich auftretenden morphologischen Prozesse und deren Implementierung im Rahmen einer Morphologiekomponente.

Das CISLEX unterscheidet folgende Typen von Wortformen:

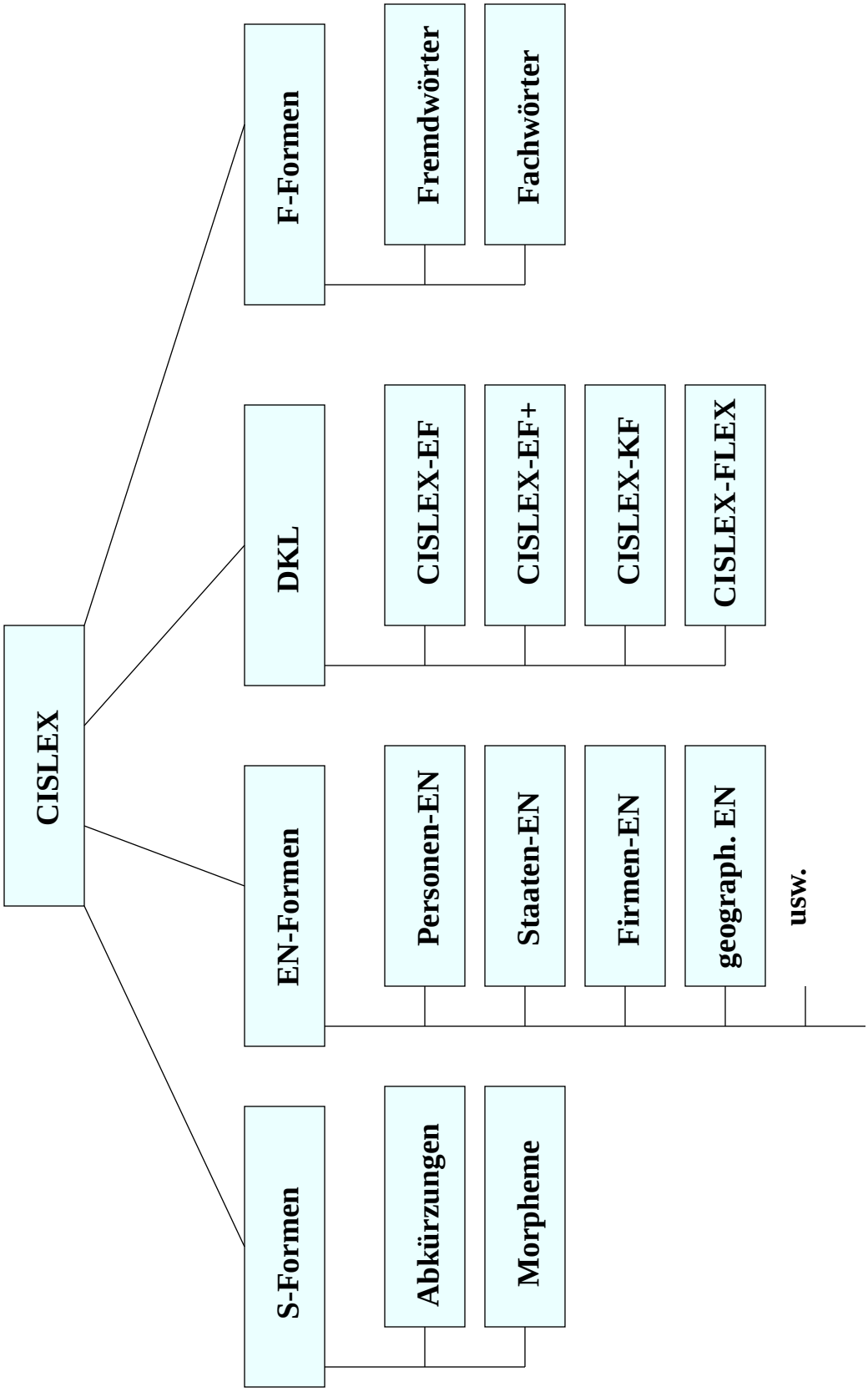
1. Einfache oder komplexe Wortformen: das sind im wesentlichen die Inhalts- und Funktionswörter[1]
2. Eigennamen aus den verschiedensten Bereichen (Vor- und Nachnamen, geographische Bezeichnungen und deren Ableitungen, Firmennamen usw.)
3. Fremd- und Fachwörter: Als Fremdwörter werden fremdsprachliche Wörter betrachtet, die sich nicht in das in Abschnitt 2.3.4 auf Seite 45 beschriebene morphologische Klassifikationsschema integrieren lassen. Als Fachwörter werden Wörter aus bestimmten Fachbereichen betrachtet, die nicht zum allgemeinen deutschen Wortschatz, wie er etwa in den großen deutschen Wörterbüchern (DUDEN, Wahrig, Mackensen usw.) verzeichnet ist, gehören.
4. Kurz- und Sonderformen wie Abkürzungen und Akronyme

Entsprechend diesen verschiedenen Typen von Wortformen ist das CISLEX in verschiedene Module gegliedert (siehe auch Abbildung 1, “Aufbau des CISLEX-Wörterbuchsystems”, auf Seite 27):

1. **das deutsche Kernlexikon (CISLEX-DKL)**, das die einfachen und komplexen Wortformen enthält und im wesentlichen das deutsche Pendant zum französichen DELA ist.
2. **das Namenslexikon (CISLEX-EN)**
3. **das Fremd- und Fachwörterbuch (CISLEX-FF)**
4. **das Lexikon der Sonderformen (CISLEX-SF)**, das neben den Kurz- und Sonderformen auch ein deutsches Morphemlexikon enthält.

1. Die exakte Definition was unter einfacher bzw. komplexer Form zu verstehen ist, folgt in Abschnitt 2.1.1 auf Seite 28ff.

Abbildung 1. Aufbau des CISLEX-Wörterbuchsystems



2.1 Das deutsche Kernlexikon DKL

Wie im Französischen DELA-System wird auch im Deutschen unterschieden zwischen einem Lexikon der einfachen Formen (DKL-EF)[1], einem Lexikon der komplexen Formen (DKL-KF) und einem Lexikon der flektierten Formen (DKL-FLEX), das auf den beiden erstgenannten basiert.

Für die Erstellung eines Grundformenlexikons im Stile des DELAS sind vorab folgende Fragen zu klären:

1. Was gehört in ein Lexikon der einfachen Formen?
2. Mit welchen syntaktischen Kategorien werden die Einträge beschrieben?
3. Mit welchen morphologischen Kategorien werden die Einträge beschrieben?

Die Frage nach den Inklusionskriterien (Frage 1) wird in diesem Abschnitt ausführlich erörtert. Es sollen dabei vor allem auch die verschiedenen Dimensionen der Lemmaauswahl diskutiert werden. In Abschnitt 2.2 auf Seite 34 werden die syntaktischen Kategorien und in Abschnitt 2.3 auf Seite 41 werden die morphologischen Kategorien des CISLEX-DKL vorgestellt. Weiterhin wird die Kodierung der einzelnen Einträge mit diesen syntaktischen und morphologischen Kategorien beschrieben, wobei auch auf Problemfälle hinsichtlich dieses Kodierschemas eingegangen wird.

2.1.1 Inklusionskriterien

Die Frage, *was gehört ins Lexikon?* stellt sich in verschiedener Hinsicht. Zunächst muß geklärt werden, welche formalen Einheiten im Lexikon repräsentiert werden sollen, ob es sich um Morpheme, Worte, Phrasen oder Sätze handeln soll. Eng damit verbunden ist die Frage, welche Kategorien ins Lexikon aufgenommen werden, ob beispielsweise Abkürzungen, Interjektionen oder Eigennamen Lemmastatus erhalten. Unabhängig von den formalen Beschreibungseinheiten muß geklärt werden, welcher Sprachbestand erfaßt werden soll, das heißt:

- Aus welchem Korpus werden die Lemmata entnommen?
- Werden Fachsprachen, und wenn ja, in welchem Umfang erfaßt?
- Welche Sprachvarietäten (Soziolekte, Idiolekte) werden beschrieben?
- Sollen dialektale Varianten erfaßt werden?
- Müssen etymologische Aspekte (veraltete Wörter oder Wortformen) berücksichtigt werden?

1. Das in Abbildung 1, “Aufbau des CISLEX-Wörterbuchsystems”, auf Seite 27 aufgeführte EF+-Lexikon unterscheidet sich nur hinsichtlich Präfigierung vom EF-Lexikon und wird hier nicht gesondert behandelt. Für die genaue Unterscheidung siehe Abschnitt 2.1.4 auf Seite 32.

Gemäß den in 1.1 aufgestellten Anforderungen an ein elektronisches Wörterbuch, insbesondere der Forderung, daß jeder Text vollständig automatisch bearbeitet werden können muß, ergibt sich für das CISLEX-System die Forderung, daß jede Einheit, die in Texten vorkommt, auch mit Hilfe des Lexikons zu identifizieren sein muß. Das CISLEX entspricht damit der Lexikondefinition von Schaeder (1981), der das Lexikon betrachtet als die Menge der sprachlichen Einheiten, “die im aktuellen Verlauf menschlicher Rede bzw. Kommunikation vorkommen”; diese Einheiten werden “in der wissenschaftlichen Betrachtung durch unterschiedliche Methoden festgestellt, aus ihrem jeweiligen Zusammenhang herausgelöst und schließlich systematisch in einem Wörterbuch dargestellt. [...] Das Lexikon enthält als Einheiten die Elemente der Sprache, die als Sprachkorpus gegeben ist.” (Schaeder 1981, S. 6).

Das DKL und insbesondere das DKL-EF ist der Kern des Lexikonsystems, indem es die einfachen Formen enthält, auf die bei der Beschreibung der Komposita und größerer lexikalischer Einheiten Bezug genommen wird, und andererseits in erster Linie den Kernwortschatz (im Bereich der einfachen Formen) abdecken soll.

2.1.2 Die Lemmaauswahl

Wie im französischen DELAS sollen auch im Deutschen wie bereits erwähnt Abkürzungen und Eigennamen nicht ins Kernlexikon aufgenommen werden, diese werden jeweils in eigenen Lexika, dem Abkürzungslexikon und dem Eigennamenlexikon, behandelt. Das Eigennamenlexikon enthält nicht nur Eigennamen selbst, sondern auch von Eigennamen derivierte Lexeme, die von ihren morphologischen und syntaktischen Eigenschaften her anderen Wortarten angehören, wie zum Beispiel adjektivische Nationalitätsbezeichnungen oder andere geographische Zugehörigkeitsbezeichnungen. Zu den Eigennamen gehören neben geographischen Bezeichnungen noch Stoffnamen, Firmennamen, Vor- und Nachnamen, Stammesbezeichnungen usw. Die Abgrenzung zwischen Eigennamen und Nomen (Appelativa) ist ein bekanntes Problem, auf das an dieser Stelle nicht näher eingegangen werden kann. Bestand und Aufbau des Eigennamenlexikons sind Gegenstand einer eigenen Arbeit.

Was Sprachvarietäten, Dialekte und etymologische Aspekte anbelangt, so wird nach dem Prinzip verfahren, daß alles, was in deutschen Texten regulär (also z.B. nicht nur in Zitatform) vorkommt oder vorkommen kann, ins Lexikon aufgenommen werden muß. Dies widerspricht natürlich dem Grundgedanken des DKL als Lexikon des Grundwortschatzes, wobei weitere Untersuchungen auf diesem Gebiet erst einmal klären müssen, was darunter genau zu verstehen ist. Hier ist jedoch geplant, im Rahmen einer Transformation des CISLEX in eine relationale Datenbank, Merkmale, die die Zugehörigkeit der Lemmata zu bestimmten Teilgebieten, die im weiteren noch näher zu spezifizieren sein werden, zu kodieren. Aufgrund dieser Merkmale wird es dann möglich sein, das CISLEX in verschiedene virtuelle Teillexika aufzuspalten, die nur die jeweils relevanten Lemmata enthalten. Eines dieser Merkmale wird dann auch die Zugehörigkeit zum Grundwortschatz beinhalten.

Neben der inhaltlichen und formalen Abgrenzung des zu erfassenden Datenbestands ist die Frage nach der Gewinnung der Daten von grundsätzlicher Bedeutung. Was diese Frage anbelangt, werden in der traditionellen Lexikographie verschiedene Wege beschritten:

1. Vergleich der Wortlisten früherer Ausgaben mit neuen Wortlisten, die durch Vergleich mit anderen Wörterbüchern, Führen von Belegkarteien und Informantenbefragung gebildet werden können (z.B. bei Wahrig und Webster).
2. Lemmaauswahl aufgrund von einer bestimmten Häufigkeit des Auftretens in einem möglichst repräsentativen Korpus, wobei Wortfelder systematisch vervollständigt werden (z.B. Trésor de la Langue Française).

Am CIS wird ein kombinierter Ansatz verfolgt. Auf der Basis von verfügbaren Wortlisten wurde ein Grundstock von Lemmata angelegt, der zum einen durch den Vergleich mit gängigen Wörterbüchern und zum anderen durch Korpusuntersuchungen ständig aktualisiert und erweitert wird. Als Korpusmaterial dienen sowohl Tageszeitungen als auch in maschinenlesbarer Form zur Verfügung stehende literarische Werke. Durch dieses kombinierte Vorgehen ist eine ständige Aktualisierung des Lexikons beispielsweise um Modeworte möglich, andererseits werden die spezifischen Probleme eines Korpuswörterbuchs, zum Beispiel das Problem der Repräsentativität eines Korpus, vermieden.

2.1.3 Der Begriff *Einfache Form*

Nachdem nun der relevante Wortschatz für die Erfassung im Lexikon definiert wurde, geht es in diesem Abschnitt darum, die *einfachen Formen* aus diesem Wortschatz zu definieren. Anders als im Französischen, wo das orthographische Kriterium der Zusammen- bzw. Getrenntschreibung ein hinreichendes Kriterium zur Unterscheidung von einfachen und komplexen Formen darstellt, gibt es im Deutschen keine Möglichkeit, von der orthographischen Form des Wortes auszugehen, da die deutschen Komposita in der Regel orthographisch nicht von den Simplizia zu unterscheiden sind.

In traditionellen Grammatiken werden die Begriffe *einfach* (bzw. Simplex) und *komplex* in der Regel nach semantischen Kriterien definiert. Ein Wort wird dann als *komplex* bezeichnet, wenn es aus mehr als einem Morphem (Flexionsmorpheme nicht mitgerechnet) besteht und sich seine Bedeutung kompositionell aus den Bedeutungen seiner Teile ergibt. Ich werde diese Art von Komplexität im folgenden als *semantisch-komplex* bezeichnen, im Unterschied zu *morphologisch-komplex*, wobei lediglich die nicht zur Flexion gehörenden morphologischen Bestandteile eines Wortes betrachtet werden. Die morphologische Komplexität ist also die Vorbedingung für semantische Komplexität. Bei den semantisch-komplexen Wörter wird häufig noch unterschieden zwischen

potentiellen Wörtern, die als Daten nicht belegt sind, jedoch theoretisch aufgrund der Morphologie und der Semantik denkbar sind,

okkasionellen Wörtern, die nicht zum Wortschatz eines Sprechers gehören sondern zu einem bestimmten Anlaß spontan neugebildet werden und

usuellen Wörtern, die zwar in ihrer Bildung noch weitgehend transparent sind, jedoch zum festen Inventar einer Vielzahl von Sprechern gehören; diese Wörter zeigen häufig schon einen semantischen Shift und damit eine starke Tendenz zur Lexikalisierung.

Aufgrund der Annahme, daß usuelle Wörter zum festen Wortschatz eines Sprechers gehören, verzeichnen traditionelle Grammatiken sowohl die usuellen semantisch-komplexen Wörter, wie auch die semantisch-einfachen Wörter im Lexikon, im Gegensatz zu den potentiellen und okkasionellen Wörtern, die im Rahmen einer Wortbildungskomponente beschrieben werden.

Da das DKL in erster Linie ein morphologisch-orientiertes Lexikon des Deutschen sein soll, scheiden derartige Kriterien von vornherein aus, ganz abgesehen davon, daß sich die semantische Komplexität ohne semantische Repräsentation *aller* Wörter nicht eindeutig (etwa durch operationale Tests) entscheiden läßt. Eine semantische und syntaktische Spezifizierung der Lexikoneinträge im Stile der Objektklassen für Nomen von G. Gross (Gross, G. 1991) bzw. der Verbklassifikation von M. Gross (Boons, Gross, Guillet, Leclère, 1976) ist zwar für die Zukunft geplant, im Moment jedoch noch nicht für den gesamten Lexembestand realisiert. Bei der Aufteilung in ein Lexikon der einfachen Formen und ein Lexikon der komplexen Formen geht es lediglich um eine möglichst effiziente und möglichst redundanzfreie Darstellung des ausgewählten Wortschatzes. So ist in diesem Zusammenhang einzig die morphologische Komplexität relevant. Das bedeutet konkret, alle komplexen Wörter, deren morphologische und morphosyntaktische Eigenschaften aufgrund des Lexikons der einfachen Formen bereits vorhersagbar sind, müssen im DKL-EF nicht aufgeführt werden, sondern kommen mit einem entsprechenden Verweis auf die Basisform ins DKL-KF.

Eine Besonderheit der deutschen Wortbildung (sowohl der Derivation als auch der Komposition) ist die Rechtsperiferität des Kopfes, das heißt, die morphologischen Eigenschaften eines Wortes sind durch dasjenige (Nicht-Flexions-)Morphem bestimmt, das am Ende des Wortes steht. Aus diesem Grund besteht in diesem Rahmen auch kein Anlaß zu einer Unterscheidung zwischen Komposition und derivationeller Präfigierung. In beiden Fällen genügt im DKL-KF ein Verweis auf die einfache Form im DKL-EF.

Es ergibt sich damit folgende Definition für den Begriff *einfache Form*:

> Ein Wort W ist eine **einfache Form** genau dann, wenn es keine sinnvolle Zerlegung $W = W_1\ W_2$ gibt, so daß W_1 eine Folge von Morphemen ist und W_2 ein Wort mit denselben morphologischen Eigenschaften wie W.

Umgekehrt ergibt sich als Definition für *komplexe Form*:

> Ein morphologisch-komplexes Wort $W = W_1 ... W_n$ ist eine **komplexe Form** genau dann, wenn $W_1,...,W_{n-1}$ Morpheme sind und W_n ein Wort mit denselben morphologischen Eigenschaften wie W ist.

Um auch bei der derivationellen Suffigierung (auch hier gilt das obengenannte Kopfprinzip der deutschen Wortbildung: das Suffix ist Träger der morphologischen Eigenschaften!) eine Redundanz bei der Kodierung zu vermeiden, werden auch die häufigsten Suffixe als spezielle Kategorien im DKL-EF mitaufgenommen. Regelmäßige Konversionen, wie zum Beispiel die Verwendung verbaler Partizipien als Adjektive oder verbaler Infinitive als Nomina werden automatisch durch Anwendung der jeweiligen Regeln flektiert.

Für den Look-up eines flektierten komplexen Wortes ergeben sich somit drei Möglichkeiten:

1. Es kann getestet werden, ob die Wortform im Lexikon der flektierten einfachen Formen ist und so auf eine einfache Form lemmatisiert werden kann. Ist dies nicht der Fall, so wird versucht von der fraglichen Wortform einen Suffix abzutrennen, der mit Hilfe des Lexikons der flektierten einfachen Formen lemmatisiert werden kann. Ist ein solcher vorhanden, so wird überprüft ob die Verbindung aus verbliebenem Anfang und Lemma des Suffixes im Lexikon der komplexen Formen zu finden ist.
2. Man greift auf ein Lexikon der flektierten komplexen Formen zurück, das aus dem Lexikon der flektierten einfachen Formen und dem Lexikon der komplexen Formen vorab generiert werden muß.
3. Die Wortform wird ohne Beachtung des Kompositalexikons mit Hilfe eines Zerlegungsalgorithmus segmentiert[1], und nur im Falle von Ambiguitäten wird das Kompositalexikon konsultiert.

In der Praxis wird ein gemischtes Verfahren, bei dem die häufigsten, lexikalisierten Komposita flektiert im Lexikon vorliegen und nur die selteneren Formen segmentiert werden müssen, am effizientesten sein.

2.1.4 Die EF+-Formen

Bei der praktischen Anwendung des DKL hat sich gezeigt, daß die oben erwähnte strikte Trennung in einfache und komplexe Formen für viele Anwendungen unnötig restriktiv ist. Auch aus linguistischer Sicht ergeben sich bei dieser Trennung Widersprüche: die Suffigierungen einfacher Basen gehören (bis auf produktive Ausnahmen, bei denen das Suffix explizit in EF kodiert wurde) wiederum zu EF, während die Präfigierungen von einfachen Basen zu KF gehören. Darüber hinaus hat sich gezeigt, daß wenn für die Definition von *einfacher Form* nicht nur flexionsmorphologische Eigen-

1. Ein solcher Algorithmus ist ohnehin notwendig, da natürlich aufgrund der Kreativität der Wortbildung unmöglich alle je auftretenden Komposita aufgelistet werden können.

schaften sondern auch die Fugenbildung beim Vorkommen in Komposita betrachtet werden, die Präfigierungen einfacher Basen sich häufig anders verhalten als die Basis. Aus diesem Grund wurde alternativ zu DKL-EF ein weiteres Lexikon DKL-EF+ erstellt, das neben allen einfachen Formen auch die Präfigierungen einfacher Basen enthält.

2.2 Die Wortarten des CISLEX-DKL

In herkömmlichen Lexika und Wörterbüchern ist die Angabe von Wortarten sehr inkonsistent. Die Grundlage ist meistens eine Klassifikation, die auf einer Mischung aus semantischen, syntaktischen und morphologischen Kriterien basiert, wobei häufig nicht einmal der Status der einzelnen Kriterien klar ist. Bei einer Wortartklassifikation für ein elektronisches Wörterbuch müssen einerseits operationale Kriterien für die Klassifikation entwickelt werden um eine konsistente Klassifizierung zu gewährleisten und andererseits muß die Klassifikation so theorieneutral wie möglich und für die intendierten Anwendungen sinnvoll sein. Damit scheidet eine semantisch motivierte Klassifikation von vornherein aus. Weiterhin sollte die Klassifikation soweit wie möglich kompatibel mit der Klassifikation der anderssprachigen DELA-Systeme sein, um für den gemeinsamen Einsatz dieser Systeme in der maschinellen Übersetzung eine Abbildung zwischen den Kategorien der einzelnen Lexika zu ermöglichen.

Da es sich beim DKL in erster Linie um ein morphologisches Lexikon handelt, erscheint eine Klassifikation nach morphologischen Kriterien, das heißt nach realisierbaren morphosyntaktischen Merkmalen und der Art der Realisation, am naheliegendsten. Demnach wären Verben charakterisiert durch die realisierbaren Merkmale *Genus verbi, Tempus, Modus,* sowie *Person* und *Numerus* des Subjekts. Nomen sind charakterisiert durch *Kasus* und *Numerus* (Genus ist bei Nomen eine inhärente Eigenschaft) und Adjektive durch *Kasus, Genus* und *Numerus* jeweils in den Ausprägungen *stark schwach* und *gemischt* sowie *Komparation.* Determinatoren, Quantoren und Pronomen würden in recht unterschiedliche Klassen zerfallen. Adverbien, Partikeln, Konjunktionen, Präpositionen und was man sonst noch zu den nichtflektierenden Funktionswörtern zählen will, wären durch rein morphologische Kriterien nicht zu unterscheiden.

Weitaus problematischer ist jedoch die Klassifikation von Lexemen, die nach gängigen Annahmen zu den flektierenden Wortklassen gehören, die für die jeweilige Wortklasse typischen morphosyntaktischen Merkmale jedoch nicht an der Oberfläche realisieren. Das sind zum Beispiel nicht flektierende Adjektive wie *lila* und *super* oder nichtflektierende Nomen wie *Abakus* und *Kasus*. Sollen Wörter diesen Typs zu den nichtflektierenden Wortarten gehören oder nimmt man für diese Fälle sogenannte 0-Morpheme an, und wie begründet man diese Annahme morphologisch? Weitere Probleme tauchen bei defektiven Paradigmen auf, ebenso wie bei Wörtern, die nur einen Teil der möglichen morphosyntaktischen Merkmale an der Oberfläche realisieren. Nach rein morphologischen Gesichtspunkten müßte man komparierbare und nicht-komparierbare Adjektive unterscheiden, Nomen, die im Singular den Genitiv markieren von solchen, die nur eine Singularform haben (z.B. alle Feminina) usw. Es lassen sich noch eine Reihe weiterer Fälle hier anführen, die belegen, daß eine rein morphologische Wortartdefinition nicht praktikabel ist. In manchen Grammatiken (z.B. in der Akademie Grammatik, Heidolph et al. 1984) werden daher zusätzlich zu den morphologischen Kriterien noch syntaktische Kriterien herangezogen. Die syntaktischen Kriterien dienen jedoch lediglich dazu, die sich aus dem morphologischen Kriterium ergebenden Klassen weiter zu differenzieren. Damit wird allerdings keine Lösung des obenge-

nannten Problems erreicht. Denn selbst bei Annahme von 0-Morphemen zur konsistenten Klassifizierung von Nomen, Adjektiven und Verben bleibt das Problem der Klassifikation von Nomen mit adjektivischer Deklination wie *Verwandter, Angestellter* oder *Beamter*, die je nach Artikelwahl unterschiedlich dekliniert werden. Eine Klassifikation, die die morphologische Klassifikation als primär betrachtet, kann diese Nomen nur als Adjektive klassifizieren.[1] Diese Klassifikation steht aber im Widerspruch zur syntaktischen Distribution dieser Elemente, die sich hinsichtlich ihrer Stellung wie Nomen verhalten. Ferner erscheint die Klassifikation eines Wortes wie *Beamter*, das im Gegensatz zu anderen Elementen diesen Typs keine feminine adjektivisch flektierende Variante kennt (die entsprechende feminine Form ist *Beamtin*) und zudem in dieser Form nicht adjektivisch verwendbar ist (die entsprechende Form ist *beamtet*), unsinnig.

Eine rein syntaktisch orientierte Klassifikation führt, wenn sie konsequent durchgeführt wird, ebensowenig wie eine rein morphologisch orientierte Klassifikation zu einem zufriedenstellenden Ergebnis. Denn als Klassifikationskriterien dienen in diesem Fall die bekannten Distributionstests, die für Verben mit unterschiedlichem Subkategorisierungsrahmen unterschiedliche Klassen liefern. Diese Tests unterscheiden auch nur attributiv verwendbare, nur prädikativ verwendbare und solche Adjektive, die in beiden Positionen vorkommen, ebenso relationale Nomen von nicht-relationalen Nomen und so weiter. Zwar sind diese Unterscheidungen alle sinnvoll, führen jedoch bei konsequenter Ausführung zu einer unüberschaubaren Anzahl von verschiedenen Wortklassen. Die Wortartklassifikation soll aber nur eine Grobeinteilung liefern, weitere Distributionseigenschaften werden sinvollerweise über verschiedene syntaktische Merkmale beschrieben.

Aus der vorangegangenen Argumentation ergibt sich für die Wortarten des DKL eine Mischklassifikation nach syntaktischen und morphologischen Kriterien[2], wobei die syntaktischen Kriterien als primär gegenüber den morphologischen Kriterien betrachtet werden. Dieses Vorgehen hat zur Folge, daß insbesondere bei den Funktionswörtern zahlreiche Mehrfachklassifikationen zu verzeichnen sind, da gerade in diesem Bereich viele Wörter an verschiedenen syntaktischen Positionen und mit unterschiedlichen syntaktischen Funktionen auftreten können. Ist für bestimmte Anwendungen eine so feine Unterscheidung nicht notwendig oder durch die damit verbundene Mehrfach-Kategorisierung zu aufwendig, können die nichtflektierenden Wortarten als eine Klasse betrachtet werden.

1. Diese Annahme ist zwar nicht völlig abwegig. In der generativen Grammatik werden Nominalphrasen mit adjektivischen Nomen häufig als Nominalphrasen mit leerem Kopf, und die adjektvischen Nomen als Adjektive analysiert. Diese Analyse widerspricht jedoch traditionellen Annahmen und ist nur im Rahmen einer Grammatiktheorie, die leere Elemente zuläßt, möglich.

2. Das Wortartensystem des DKL ist damit im Funktionswortbereich vergleichbar mit anderen vorwiegend syntaktisch orientierten Klassifikationen, wie sie etwa in Bergenholtz/Schaeder (1977) beschrieben werden.

Im DKL werden folgende Wortklassen, die im weiteren näher erläutert werden, kodiert:

Flektierende Wortklassen:

- Nomen N
- Adjektiv A
- Verb V
- Determinatoren DET
- Pronomen PRON

Nicht-flektierende Wortklassen:

- Adverb ADV
- Partikel PART
- Präposition PRAEP
- Konjunktion KONJ
- Interjektion INTJ
- Verbpartikel VPART

2.2.1 Nomen

Als Nomen werden im DKL alle lexikalischen Einheiten betrachtet, die Kopf einer Nominalphrase sein können und im Unterschied zu Pronomen zusammen mit einem Artikel auftreten können. Nomen sind bezüglich Genus und der Verwendung im Singular und Plural markiert. Nach morphologischen und syntaktischen Kriterien ergeben sich bei den Nomen folgende Subklassen:

N: reguläre Nomen mit nominaler Deklination

NA: Nomen mit adjektivischer Deklination

NQ: quantorenähnliche Nomen

Die *quantorenähnlichen Nomen* unterscheiden sich von regulären Nomen dadurch, daß sie als Maßnomen das Kopfnomen einer Nominalphrase modifizieren können. Typische Beispiele sind etwa *Pfund*, *Kilo*, usw.[1] Zu den quantorenähnlichen Nomen zählen aufgrund ihrer Distribution auch Wörter wie *bißchen*, *paar* oder *wenig* in Verwendungen wie:

(1) ein bißchen/wenig Wein bzw. ein/die paar Äpfel

1. Die quantorenähnlichen Nomen werden in der Literatur auch als Monoflexive bezeichnet, da sie in ihrer Funktion als Maßnomen nicht in Numeruskongruenz mit dem Determinator stehen müssen.

Bei den adjektivischen Nomen ist das Genus nicht wie bei den anderen Nomen ein inhärentes Merkmal. Es muß explizit kodiert werden, welche Genera bei dem Wort markiert werden können.

2.2.2 Adjektive

Adjektive sind dadurch bestimmt, daß sie pränominal (in der Nominalphrase zwischen Determinator und Kopfnomen) oder prädikativ vorkommen können. In der attributiven Verwendung kongruieren Adjektive, wenn sie flektierbar sind, hinsichtlich Kasus, Genus und Numerus mit dem Kopfnomen; die Art der Deklination ist vom vorangehenden Determinator abhängig:

(2) der saure Wein - ein saurer Wein

Ein Teil der Adjektive kann in adverbialer Funktion auftreten:

(3) ein gutes Essen - er kocht gut

Die Adjektive müssen also hinsichtlich ihrer Distribution nach den Merkmalen *attributiv, prädikativ* und *adverbial* kodiert werden. Außerdem muß die Möglichkeit der Komparation, die nicht bei allen Adjektiven gegeben ist, explizit kodiert werden.

2.2.3 Verben

Die Verben sind im Gegensatz zu den anderen Wortklassen in erster Linie morphologisch definiert: Ein Verb ist ein Wort, das ein verbales Paradigma hat. Wobei unter *verbalem Paradigma*[1] folgende Formen verstanden werden: alle finiten Verbformen, das sind alle Formen, die hinsichtlich Tempus, Modus, Person und Numerus markiert sind und die Imperativformen sowie die infiniten Formen Infinitiv, Partizip Präsens und Partizip Perfekt. Bei defektiven Paradigmen (zum Beispiel unpersönliche Verben) entscheidet dann die Distribution über die Wortklassenzugehörigkeit, d.h. wenn ein Wort zwar kein vollständiges verbales Paradigma besitzt, es aber ein Verb gibt, das bis auf die nicht realisierbaren Fälle die gleiche Distribution hat, dann wird dieses Wort als Verb klassifiziert. Wobei die Tempusmarkierung als spezielle morphologische Eigenschaft von Verben anzusehen ist. Nach der Distribution werden bei den Verben folgende Subklassen unterschieden:

Auxiliare und Modalverben: Sie können nicht alleine eine Verbalphrase bilden sondern kommen zusammen mit einem Vollverb vor

Vollverben: Sie können im Gegensatz zu den Auxiliaren und Modalverben den Kopf einer Verbalphrase bilden.

1. Der Paradigmenbegriff ist hier rein morphologisch zu sehen und bezieht sich nur auf die verschiedenen morphologischen Ausprägungen des Wortes. Analytische Realisierungen bestimmter Merkmale gehören nicht ins Paradigma.

Bei den Verben gibt es eine Reihe interessanter Distributionseigenschaften, die jedoch im Rahmen des DKL nicht alle kodiert werden können. Eigenschaften wie Reflexivität, unterschiedliche Passivtypen, Verbrahmen usw. werden im Rahmen einer gesonderten syntaktischen Verbklassifikation kodiert.

2.2.4 Determinatoren

Als Determinatoren werden im DKL Wörter bezeichnet, die an erster Stelle in einer Nominalphrase, also vor dem Adjektiv stehen, außer den pränominalen Genitivattributen. Typische Vertreter der Determinatoren sind der definite Artikel, Demonstrativa oder Possessiva. Auch ein Teil der traditionellerweise als Quantoren bezeichnete Elemente gehören zu den Determinatoren. Hier ist die Abgrenzung von Determinatoren und Adjektiven aufgrund rein distributioneller Kriterien etwas schwierig. Als charakteristische Eigenschaften der Determinatoren wird diesbezüglich eine Deklinationsforderung gegenüber Adjektiven betrachtet. Wie bereits in Abschnitt 2.2.2 auf Seite 37 beschrieben, richtet sich die Deklination eines pränominalen Adjektivs nach dem vorangehenden Determinator. Sind mehrere pränominale Adjektive vorhanden, so werden alle parallel dekliniert; eine Ausnahme ist nur bei starker Deklination im Dativ-Maskulin oder Neutrum der Fall, wo das erste Adjektiv nach dem starken Muster auf *em* endet, während alle folgenden Adjektive nach dem schwachen Muster auf *en* enden. Die Quantoren werden je nach dem Deklinationsverhalten in bezug auf folgende Adjektive und nach der Möglichkeit eines vorangehenden Determinators entweder als Adjektiv (*viele*, *wenige*, *solche*, ...) oder als Determinator (*jed_*, *manch_*, ...) klassifiziert. Die Numeralia, die bis auf *ein* unflektierbar sind, können sowohl vor, nach und auch zwischen pränominalen Adjektiven stehen und sind aus diesem Grund den Adjektiven zuzurechnen. Die folgenden Beispiele verdeutlichen diese Klassifikation:

(4) manche guten Weine / *die manchen guten Weine (Determinator)

(5) viele gute Weine / die vielen guten Weine (Adjektiv)

(6) meine letzten 3 / 3 letzten Bier (Adjektiv)

Probleme bereiten bei dieser Klassifikation die Wörter *ein* und *beide*:

(7) ein volles Glas Sekt - das eine volle Glas Sekt

(8) beide zähen Steaks - die beiden zähen Steaks

Diese Wörter verhalten sich, wenn kein anderer Determinator vorangeht, wie ein Determinator, wenn ein Determinator vorangeht, wie ein Adjektiv. Ob man dieses Verhalten zum Anlaß nimmt, eine eigene Wortart zu definieren oder sie doppelt klassifiziert ist letztendlich nicht entscheidend. Im DKL werden sie doppelt klassifiziert, um die Anzahl der Wortklassen möglichst gering zu halten.

2.2.5 Pronomen

Pronomen sind Wörter, die allein eine Nominalphrase bilden können und nicht zusammen mit Determinatoren und attributiven Adjektiven vorkommen. Zu den Pronomen gehören Personalpronomen, Relativpronomen, Reflexivpronomen, Fragepronomen usw. Neben diesen nur pronominal verwendbaren Wörtern gehen wir davon aus, daß jeder Determinator auch als Pronomen auftreten kann.

2.2.6 Präpositionen

Die Präpositionen sind dadurch gekennzeichnet, daß sie als Kopf einer Präpositionalphrase den Kasus der folgenden Nominalphrase (im Falle von Postpositionen, die hier nicht von den Präpositionen unterschieden werden, ist natürlich die vorangehende Nominalphrase zu betrachten) festlegen, in manchen Fällen auch die Präposition einer folgenden Präpositionalphrase. Im Gegensatz zu Partikeln kann die Präposition nicht ohne ihr Komplement, die Nominal- oder Präpositionalphrase, innerhalb des Satzes verschoben werden. Präpositionen sind im Gegensatz zu den Adverbien nicht satzgliedfähig[1] und können im Gegensatz zu Konjunktionen keine Satzglieder miteinander verknüpfen. An syntaktischer Information wird bei den Präpositionen der Kasus bzw. die Präposition des Komplements kodiert. Typische Präpositionen sind *auf*, *bei*, *neben*, *nach*, *zu* usw. Die Präpositionen sind aufgrund des von ihnen regierten Kasus bzw. der regierten Präposition subkategorisiert:

PREP0	keine Kasusforderung
PREP1	Akkusativ
PREP2	Genitiv
PREP3	Dativ
PREP4	Akkusativ oder Dativ
PREP5	Genitiv oder Dativ
PREP6	Genitiv oder *von*-PP mit Dativ
PREP7	Akkusativ oder PP
PREP8	*auf*-PP mit Akkusativ

2.2.7 Konjunktionen

Konjunktionen verknüpfen Konstituenten miteinander. Sie haben selbst keinen Satzgliedstatus, können also nicht allein im Vorfeld eines deutschen Satzes stehen. Sie können innerhalb des Satzes nicht verschoben werden. Die Konjunktionen werden

1. Als Kriterium für ein Satzglied gelten die Standardkriterien, wie Verschiebeprobe und Vorfeldtest.

unterschieden in subordinierende und koordinierende Konjunktionen. Typische Konjunktionen sind *aber, denn, und, weil,* usw.

2.2.8 Adverbien

Die Adverbien haben als einzige Klasse der nicht-flektierenden Wortarten Satzgliedcharakter, d.h. sie können allein das Vorfeld eines Verbzweitsatzes bilden. Sie lassen sich deshalb relativ leicht von den anderen nicht flektierbaren Wortklassen unterscheiden. Eine weitere Subklassifikation wird im Rahmen des DKL nicht vorgenommen.

2.2.9 Partikeln

Der Terminus *Partikel* wird in der Literatur unterschiedlich gebraucht. Häufig werden als Partikeln alle nicht-flektierenden Wortarten bezeichnet. Wir gehen hier von einem engeren Partikelbegriff aus, der eine Klasse von nicht-flektierenden Wörtern bezeichnet, die sich distributionell von Präpositionen, Adverbien und Konjunktionen unterscheiden. Von den Adverbien unterscheiden sich die Partikeln im wesentlichen dadurch, daß sie nicht vorfeldfähig sind. Im Gegensatz zu Konjunktionen verknüpfen sie keine Satzteile miteinander sondern sind in gewissem Rahmen innerhalb des Satzes verschiebbar, und Partikeln haben im Gegensatz zu Präpositionen keine Kasusrektion. Die Klasse der Partikeln ist also in gewissem Sinn eine Restklasse der nicht-flektierenden Wortarten. Typische Partikeln sind *also, bloß, eben, sehr, nur,* usw.

2.2.10 Interjektionen

Die Interjektionen nehmen bei den Wortarten eine Sonderstellung ein, da sie selbst Satzfunktion übernehmen können. Wenn Interjektionen innerhalb eines Satzes vorkommen, so ist das Auftreten beschränkt auf Verbzweitsätze und dort auf die Position vor dem Vorfeld. Beispiele für Interjektionen sind: *ätsch, ahoi, hatschi, frischauf, schwuppdiwupp* usw.

2.2.11 Verbpartikeln

Die Klasse der Verbpartikel enthält solche Elemente, die das Vorderglied von Partikelverben bilden und deshalb bei Verbzweitstellung frei vorkommen. Die abtrennbaren Verbpartikel müssen in der Regel nicht explizit kodiert werden, da es sich um Komposition von Verben mit Präpositionen oder anderen auch selbständig vorkommenden Elementen handelt. Lediglich Fälle wie *inne, ein,* usw., die nicht frei in einer Wortart vorkommen müssen, hier kodiert werden.

2.2.12 Restklassen

Für Wörter, die sich in das obige Schema nicht einordnen lassen, gibt es wie im französischen System auch die Restklasse XINC.

2.3 Die morphologischen Kategorien von CISLEX-EF

2.3.1 Die morphosyntaktischen Merkmale der einzelnen Wortklassen

Die flektierenden Wortklassen Nomen, Verben, Adjektive, Pronomen und Determinatoren müssen nach morphologischen Kriterien weiter subklassifiziert werden. Diese Subklassifizierung richtet sich danach, wie die Wortart-typischen morphosyntaktischen Merkmale realisiert werden. Bei der Subklassifizierung stand die Operationalisierbarkeit der morphologischen Kategorien im Vordergrund. Die Frage, was im linguistischen Sinn ein korrekter Stamm und was die entsprechenden Flexionsendungen sind, wurde vernachlässigt zugunsten einer formalorientierten Oberflächenbeschreibung des Verhältnisses zwischen den verschiedenen Flexionsformen. Die morphologischen Klassen sind einzig dadurch bestimmt, wie aus der Grundform die anderen Formen des Paradigmas gebildet werden. Die dabei relevanten Operationen werden in Abschnitt 2.4.1 auf Seite 58ff näher beschrieben. Hier soll es lediglich darum gehen, die Kriterien der morphologischen Subklassifikation und im folgenden Abschnitt die sich daraus ergebenden Klassen vorzustellen. Im Hinblick auf diese Klassifikation stellt sich natürlich sofort die Frage, welche Form eines Wortes als Grundform im CISLEX aufgenommen wird, und ob diese Grundform selbst die Basis der morphologischen Operationen ist oder ob diese Grundform nur als Nennform fungiert und die morphologischen Paradigmenbeschreibungen auf einer ganz anderen “Grundform” operieren, die aus der Nennform im CISLEX erst gebildet werden muß. Dieses Problem wird in Abschnitt 2.3.2 diskutiert.

2.3.1.1 Nomen

Bei den Nomen werden entsprechend den syntaktischen Subklassen auch jeweils unterschiedliche morphologische Subklassen gebildet. Die quantifizierenden Nomen sind nicht flektierbar und werden daher nicht weiter unterschieden. Die adjektivischen Nomen werden nach den orthographischen Regularitäten bei der Bildung der Flexionsformen analog zu den Adjektiven (siehe dort) klassifiziert. Weiterhin muß beachtet werden, daß manche adjektivischen Nomen nur in bestimmten Genera vorkommen.

Bei den regulären Nomen werden die Merkmale Kasus und Numerus realisiert, Genus ist ein inhärentes Merkmal, das gesondert kodiert wird. Im Singular kann der Genitiv und der Dativ markiert sein, außer bei femininen Nomen, die im Singular unveränderlich sind. Im Plural wird nur der Dativ explizit markiert. Es gibt also maximal 5 verschiedene Flexionsformen: Singular-Nominativ/Akkusativ, Singular-Dativ, Singular-Genitiv, Plural-Dativ und Plural-Nominativ/Genitiv/Akkusativ. Bei der Betrachtung der Paradigmen verschiedener Nomen fällt auf, daß die Singularformen sich wesentlich seltener unterscheiden als die Pluralformen. Es gibt 13 orthographisch verschiedene Möglichkeiten der Bildung der Singularformen gegenüber weit über 100 Möglichkeiten im Plural. Aus diesem Grund werden Singularmorphologie und Pluralmorphologie getrennt kodiert. Die Eigenschaft von Nomen nur pluralisch bzw. nur sin-

gularisch aufzutreten wird als spezielle morphologische Singular- bzw. Pluralklasse aufgefaßt.

Bei der Kodierung der Plurale und Singulare Tantum sind verschiedene Aspekte der Begriffe *Singularia* und *Pluralia Tantum* zu unterscheiden. Zum einen werden diese Begriffe semantisch motiviert verwendet, wie etwa im DUDEN, der ein Wort wie *Abendstern* als nur singularisch beschreibt. Dabei handelt es sich eindeutig um semantische oder pragmatische Beschränkungen. Demgegenüber gibt es Wörter, von denen aus morphologischen Gründen kein anderes Numerus möglich ist, wie etwa *Leute*, hier ist das morphologische Paradigma unvollständig, wohingegen es bei *Abendstern* eine morphologisch einwandfreie Pluralform gibt. Da es im DKL in erster Linie um eine morphologische Beschreibung der Lexeme geht, wird hier ein morphologischer Singularia bzw. Pluralia Tantum Begriff zugrunde gelegt.[1] Das bedeutet, daß wenn immer die Singular- bzw. Pluralform aus morphologischen Gründen möglich ist, so wird die entsprechende Singular- bzw. Pluralklasse kodiert. Die Entscheidung, ob eine bestimmte Numerusform möglich ist, läßt sich bei suffigierten einfachen Formen in der Regel leicht treffen, da das Suffix allein über die morphologischen Eigenschaften entscheidet. So erhalten beispielsweise alle Wörter auf *logie* dieselbe Pluralklasse, unabhängig davon, ob der Plural aus semantischen Gründen möglich ist, wie bei *Anthologie* oder ob das Wort in der Regel nur im Singular gebraucht wird, wie *Biologie*. Schwieriger ist die Entscheidung bei einmorphigen semantischen Singulare oder Plurale Tantum. Insbesondere standardsprachlich nur singularisch gebrauchte Stoffbezeichner können in den entsprechenden Fachsprachen auch pluralisch gebraucht werden, wie *Milche* als Plural von *Milch* im Molkereiwesen. Semantische Singulare und Plurale Tantum können dann in einem weiteren Kodierschritt als Defekte innerhalb des Paradigmas markiert werden.

2.3.1.2 Adjektive

Bei den Adjektiven gibt es im morphologischen Bereich zum einen die Graduierung und zum anderen die Realisierung der Kongruenzmerkmale. Graduierung ist (wenn überhaupt) in allen 3 Stellungsvarianten des Adjektivs (prädikativ, attributiv und adverbial) möglich. Die Kongruenzmerkmale Kasus, Genus und Numerus werden nur im pränominalen Fall realisiert. Die Art der Realisierung dieser Merkmale hängt davon ab, ob ein Determinator vorangeht, und wenn ja, von dessen Subkategorisierung bzgl. der Adjektivdeklination. Ist kein Determinator vorhanden, so flektiert das Adjektiv nach dem sogenannten starken Muster, nach dem definiten Artikel nach dem schwachen Muster und nach Possessiva und einigen weiteren Determinatoren gemischt. So ergibt sich zwar eine Anzahl von 3*4*3*2 = 72 verschiedenen Merkmalskombinationen. Diese werden jedoch durch nur 5 verschiedene Formen realisiert.

1. Die Unterscheidung von sematischem und morphologischem Numerus ist auch aus anderen Gründen notwendig:

   ein Apfel - zwei Äpfel - 0,5 Äpfel - 0 Äpfel

Bei *0,5/0 Äpfel* handelt es sich eindeutig um morphologischen und nicht um semantischen Plural.

Für die morphologische Subklassifikation sind die orthographischen Regularitäten bei der Bildung der Flexionsformen und der Komparationsformen relevant. Adjektivtypische orthographische Eigenschaften sind beispielsweise die e-Elision vor Dentalen (*dunkel - dunkler*) oder Umlautung bei der Graduierung (*alt - älter*). Ähnlich wie im Falle der Singulare und Plurale Tantum bei Nomen wird bei der morphologischen Klassifizierung der Adjektive von einem morphologischen Komparativ- bzw. Superlativbegriff ausgegangen. Das tatsächliche Vorkommen der Komparationsformen wird durch ein Distributionsmerkmal beschrieben. Ebenso müssen die 3 Stellungstypen durch Distributionsmerkmale beschrieben werden.

2.3.1.3 Verben

Die morphologische Klassifikation der Verben geschieht unabhängig von den syntaktischen Subklassen. Sie richtet sich danach wie die finiten und infiniten Formen gebildet werden. Die infiniten Formen des Verbparadigmas sind der Infinitiv, das Partizip Präsens und das Partizip Perfekt. Die finiten Formen beinhalten neben den Imperativformen im Singular und Plural die Tempusformen Präsens und Imperfekt in den Modi Indikativ und Konjunktiv für alle Person-Numerus-Kombinationen. Ein Verbparadigma umfaßt somit 3 infinite Verbformen und 26 finite Verbformen. Die Verben sind unterteilt in starke Verben, deren Tempusmarkierung durch Ablaut geschieht, schwache Verben, bei denen dem gesamten Paradigma derselbe Stamm zugrundeliegt und unregelmäßige Verben. Zu den unregelmäßigen Verben gehören sowohl völlig unregelmäßige Bildungen als auch Präteritopräsentia und die sogenannten Rückumlautverben. Diese 3 Typen werden nach den orthographischen Besonderheiten bei der Bildung der Formen weiter unterteilt. Bei den schwachen Verben handelt es sich lediglich um phonologisch bzw. orthographisch bedingte Veränderungen, die bei der Konkatenation mit der Flexionsendung auftreten.

Die morphologischen Klassen treten in verschiedenen Varianten auf. Die echten Präfixverben, also solche, deren Präfix nicht abtrennbar ist (*be, ent, ver, zer,...*), unterscheiden sich bei der Formenbildung von ihrem Basisverb lediglich dadurch, daß im Partizip Perfekt kein *ge* präfigiert wird. Dies wird durch ein *s* nach dem numerischen Paradigmenkode markiert. Die Verben mit abtrennbarem Präfix haben gegenüber dem Basisverb ein umfangreicheres Paradigma. Die finiten Formen umfassen sowohl die finiten Formen des Basisverbs (wenn das Verb in Verbzweitsätzen gebraucht wird, die Verbpartikel also abgetrennt ist, bzw. im Imperativ nur die Basisform) sowie alle Tempusformen des zusammengesetzten Verbs. Beim Partizip Perfekt wird *ge* nach dem Präfix infigiert (sofern das Paradigma des Basisverbs überhaupt Präfigierung von *ge* vorsieht), im Infinitiv gibt es sowohl die Form mit infigiertem *zu* als auch ohne. Aufgrund dieses Verhaltens werden die Subklassen stark, schwach und unregelmäßig weiter unterteilt in solche für Verben mit abtrennbarem Präfix und solche ohne abtrennbarem Präfix.

2.3.1.4 Determinatoren und Pronomen

Die Determinatoren und Pronomen werden aufgrund ihrer geringen Anzahl direkt als Vollformen aufgenommen. Trotzdem wurden auch sie nach den realisierbaren Merkmalen und der Art, wie diese Merkmale realisiert werden klassifiziert.

2.3.2 Das Grundformenproblem

Der Ansatz der morphologischen Klassifizierung im CISLEX ist wortbasiert, insofern die morphologischen Klassen und die Regeln von einer kanonischen Grundform ausgehen und nicht von Stämmen, wie das in morphembasierten Ansätzen der Fall ist. Hierin unterscheidet sich das morphologische System des CISLEX von dem des DELAS, bei dem der längste gemeinsame String eines Paradigmas als Eingabe für die Generierung der Vollformen benutzt wird. Der Gedanke eines Stammlexikons im linguistischen Sinn wurde aus den in Abschnitt 1.3 beschriebenen Gründen hier nicht weiter verfolgt. Die Grundform selbst hat jedoch keinerlei theoretischen Status, es handelt sich dabei lediglich um eine nach bestimmten Konventionen ausgezeichnete Form des Paradigmas. Beim CISLEX wurden die Standardkonventionen für Grundformen beibehalten, so daß sich für die einzelnen Wortarten folgende Grundformen ergeben:

Nomen: Nominativ-Singular (bei Plurale Tantum: Nominativ-Plural)

Nomen mit adjektivischer Deklination: die *e*-Form

Verben: Infinitiv (bei Partikelverben der Infinitiv ohne infigiertes *zu*)

Adjektive: die unflektierte Positivform

Pronomen und Determinatoren: die jeweils unflektierte Form bzw. die Nominativ-Singular-Form.

2.3.3 Die Kodierung

Die Kodes der flektierenden Wortklassen sind alle nach demselben Schema aufgebaut:

<Kategoriensymbol> (<Subkl.>) <Paradigmennr.> <evt. weitere Merkmale>

Als Kategoriensymbole für die flektierenden Wortklassen werden verwendet:

N	für Nomen
ADJ	für Adjektive
V	für Verben
DET	für Determinatoren
PRON	für Pronomen

Bei den Nomen sind neben den “normalen” Nomen die Subklassen **NA** für adjektivische Nomen und **NQ** für quantoren-ähnliche Nomen möglich. Bei den Verben gibt es die Subklassen **VST**, **VSTT**, V**SW**, **VSWT**, **VUNR** bzw. **VUNRT** für starke Verben,

starke Verben mit abtrennbarem Präfix, schwache Verben, schwache Verben mit abtrennbarem Präfix, unregelmäßige Verben bzw. unregelmäßige Verben mit abtrennbarem Präfix.

Die Paradigmennummer ist den Tabellen in Abschnitt 2.3.4 zu entnehmen. Zusätzlich wird bei regulären Nomen das Genus explizit durch die Werte *fem* für feminin, *mask* für maskulin und *neut* für neutrum kodiert. Ist das Genus nicht zu bestimmen, wie etwa bei Plurale Tantum (z. B. *Leute*), so wird dies durch das Merkmal *no_gen* zum Ausdruck gebracht. Beispiele für die Kodierung der einfachen Formen befinden sich in Anhang A auf Seite 198.

2.3.4 Morphologische Klassifikation

2.3.4.1 Reguläre Nomen

Die Nomen mit regulärer Flexion werden morphologisch subklassifiziert durch ein Paar aus Singulardeklinationstyp und Pluraldeklinationstyp. Die folgenden Tabellen zeigen die exakte Definition der Singular- und Pluralkodes bei Nomen. In den Spalten werden die Operationen und anzufügenden Endungen eingetragen. Es gelten bei der orthographischen Beschreibung folgende Schreibkonventionen: “ bedeutet Umlaut, * Umlaut von Doppelvokal, <n>- bedeutet, daß von der Grundform n Zeichen am Ende abgezogen werden müssen, bevor die Endung angehängt wird. Systematische Varianten innerhalb eines Paradigmas werden durch Kommas separiert, Optionalität wird durch Klammerung notiert.

Tabelle 1. Singular-Deklinationstypen der Nomen

Klasse	Nom	Gen	Dat	Akk	Beispiel
0	0	0	0	0	Frau
1	0	(e)s	(e)	0	Weg
2	0	s	0	0	Auge
3	0	en	en	en	Mensch
4	0	n	n	n	Bauer
5	0	ses	(se)	0	Ergebnis
6	0	ns	n	0	Name
7	0	ens	en	0	Herz
8	0	1-sses	1-sse,0	0	Koloß
9	0	1-ssens	0	0	Voß
10	0	es	(e)	0	Haus
11	0	(n)	(n)	(n)	Abate

Klasse	Nom	Gen	Dat	Akk	Beispiel
12	0	(es)	0	0	Sex
13	0	(s)	0	0	Shake
T	Plurale Tantum				Leute

Tabelle 2. Plural-Deklinationstypen der Nomen

Klasse	Nom	Gen	Dat	Akk	Beispiel
0	0	0	0	0	Korken
1	0	0	n	0	Arbeiter
2	e	e	en	e	Weg
3	en	en	en	en	Mensch
4	n	n	n	n	Bauer
5	nen	nen	nen	nen	Lehrerin
6	s	s	s	s	Auto
7	se	se	sen	se	Kenntnis
8	er	er	ern	er	Kind
9	x	x	x	x	Eau
10	ta	ta	ta	ta	Komma
11	"0	"0	"n	"0	Mutter
12	"e	"e	"en	"e	Kraft
13	"0	"0	"0	"0	Graben
14	"er	"er	"ern	"er	Mann
15	"en	"en	"en	"en	Statt
16	ien	ien	ien	ien	Material
17	ten	ten	ten	ten	Bau
18	1-en	1-en	1-en	1-en	Firma
19	1-sse	1-sse	1-ssen	1-sse	Koloß
20	1-ssens	1-ssens	1-ssens	1-ssens	Voß
21	"1-sse	"1-sse	"1-ssen	"1-sse	Fluß
22	"1-sser	"1-sser	"1-ssern	"1-sser	Faß
23	2-a	2-a	2-a	2-a	Lexikon
24	2-en	2-en	2-en	2-en	Museum

Tabelle 2. Plural-Deklinationstypen der Nomen

Klasse	Nom	Gen	Dat	Akk	Beispiel
25	1-tia	1-tia	1-tia	1-tia	Expektorans
26	1-zien	1-zien	1-zien	1-zien	Expektorans
27	1-te	1-te	1-ten	1-te	Klima
28	1-nten	1-nten	1-nten	1-nten	Atlas
29	2-i	2-i	2-i	2-i	Modus
30	2-ina	2-ina	2-ina	2-ina	Nomen
31	2-ora	2-ora	2-ora	2-ora	Tempus
32	2-era	2-era	2-era	2-era	Genus
33	1-i	1-i	1-i	1-i	Topos
34	2-e	2-e	2-en	2-e	Kursus
35	2-izes	2-izes	2-izes	2-izes	Index
36	2-res	2-res	2-res	2-res	Pater
37	1-den	1-den	1-den	1-den	Bronchitis
38	2-oden	2-oden	2-oden	2-oden	Tripus
39	1-zes	1-zes	1-zes	1-zes	Matrix
40	1-zen	1-zen	1-zen	1-zen	Matrix
41	1-ies	1-ies	1-ies	1-ies	Hobby
42	1-tes	1-tes	1-tes	1-tes	Dos
43	1-ssen	1-ssen	1-ssen	1-ssen	Hosteß
44	2-ier	2-ier	2-ier	2-ier	Dinosaurus
45	e	e	e	e	Clavicula
46	1-ten	1-ten	1-ten	1-ten	Charis
47	es	es	es	es	Comtess
48	2-er	2-er	2-ern	2-er	Kanonikus
49	"	"	"	"	Epiphora
50	1-ces	1-ces	1-ces	1-ces	Helix
51	1-ines	1-ines	1-ines	1-ines	Imago
52	2-een	2-een	2-een	2-een	Kaktus
53	1-enen	1-enen	1-enen	1-enen	Kilbi
54	1-gen	1-gen	1-gen	1-gen	Larynx

Tabelle 2. Plural-Deklinationstypen der Nomen

Klasse	Nom	Gen	Dat	Akk	Beispiel
55	1-ges	1-ges	1-ges	1-ges	Lex
56	2-es	2-es	2-es	2-es	Lenis
57	1-e	1-e	1-e	1-e	Lira
58	3-zi	3-zi	3-zi	3-zi	Magnifikus
59	1-sses	1-sses	1-sses	1-sses	Miß
60	in	in	in	in	Mudschahed
61	1-y	1-y	1-y	1-y	Papirossa
62	1-n	1-n	1-n	1-n	Peninsula
63	nes	nes	nes	nes	Pernio
64	2-eis	2-eis	2-eis	2-eis	Polis
65	2-ides	2-ides	2-ides	2-ides	Präses
66	2-iden	2-iden	2-iden	2-iden	Präses
67	2-zen	2-zen	2-zen	2-zen	Scherwonez
68	3-zera	3-zera	3-zera	3-zera	Ulkus
69	im	im	im	im	Cherub
70	inen	inen	inen	inen	Cherub
71	1-	1-	1-	1-	Kambio
72	2-ce	2-ce	2-ce	2-ce	Penny
73	4-uareg	4-uareg	4-uareg	4-uareg	Targi
74	2-ites	2-ites	2-ites	2-ites	Antistes
75	"1-ten	"1-ten	"1-ten	"1-ten	Fakultas
76	3-eet	3-eet	3-eet	3-eet	Foot
77	1-des	1-des	1-des	1-des	Glottis
78	1-ra	1-ra	1-ra	1-ra	Jus
80	1-ia	1-ia	1-ia	1-ia	Universale
81	1-ien	1-ien	1-ien	1-ien	Universale
83	"n	"n	"n	"n	Schade
85	i	i	i	i	Bani
86	1-ot	1-ot	1-ot	1-ot	Agora
87	"*e	"*e	"*en	"*e	Saal

Tabelle 2. Plural-Deklinationstypen der Nomen

Klasse	Nom	Gen	Dat	Akk	Beispiel
88	"*er	"*er	"*ern	"*er	Aast
89	7-iatan-tum	7-iatan-tum	7-iatan-tum	7-iatan-tum	Pluraletantum
90	1-izia	1-izia	1-izia	1-izia	Simplex
91	a	a	a	a	Synonym
92	den	den	den	den	Zubehör
93	1-enen	1-enen	1-enen	1-enen	Laui
94	"*e	"*e	"*en	"*e	Koog
95	ka	ka	ka	ka	Quinder
96	3-s	3-s	3-s	3-s	Drumlin
97	1-hi	1-hi	1-hi	1-hi	Largo
98	ar	ar	ar	ar	Kenning
99	sen	sen	sen	sen	Kirmes
100	1-jim	1-jim	1-jim	1-jim	Goi
101	1-ta	1-ta	1-ta	1-ta	Hepar

2.3.4.2 Adjektive

Die Adjektive werden nach den orthographischen Besonderheiten, die bei der Bildung der Deklinationsformen auftreten, subklassifiziert. Im Gegensatz zu den Nomen gibt es bei den Adjektiven keine prinzipiell verschiedenen allomorphen Realisierungen der Flexions- oder Graduierungsmerkmale. Es handelt sich, die Umlautung bei der Graduierung ausgenommen, lediglich um phonologisch bzw. orthographisch bedingte Unterschiede. Die 18 verschiedenen Klassen werden gemäß folgender Tabelle gebildet:

Tabelle 3. Adjektivklassen

	Prädikativ	Deklination	Komparation	Superlativ	Beispiel
0		ohne Besonderheiten			schön
1		e-Ausfall	e-Ausfall		dunkel
2		(e-Ausfall)	(e-Ausfall)		bescheiden
3		ß -> ss	ß -> ss	ß -> ss,-est	kraß

Tabelle 3. Adjektivklassen

	Prädikativ	**Deklination**	**Komparation**	**Superlativ**	**Beispiel**
4			Umlaut	Umlaut	jung
5			(Umlaut)	(Umlaut)	grob
6			Umlaut	Umlaut, -est	alt
7			Umlaut	Umlaut, -t	groß
8			(Umlaut)	(Umlaut), -est	gesund
9		ß -> ss	ß -> ss, (Umlaut)	ß -> ss, (Umlaut),-est	naß
10				-(e)st	blau
11				-est	heiß
12	(e)			-(e)st	blöd(e)
13	(e)			-est	leis(e)
14	(e)	keine Besonderheiten			mürb(e)
15	e			-est	weise
16	(e)			(Umlaut)	bang(e)
17	(e)			Umlaut	lang(e)

2.3.4.3 Verben

Die morphologische Klassifikation der Verben erfolgt im wesentlichen in Anlehnung an die Arbeit von Müller (1991). Die schwachen Verben unterscheiden sich wie die Adjektive nur durch phonologisch oder orthographisch bedingte Besonderheiten bei der Verkettung mit den entsprechenden Flexiven. Es werden folgende Phänomene unterschieden:

- *e*-Einschub vor *st*: *atmest, arbeitest* (eevst)
- *e*-Einschub vor *t*: *atmet, arbeitet* (eevt)
- *s*-Ausfall nach Sibilanten (*s,z,x,ß*): *du mixt, tanzt, rast, schweißt* (sanS)
- *e*-Ausfall nach *r*: wir klettern (eanr)
- *e*-Ausfall vor *r* (optional, immer zusammen mit *e*-Ausfall nach *r*): *klettre - klettere* (eavr)
- *e*-Ausfall nach *l*: *wir sammeln* (eanl)
- *e*-Ausfall vor *l* (teilweise optional): *ich sammle* (eavl)
- *e*-Ausfall nach *e* (optional): *kie(e)n* (eane)

- Imperativ-*e*-Einschub (teils optional, teils obligatorisch): *samm(e)le, sammel* (imee)
- Wechsel von *ss* und *ß*: *stressen - streßt* (ss-ß)

Diese Besonderheiten führen zu folgenden Klassen:

Tabelle 4. Morphologische Klassifikation der schwachen Verben

Merkmale	**Flexiv**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Infinitiv	en				eanr	eanr	eanl	eane	
1.Sing Präs Ind	e				eavr		eavl	eane	
2.Sing Präs Ind	st		eevst	sanS					sanS, ss-ß
3.Sing Präs Ind	t		eevt						ss-ß
1.Plu Präs Ind	en				eanr	eanr	eanl	eane	
2.Plu Präs Ind	t		eevt						ss-ß
3.Plu Präs Ind	en				eanr	eanr	eanl	eane	
1.Sing Präs Konj	e				eavr		eavl	eane	
2.Sing Präs Konj	est				eanr	eanr	eanl	eane	
3.Sing Präs Konj	e				eavr		eavl	eane	
1.Plu Präs Konj	en				eanr	eanr	eanl	eane	
2.Plu Präs Konj	et				eanr	eanr	eanl	eane	
3.Plu Präs Konj	en				eanr	eanr	eanl	eane	
1.Sing Prät Ind	te		eevt						ss-ß
2.Sing Prät Ind	test		eevt						ss-ß
3.Sing Prät Ind	te		eevt						ss-ß
1.Plu Prät Ind	ten		eevt						ss-ß
2.Plu Prät Ind	tet		eevt						ss-ß
3.Plu Prät Ind	ten		eevt						ss-ß
1.Sing Prät Konj	te		eevt						ss-ß
2.Sing Prät Konj	test		eevt						ss-ß
3.Sing Prät Konj	te		eevt						ss-ß
1.Plu Prät Konj	ten		eevt						ss-ß
2.Plu Prät Konj	tet		eevt						ss-ß

Tabelle 4. Morphologische Klassifikation der schwachen Verben

3.Plu Prät Konj	ten		eevt						ss-ß
Part Präs	end				eanr	eanr	eanl	eane	
Part Perfekt	t		eevt						ss-ß
Imperativ Sing		imee	imee	imee	imee eavr	imee	imee eavl	imee	imee
Imperativ Plu	t		eevt						ss-ß

Die Klassen lassen sich durch folgende Eigenschaften der Verben charakterisieren:

Klasse 1: Verben ohne Besonderheiten.
Klasse 2: Verben deren Infinitiv endet auf: nasale Doppelkonsonanz (außer *lm, ln, rm, rn*) + *en*, d.h. A*m+en* und B*n+en*, wobei A∈ {*b, ch, d, f, g, j, k, p, sch, t, w*} und B=A, oder B = *m*; außerdem Dental +*en*. (*trocknen, atmen, landen, warten*)
Klasse 3: Infinitiv auf Sibilant + *en* (*mixen, tanzen, rasen, schweißen*)
Klasse 4: Infinitiv auf Konsonant + *ern* (außer Klasse 5!) (*klettern, bessern*)
Klasse 5: Infinitiv auf
Konsonant außer *g, h, l, n, r* + *gern* (*metzgern*) Vokal + *hern (nähern*)
Konsonant außer *h, l* + *lern* (*tischlern*)
Konsonant außer *h,n* + *nern* (*klempnern*)
Klasse 6: Infinitiv auf *-eln* (*sammeln*)
Klasse 7: Infinitiv auf *-ien* (*knien*)

Bei den starken Verben werden wie bei Müller (1991) vorgeschlagen, die orthographischen Besonderheiten bei der Anfügung der Flexionsendung unabhängig vom Ablautverhalten zur morphologischen Subklassifizierung der Verben herangezogen. Neben den bereits bei den schwachen Verben angeführten Phänomenen kommen hier noch die folgenden zum Tragen:

- optionaler *e*-Einschub vor *st* im Auslaut: *du schnitt(e)st* (eevsto)
- obligatorischer *t*-Ausfall nach *t*: *er hält* (tant)
- obligatorischer *st*-Ausfall nach *st*: *du birst* (stanst)

Das führt zu 12 vom Ablautverhalten unabhängigen, verschiedenen morphologischen Subklassen der starken Verben:

Tabelle 5. Morphologische Klassifikation der starken Verben

Merkmale	**Fl**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Infinitiv	en												

Tabelle 5. Morphologische Klassifikation der starken Verben

1.Sing Präs Ind	e												
2.Sing Präs Ind	st		sanS	sanS	eevs t				stan st		ss/ß sanS	sanS	sanS ss/ß
3.Sing Präs Ind	t				eevt		tant		tant		ss/ß		ss/ß
1.Plu Präs Ind	en												
2.Plu Präs Ind	t				eevt	eevt	eev		eevt		ss/ß		ss/ß
3.Plu Präs Ind	en												
1.Sing Präs Konj	e												
2.Sing Präs Konj	est												
3.Sing Präs Konj	e												
1.Plu Präs Konj	en												
2.Plu Präs Konj	et												
3.Plu Präs Konj	en												
1.Sing Prät Ind											ss/ß	ss/ß	
2.Sing Prät Ind	st			sanS	eevs to	eevs to	eevs to	eevs to	eevs to		sanS ss/ß	sanS ss/ß	sanS
3.Sing Prät Ind											ss/ß		
1.Plu Prät Ind	en									eane			
2.Plu Prät Ind	t				eevt	eev	eevt	eevt	eev		ss/ß	ss/ß	
3.Plu Prät Ind	en									eane			
1.Sing Prät Konj	e												
2.Sing Prät Konj	est												
3.Sing Prät Konj	e												
1.Plu Prät Konj	en												
2.Plu Prät Konj	et												
3.Plu Prät Konj	en												
Part Präs	end												
Part Perfekt	ge- en												
Imperativ Sing	(e)										(ss/ ß)		(ss/ ß)
Imperativ Plu	t				eevt	eevt	eevt		eevt		ss/ß		ss/ß

Bei den unregelmäßigen Verben lohnt sich aufgrund der geringen Anzahl eine weitere Subklassifikation nicht. Sie werden einfach aufgezählt.

2.3.5 Problemfälle

In diesem Abschnitt sollen einige Problemfälle für die oben beschriebene Klassifikation vorgestellt werden. Da sind sowohl prinzipielle Problembereiche aufzuführen, wie auch die Problematik der Einordnung einzelner Lexeme in dieses Schema.

2.3.5.1 Mehrfachklassifikation

Da die Wortklassen neben morphologischen Eigenschaften vor allem auf syntaktischen Funktionen beruhen, kommt es bei Wörtern, die verschiedene syntaktische Funktionen erfüllen können, zu einer Mehrfachklassifikation. Davon betroffen sind sowohl die Funktionswortarten als auch die Hauptwortarten, wie am Beispiel der adjektivischen Nomen deutlich wurde. Hier muß man, um eine unnötige Vervielfachung der Einträge zu verhindern, zwischen dem systematischen Vorkommen in anderen Funktionen und zufälligen Idiosynkrasien unterscheiden. Gibt es eine Regularität, nach der alle (oder bis auf Ausnahmen alle) Elemente einer Wortklasse auch in einer anderen syntaktischen Funktion auftreten, so ist dies als allgemeine Regel zu formulieren, die mit dem Lexikon nichts zu tun hat. Potentiell kann nahezu jedes Adjektiv nominal gebraucht werden:

(1) das Schöne, der Gelbe, die Schlaue, ...

Eine explizite Kodierung aller Adjektive als adjektivische Nomen wäre redundant, da die Flexionsformen beim Gebrauch als Nomen mit denen des Adjektivs übereinstimmen. Nur adjektivische Nomen vom Typ *Beamter* müssen explizit als adjektivisches Nomen kodiert werden, da es bei ihnen kein entsprechendes Adjektiv gibt. Auch das adverbiale Auftreten der Adjektive wird durch ein Merkmal +*adverbial* beim Adjektiv kodiert. Nur in Fällen, in denen das Adverb lexikalisiert ist und die entsprechenden Konstruktionen ambig sind, wie im folgenden Beispiel, wird ein gleichlautendes Adverb explizit kodiert.

(2) Er sagte das bestimmt.

Dieser Satz kann zum einen interpretiert werden als *er sagte das sicher*, zum anderen als *er sagte das nachdrücklich*.

Ein ähnlicher Fall wie bei den adjektivischen Nomen liegt bei Pronomen und Determinatoren vor. Im DKL wird davon ausgegangen, daß sämtliche Determinatoren auch pronominal auftreten können, also auch der definite Artikel.[1] Probleme treten hier

1. Fälle wie *der, den ich gestern habe* oder *das da drüben* werden als pronominale Verwendung des definiten Artikels gewertet und nicht etwa als Determinatorphrase mit Satz- bzw. Adverbialkomplement des Determinators.

lediglich dann auf, wenn die pronominale Form nicht mit der entsprechenden Form des Determinators übereinstimmt:

(3) Ein/kein Kind hat den Ball geholt.

(4) Eines/keines hat den Ball geholt.

Bei Determinatoren wie *beide* tritt sogar der Fall auf, daß das Merkmal Singular zwar bei pronominalem Gebrauch nicht aber beim Determinator realisiert werden kann:

(5) Beides wurde zu meiner Zufriedenheit erledigt.

(6) *Beide Auftrag wurde zu meiner Zufriedenheit erledigt.

In diesen Fällen müssen die im Determinatorparadigma nicht auftretenden Formen bei den Pronomen explizit kodiert werden.

Im Gegensatz zu diesen Fällen handelt es sich bei der Mehrfachklassifizierung vieler Funktionswörter um zufällige Idiosynkrasien, die alle explizit kodiert werden müssen.

2.3.5.2 Konversionsprozesse

Während es sich bei den oben besprochenen Fällen um Wortartwechsel bei unveränderter Morphologie (höchstens bei einzelnen Formen eines Paradigmas konnten Unterschiede auftreten) handelte, sollen nun Fälle betrachtet werden, bei denen Lexeme ohne explizite Derivation in einer anderen Wortart und mit entsprechend anderer Morphologie auftreten. Beispiele hierfür sind unter anderem die partizipialen Adjektive und nominalisierten Infinitive. Während im Fall der nominalisierten Infinitive die Morphologie prädiktabel ist (die Genitiv-Singular-Form lautet Infinitiv+*s*, Plural ist auch morphologisch nicht möglich), lassen sich die morphologischen und morphosyntaktischen Eigenschaften Graduierbarkeit und attributiver, adverbialer oder prädikativer Gebrauch nicht vom Verb ableiten. Eine Lösungsmöglichkeit wäre die Kodierung der adjektivischen Eigenschaften des Partizips im Rahmen des Verbparadigmas. Aus Gründen der Systematik wurde beim DKL jedoch der Weg der expliziten Kodierung aller Partizipien als Adjektive gewählt.[1]

2.3.5.3 Unikale und gebundene Lexeme

In Analogie zu den Begriffen *unikales Morphem* und *gebundenes Morphem* sollen hier die Begriffe *unikales Lexem* und *gebundenes Lexem* eingeführt werden.

> Ein *unikales Lexem* ist ein Lexem, das nur innerhalb bestimmter fester idiomatischer Wendungen vorkommt. Die Semantik und

1. Natürlich werden dazu die Partizipien mit einer Defaultkodierung als Adjektiv automatisch aus den Verben extrahiert, so daß nur noch die Defaultwerte für die morphologische Klasse und die Distributionsmerkmale in den Sonderfällen korrigiert werden müssen.

Etymologie unikaler Lexeme ist für das Verständnis der idiomatischen Wendung häufig völlig irrelevant.

Ein *gebundenes Lexem* ist ein Lexem, das nur in bestimmten Konstruktionen aber nie frei (d.h. außerhalb dieser Konstruktionen) vorkommt. Gebundene Lexeme sind im Gegensatz zu unikalen Lexemen in einem bestimmten Konstruktionstyp produktiv.

Ein Lexem, das weder unikal noch gebunden ist, wird als *freies Lexem* bezeichnet.

Der wesentliche Unterschied zwischen unikalen und gebundenen Lexemen ist die Produktivität. Unikale Lexeme sind in der Regel auf eine feste Wendung beschränkt (wie z.B. *Daffke* in *aus Daffke*, was in Berlin soviel wie *zum Spaß* bedeutet). Gebundene Lexeme dagegen sind innerhalb eines Konstruktionstyps in gewissem Umfang produktiv. Dazu zählen etwa abtrennbare Verbpartikel, die nicht frei vorkommen, wie zum Beispiel *instand* in *instandsetzen/bringen/halten/...* oder Prädikative von Stützverbkonstruktionen, wie *schuld, schade, imstande*. Während bei den unikalen Lexemen die Wortartklassifikation in der Regel keine Probleme bereitet, ist der Status der gebundenen Lexeme häufig unklar. Dadurch, daß sie auf bestimmte Kollokationen beschränkt sind, lassen sich die distributionellen Tests zur Klassifizierung kaum anwenden. Ungeachtet dessen, daß die distributionellen Eigenschaften von gebundenen und unikalen Lexemen im Rahmen eines speziellen Lexikons der Mehrwortlexeme explizit zu beschrieben sind, müssen diese Lexeme als einfache Formen bezüglich ihrer morphologischen Eigenschaften kodiert werden. Es bieten sich bei der Kodierung 3 Möglichkeiten: die unfreien Lexeme werden der Wortart, die am ehesten in Frage kommt, zugeordnet, obwohl sie die Definition dieser Wortart nicht erfüllen, alternativ könnte man eine bzw. mehrere Klassen speziell für solche Morpheme annehmen, wobei es eventuell zu einer Vielzahl extrem schwach besetzter Klassen kommt und drittens besteht die Möglichkeit, diese Lexeme als nicht kategorisierbar, also XINC zu klassifizieren. Da jede dieser Vorgehensweisen ihre Nachteile hat, wird beim DKL von Fall zu Fall entschieden, welches Vorgehen gewählt wird. Falls sich ein Lexem in eine der definierten Wortklassen aufgrund der beschriebenen Kriterien einordnen läßt, so wird diese Wortart gewählt. Das ist bei den unikalen Morphemen, die Kopf einer entsprechenden Phrase sind, häufig der Fall aber auch bei den oben erwähnten Prädikativen, die sich wie prädikative Adjektive verhalten. Falls es eine hinreichende Anzahl gebundener Lexeme gibt, die sich alle distributionell und evtl. auch morphologisch ähnlich verhalten aber in keine der definierten Wortklassen fallen, so wird eine neue Klasse definiert, wie etwa bei den abtrennbaren Verbpartikeln.[1] Nur in Fällen, in denen keine

1. Bei den abtrennbaren Verbpartikeln stellt sich dann allerdings die Frage, ob in dieser Klasse auch abtrennbare Verbpartikel, die bereits in anderen Klassen aufgeführt sind, doppelt klassifiziert werden. Eine solche Mehrfachklassifizierung wäre redundant und ist durch die Annahme, daß Präpositionen, Adverben, Nomen und explizit als Verbpartikel kodierte Lexeme in dieser Funktion auftreten können, zu verhindern.

Einordnung ins Wortklassenschema möglich ist und keine geschlossene Gruppe von Lexemen mit gleichem Verhalten existiert, wird die Restkategorie XINC zugewiesen.

2.4 Das EF-FLEX-Lexikon

Die Ausführungen in den vorigen beiden Abschnitten betrafen das DKL-EF, das Lexikon der einfachen Formen. Aus diesem Lexikon der einfachen Formen wird das EF-FLEX, das Lexikon der flektierten Formen generiert, das in den meisten Anwendungen zum Einsatz kommt. Umfangreiches Beispielmaterial zu den Einträgen im FLEX-Lexikon ist in Anhang B auf Seite 200 aufgeführt. Abschnitt 2.4.1 ist einer detaillierten Beschreibung der morphologischen Prozesse des Deutschen gewidmet. Es werden an dieser Stelle die zum Teil bereits in den Kodiertabellen (siehe Abschnitt 2.3.4 auf Seite 45) genannten zur Unterscheidung der einzelnen morphologischen Klassen wichtigen Prozesse sowohl deskriptiv als auch operational beschrieben. Die eigentliche Morphologiekomponente kommt sowohl bei der Generierung der Vollformen, als auch bei der Eingabe neuer Einträge ins DKL zum Einsatz. In Abschnitt 2.4.2.1 wird die Morphologiekomponente ausführlich erläutert. Danach wird auf die speziellen Funktionen zur Generierung des DKL-FLEX noch näher eingegangen.

2.4.1 Die morphologischen Prozesse des Deutschen und ihre Implementierung

2.4.1.1 Morphologische Prozesse

Als *morphologischen Prozeß* bezeichnet man diejenigen Operationen, die eine morphologische Funktion kodieren. Die morphologischen Prozesse, die in den verschiedenen Sprachen auftreten, lassen sich in bestimmte Typen unterscheiden. Prinzipiell unterscheidet man zwischen *additiven* morphologischen Prozessen und *nicht-additiven* morphologischen Prozessen sowie der *Suppletion*. Unter additiven morphologischen Prozessen versteht man im Gegensatz zu nicht-additiven solche Prozesse, bei denen das sprachliche Material in irgendeiner Weise vermehrt wird.

Bei den additiven Prozessen unterscheidet man wiederum danach, ob tatsächlich neue Segmente hinzugefügt werden oder ob ein vorhandenes Segment nur additiv verändert wird. Die letztgenannten nennt man auch *modulatorisch-additiv.* Ein Beispiel für einen modulatorisch-additiven Prozeß die lateinische Pluralmarkierung durch Längung eines Vokals (etwa bei domus = Haus). Weitaus verbreiteter sind jedoch die additiven Prozesse, die Segmente hinzufügen. Hier unterscheidet man zwischen *Affigierung* und *Reduplikation*. Bei der Reduplikation wird eine Phonemfolge des Stamms vor oder nach dem Stamm wiederholt. Je nach Grad der Veränderung der Phonemfolge spricht man von *partieller* oder von *totaler* Reduplikation. Reduplikation spielt im heutigen Deutsch in der Flexionsmorphologie keine Rolle mehr, sie ist in der Wortbildung jedoch noch vereinzelt anzutreffen, wie etwa in *tagtäglich* oder *wortwörtlich*. Für die Flexionsmorphologie von größerer Bedeutung ist dagegen die Affigierung mit ihren Unterklassen *Präfigierung, Suffigierung*, *Infigierung* und *Zirkumfigierung.*

Bei den nicht-additiven Prozessen lassen sich je nach Art der Operation ebenfalls verschiedene Typen voneinander abgrenzen. Für die Flexionsmorphologie wichtig sind

die *modulatorischen* Prozesse. Bei ihnen handelt es sich um Phonemveränderungen, die keine quantitative Veränderung bewirken, wie dies im Deutschen beim Ablautverhalten der starken Verben der Fall ist. Aber auch *subtraktive Prozesse*, auch *Elisionen* genannt, trifft man im Deutschen häufig an; ebenso die *ø-Prozesse*, wenn morphologische Merkmale an der Oberfläche nicht realisiert werden. Im Deutschen völlig ungebräuchlich, jedoch in Indianersprachen verbreitet, ist die *Metathese*. Hierbei handelt es sich um Phonemumstellungen innerhalb des Stamms.

Mit den bisherigen Mitteln schwer zu vergleichen ist die *Suppletion*. Hier wird innerhalb eines Paradigmas an bestimmten Stellen ein völlig neuer Stamm verwendet.

2.4.1.2 Für die Kodierung relevante Prozesse

Die morphologischen Prozesse dienen als diskriminierende Merkmale bei der morphologischen Klassifikation der Lexeme. Es wird nicht zwischen morphologisch motivierten Prozessen und orthographisch bzw. phonologisch motivierten Prozessen unterschieden. Die beiden Grundtypen von Operationen sind:

- Anfügen einer bestimmten Zeichenkette am rechten Rand (Suffigierung)
- Anfügen einer bestimmten Zeichenkette am linken Rand (Präfigierung)
- Einfügen einer bestimmten Zeichenkette nach einer bestimmten Anzahl von Zeichen vom rechten Rand (Infigierung)
- Löschen einer bestimmten Anzahl von Zeichen am rechten Rand (Subtraktion)

Neben diesen beiden allgemeinen Operationen gibt es noch Zeichenketten-spezifische Operationen:

- Wechsel zwischen *ss* und *ß*.
- *e*-Ausfall vor Dentalen
- Umlautung: *a-> ä, aa-> ä, o->ö, oo->ö, u->ü, uu->ü*

Ablautprozesse werden wie Suppletion nicht durch Operationen auf Stämmen sondern durch Auflistung der allomorphen Stämme beschrieben. Weitere Operationen, wie *e*-Einschub vor Sibilanten, Ersetzung einer Zeichenkette durch eine andere, usw. sind durch eine Kombination der oben angeführten Operationen definiert. Die Zeichenketten-unabhängigen Operationen sind in diesen Fällen die Bausteine.

Die Definition der morphologischen Prozesse beruht auf der Grundannahme, daß sich morphologische Prozesse in zwei Teilprozesse aufteilen lassen:

1. stamminterne Prozesse, die allomorphe Stämme generieren
2. konkatenative Prozesse, die auf allomorphen Stämmen operieren

Die folgenden Tabellen zeigen die Realisierungsmöglichkeiten der morphologischen Funktionen bei den einzelnen Wortarten:

Tabelle 6. Morphologische Prozesse bei Nomen

Kasusmarkierung	**Numerusmarkierung**	**orthographisch bzw. phonologisch motivierte Prozesse**
Suffigierung	Suffigierung Umlautung Subtraktion Suppletion (selten)	Wechsel von *ss* und *ß*

Tabelle 7. Morphologische Prozesse bei Adjektiven

Kongruenzmerkmale	**Graduierung**	**orthographisch bzw. phonologisch motivierte Prozesse**
Suffigierung	Suffigierung Umlautung Suppletion	Wechsel von *ss* und *ß* *e*-Ausfall vor Dental

Tabelle 8. Morphologische Prozesse bei Verben

Tempus/Modus	**Person/Numerus**	**infinite Formen**	**orthographisch bzw. phonologisch motivierte Prozesse**
Suffigierung Ablaut Suppletion	Suffigierung Ablaut Suppletion	Suffigierung Präfigierung Infigierung Ablaut Suppletion	Wechsel von *ss* und *ß* *e*-Ausfall vor Dental

2.4.2 Automatische Vollformengenerierung und Lexikonerweiterung

Wie bereits in der Einleitung erwähnt, wird die Morphologiekomponente von zwei Programmen genutzt. Einerseits dient sie dazu, aus den Einträgen des CISLEX die Vollformen mitsamt einer Liste der durch die jeweilige Form repräsentierten morphosyntaktischen Merkmale zu generieren, andererseits dient die Morphologiekomponente im Rahmen eines Eingabetools der Bestimmung des Paradigmas. Das Eingabetool ermöglicht die Kodierung von Einträgen des DKL ohne genaue Kenntnis des Klassifikationsschemas. Die Klassifikation beruht auf bestimmten eingegebenen Wortformen, aus denen sich intern eindeutig die Klasse bestimmen läßt. Der Benutzer muß die Wortart auswählen und wird dann über Dialogfenster die notwendigen Informatio-

nen abgefragt, die das System benötigt, um die vollständige Kodierung vorzunehmen. Sobald die Klasse erkannt ist, werden alle Formen generiert und dem Benutzer zur Kontrolle ausgegeben, bevor die neuen Einträge ins Lexikon übernommen werden.

Das bedeutet, daß die Morphologiekomponente folgende beiden Funktionen ausführt:

- <Grundform, Flexionsform, Merkmale> -> morphologische Klasse (zur Eingabe)
- <Grundform, Klasse, Merkmale> -> Flexionsform (zur Vollformgenerierung)

Die Analyse von Flexionsformen zu Grundform, Klasse und Merkmalen ist theoretisch möglich, als morphologische Analysekomponente in der Praxis jedoch zu ineffizient.

Beide Programmteile sind in PrologII+ implementiert. Der Kern, die eigentliche Morphologiekomponente besteht wiederum aus zwei Teilen:

1. Paradigmenbeschreibungen in Form einer Prologdatenbank
2. Regelkomponente zur Ausführung der morphologischen Prozesse

2.4.2.1 Die Morphologiekomponente

2.4.2.1.1 Paradigmenbeschreibungen

Die Paradigmenbeschreibungen sind in Form von Prologfakten abgespeichert, die Argumente sind die Wortart, morphologische Klasse und ein n-Tupel mit Beschreibungen der verschiedenen Formen:

`paradigma(W_ortart,K_lasse,<F_orm`$_1$`,..,F_orm`$_n$`>)->;`

Die Beschreibung der Formen ist jeweils ein Paar, bestehend aus Endung und Liste der auszuführenden Operationen:

`<E_ndung,O_peration`$_1$`....O_peration`$_k$`.nil>`

Die Operationen sind wie in Abschnitt 2.4.1 definiert.

Es existieren Prolog-Paradigmenbeschreibungen für alle flektierenden Wortklassen, also für Nomen, Verben, Adjektive und Determinatoren/Quantoren. Die Beschreibungen der Determinatoren/Quantoren sind nur der Vollständigkeit halber definiert. Tatsächlich werden Determinatoren und Quantoren im DKL bereits als Vollformen aufgeführt.

Nomen
Wie in Abschnitt 2.3.4 bereits ausgeführt, werden die Nomen getrennt nach Singulartyp und Pluraltyp klassifiziert. Im Singular werden alle Formen beschrieben, während im Plural nur die Nominativ- und die Dativform beschrieben wird. Es gibt zwar kein Paradigma, bei dem alle 4 Singular-Kasusformen verschieden sind, aber es gibt auch

keine zwei Kasusformen, die in *allen* Singularparadigmen gleich sind. Im Plural sind prinzipiell die Nominativ-, Akkusativ- und Genitivformen gleich. Lediglich der Dativ kann extra markiert sein.

Beispiel:

```
paradigma(<nomen,nomsing>,NS0,
  <<"",nil>,<"",nil>,<"",nil>,<"",nil>>)->;

paradigma(<nomen,nomplu>,NP11,
  <<"",umlaut(x,u_x).nil>,<"n",umlaut(x,u_x).nil>>)->;
```

Die beiden Beispiel-Paradigmenbeschreibungen definieren das Flexionsverhalten von Nomina vom Typ “Mutter”. Die Formbeschreibungen des Singularparadigmas sind im obigen Beispiel alle identisch: es gibt keinen allomorphen Stamm und keine Flexionsendung. Im Plural wird in beiden Fällen umgelautet; im Falle von Nominativ, Akkusativ und Genitiv wird keine Endung angehängt und im Dativ wird die Endung “n” angefügt.

Adjektive

Bei den Adjektiven ist unter morphologisch-orthographischen Gesichtspunkten lediglich die Bildung des Stamms der Komparationsformen sowie der Stamm der attributiven Positivformen interessant. Die Flexive für Kasus, Genus und Numerus werden an diese Stämme völlig regulär angefügt. Die Formbeschreibungen der Adjektivparadigmen sind aus diesem Grund ein 9-Tupel der Form:

```
<P_rädikativstamm, A_ttributivstamm, K_omparativstamm,
  S_uperlativstamm,<"e",nil>,<"em",nil>,<"en",nil>,
  <"er",nil>,<"es",nil>>
```

Die ersten 4 Stellen der Formenbeschreibung enthalten die Beschreibung der 4 verschiedenen Stämme, Prädikativstamm im Positiv, Attributivstamm im Positiv, Komparativstamm und Superlativstamm. Die weiteren 5 Stellen enthalten die Flexionsendungen für die Kasus-Genus-Numerus-Markierung. Es ist Aufgabe der Regelkomponente, die entsprechende Beschreibung der Stammbildung mit der jeweiligen Beschreibung der Flexionsendung zu kombinieren.

Beispiel:

```
paradigma(adj,ADJ1,
  <<"",nil>,<"",ee_ausfall_vor(d).nil>,
  <"er",ee_ausfall_vor(d).nil>,<"st",nil>,<"e",nil>,
  <"em",nil>,<"en",nil>,<"er",nil>,<"es",nil>>
  )->;
```

Das Paradigma ADJ1 beschreibt die Formen von Adjektiven vom Typ *dunkel*, die als einzige Besonderheit die e-Elision vor Dental bei der Bildung von Komparativ-, Superlativ- und Attributivstamm aufweisen.

Verben
Bei den Verben wird zwischen starken und schwachen Verben unterschieden. Während bei den schwachen Verben 13 Formbeschreibungen genügen um für alle Klassen den gesamten Formenbestand von 29 Formen im Paradigma zu beschreiben, war bei den starken Verben eine solche Reduzierung nicht möglich, hier werden sämtliche 29 Formen beschrieben. Paradigmenbeschreibungen für schwache Verben haben demnach ein 13-Tupel als Formenbeschreibung, während die Paradigmenbeschreibungen der starken Verben ein 29-Tupel haben. Es wurde bei den starken Verben darauf verzichtet, die Ablautphänomene mit in die Paradigmenbeschreibung aufzunehmen, da dies eine Explosion von Klassen bedeutet hätte (vgl. Abschnitt 2.3.4.3). Die Paradigmenbeschreibungen beinhalten lediglich die orthographischen Regularitäten bzw. Irregularitäten, die bei der Realisierung des Tempus-Person-Numerus-Morphems am jeweiligen Stamm auftreten.
Bei der Kodierung wird das Ablautverhalten der starken Verben und die Konkatenation mit den Tempus-Person-Numerus-Morphemen zwar in einem Kode kombiniert, für die Generierung, sowie für die Analyse muß die Information über drei Primär- und zwei Sekundärstämme jedoch separat geliefert werden.

Beispiele:

```
paradigma(<verb,stark>,VST1,<<"en",nil>,<"e",nil>,<"st",nil>
  ,<"t",nil>,<"en",nil>,<"t",nil>,<"en",nil>,<"e",nil>,<"est
  ",nil>,<"e",nil>,<"en",nil>,<"et",nil>,<"en",nil>,<"",nil>
  ,<"st",nil>,<"",nil>,<"en",nil>,<"t",nil>,<"en",nil>,<"e",
  nil>,<"est",nil>,<"e",nil>,<"en",nil>,<"et",nil>,<"en",nil
  >,<"end",nil>,<"en",prefix("ge").nil>,<"e",nil>.<"",nil>.n
  il,<"t",nil>>) ->;

paradigma(<verb,schwach>,VSW1,<<"en",nil>,<"e",nil>,<"st",ni
  l>,<"t",nil>,<"est",nil>,<"et",nil>,<"te",nil>,<"ten",nil>
  ,<"test",nil>,<"tet",nil>,<"end",nil>,<"t",prefix("ge").ni
  l>,<"",nil>.<"e",nil>.nil>)->;
```

Beide Paradigmen beschreiben starke bzw. schwache Verben, die keine orthographischen Besonderheiten aufweisen.

Wegen des geringen Umfangs und der Abgeschlossenheit der Subklassen werden für völlig unregelmäßige Verben (Rückumlautverben, Präteritopräsentia,...) keine Paradigmenbeschreibungen erstellt. Hier erfolgt die Generierung der Vollformen manuell.

2.4.2.1.2 Regeln

Die Regeln der Morphologiekomponente interpretieren in erster Linie die Paradigmenbeschreibungen. Sie sind bidirektional, indem sie sowohl der Analyse von flektierten Formen, als auch der Generierung der Flexionsformen dienen. Insbesondere die Regeln, die die Operationen interpretieren, sind sowohl in der Lage, die Operation auf

einer Grundform bzw. auf einem Stamm, als auch die inverse Operation auf einer flektierten Form auszuführen.

Die Schnittstelle zum Gernerierungsprogramm ist das Prädikat *generiere_formen*:

```
generiere_formen(W_ortklasse,S_ubklasse,K_ode,W_ort,<S_tamm1
  , ...,Stammj>,<F_orm1,...,F_ormm>)->
  paradigma(<W_ortklasse,S_ubklasse>,K_ode,<F_ormbeschr1,...
  , Formbeschrn>)
  flektiere(S_tammi,F_ormbeschrj,F_orm1)
  .
  .
  .
  flektiere(S_tammℓ,F_ormbeschri,F_ormk);
```

generiere_formen hat als Argumente die Wortart, die Subklasse (z.B. stark/schwach bei Verben), den morphologischen Kode des Wortes aus dem DKL, das Wort in seiner Grundform, ein Tupel, das die verschiedenen Stämme enthält, und ein Tupel, das die Flexionsformen enthält. Das Stämmetupel enthält bei starken Verben die abgelauteten Stämme, bei schwachen Verben, den Verbstamm und ansonsten die Grundform des Wortes. Beim Aufruf dieses Prädikats zur Generierung sind die Argumente *Wortklasse, Subklasse, Kode, Wort* und *Stämme* belegt.

Das Prädikat *generiere_formen* könnte (mit anderer Variablenbelegung) auch als Schnittstelle zur Formanalyse aufgrund von eingebenen Formen zu einer fixen Grundform dienen. Aus Effizienzgründen ist bei dieser Schnittstelle lediglich die Abfolge der Prädikate *flektiere* und *paradigma* geändert:

```
analysiere_formen(W_ortklasse,S_ubklasse,K_ode,W_ort,<S_tamm
  1, ...,Stammj>,<F_ormℓ,...,F_ormm>)->
  flektiere(S_tammi,F_ormbeschrj,F_orm1)
  .
  .
  .
  flektiere(S_tammℓ,F_ormbeschri,F_ormk)
  paradigma(<W_ortklasse,S_ubklasse>,K_ode,<F_ormbeschr1,...
  , Formbeschrn>);
```

Hier sind beim Aufruf maximal folgende Argumente belegt: *Wort, <Stamm$_1$,..., Stamm$_j$>,* ein Teil der Formen aus *<F_orm$_1$,...,F_orm$_k$>, Wortklasse* und *Subklasse*. Die Stämme sind, bedingt durch die Eingabe, nur bei den starken Verben bekannt. Die Informationen über Wortklasse und Subklasse sind für die Ausführung nicht notwendig, ergeben sich jedoch dadurch, daß die Eingabemasken von der Wortklasse und Subklasse abhängig sind.

Das Prädikat *flektiere* beschreibt die Relation zwischen Stamm, Flexionsform und Formbeschreibung. Es berechnet zum einen aus Stamm und Flexionsform die Formbe-

schreibung, zum anderen aus Stamm und Formbeschreibung die Flexionsform oder aus Flexionsform und Formbeschreibung den Stamm.

```
flektiere(s_tamm,<e_ndung,o_perationen>,w_ort)->
  allo_stamm(s_tamm,o_perationen,s_tamm’)
  conc_string(s_tamm',e_ndung,w_ort);
```

Gemäß der Trennung von suffigierenden Prozessen und nicht-suffigierenden Prozessen, beschreiben die Operationen der Formbeschreibung den allomorphen Stamm. Das Prädikat *allo_stamm* beschreibt die Bildung des allomorphen Stammes. Die Suffigierung mit dem Flexiv geschieht durch das interne Prädikat *conc_string*, dessen drittes Argument die Konkatenation der beiden ersten Argumente ist.

Im Rahmen des Prädikats *allo_stamm* werden nacheinander die einzelnen Operationen der Liste aus der Formenbeschreibung ausgeführt. Dies geschieht in dem Prädikat *fuehre_aus*, das auf Stämmen bzw. Flexionsformen in Listenform operiert. Für jede morphologische Operation (vgl. Abschnitt 2.4.1) existiert eine Version von *fuehre_aus*. Hier ein Beispiel für das Einfügen eines Infixes nach einer bestimmten Anzahl von Zeichen:

```
fuehre_aus(s1,infix(o_ffset,s_tring),s2)->
  split(s_tring,l_s)
  fuege_ein(s1,o_ffset,l_s,s2);
```

split ist ein vordefiniertes Prädikat zur Umwandlung von Strings in Listen, *fuege_ein* fügt die Liste l_s nach o_ffset-vielen Elementen vom Kopf der Liste s1 aus gerechnet ein und ergibt dann die Liste s2.

2.4.2.2 Vollformengenerierung

Zur Vollformengenerierung wird das Lexikon zuerst mit Hilfe eines Perl-Skripts (*plexconv*) in eine Prologdatenbank konvertiert, dabei werden auch Sonderzeichen und Umlaute, die in Prolog nicht verarbeitet werden können, in eindeutige Sequenzen anderer Zeichen konvertiert (diese Sequenzen werden nach der Generierung wieder in das ursprüngliche Zeichen rückkonvertiert). Die Lexikoneinträge der Prologdatenbank haben folgendes Format:

```
<W_orttyp,W_ort,[S_taemme,]K_ode>->;
```

Der Worttyp ist ein kombiniertes Kürzel, das Wortklasse und Subklasse zusammenfaßt, Stämme sind wiederum nur bei starken Verben möglich.

Beispiele:

```
<vv_st,"zwingen",<"zwing","zwing","zwang","zwa!eng","zwung">
  , VST1> ->;

<adj,"brutal",ADJ0>->;
```

Mit der Morphologiekomponente werden aus diesem Input zu jedem Wort die verschiedenen Flexionsformen generiert und mit einem Merkmalskode versehen. Der Output ist ein Lexikon in einem vorläufigen Format:

`%Wortform,Grundform.Kode:merkmalsbündel`

Die Morphologiekomponente generiert zu jeder möglichen Merkmalskombination die entsprechende Form. Eine weitere Perlroutine, *pformconv*, konvertiert die Ausgabe der Morphologiekomponente ins endgültige Format, in dem jede Form nur einmal auftaucht und mit einer Liste aller möglichen Merkmalskombinationen versehen ist. Eine weitere Aufgabe dieser Routine ist die Rekonstruktion von Umlauten und anderen Sonderzeichen. Die Einträge haben nach dieser Konvertierung das endgültige FLEX-Format:

`%Wortform,Grundform.Kode:merkmalsbündel`$_1$`:...:merkmalsbündel`$_n$

Beispiel:

%zwinget,zwingen.VST1:2mGc
%zwangt,zwingen.VST1:2mVi
%zwang,zwingen.VST1:1eVi:3eVi
%zwingest,zwingen.VST1:2eGc
%zwingt,zwingen.VST1:2mGi:3eGi
%zwingend,zwingen.VST1:OE
%zwingst,zwingen.VST1:2eGi
%gezwungen,zwingen.VST1:OZ
%zwängen,zwingen.VST1:1mVc:3mVc
%zwangst,zwingen.VST1:2eVi
%zwingen,zwingen.VST1:OI:1mGi:3mGi:1mGc:3mGc
%zwängest,zwingen.VST1:2eVc
%zwangen,zwingen.VST1:1mVi:3mVi
%zwänge,zwingen.VST1:1eVc:3eVc
%zwinge,zwingen.VST1:1eGi:1eGc:3eGc
%zwänget,zwingen.VST1:2mVc

Für die verschiedenen am Generierungsprozeß beteiligten Dateien gilt folgende Namenkonvention:

<wortart>.pro: Lexikon im Prologformat

<wortart>.form: Output des Morphologieprogramms

<wortart>.flek: Vollformenlexikon im FLEX-Format

<wortart>.fl: Liste der Vollformen

Die Merkmalskombinationen sind im FLEX-Format aus Platzgründen nicht explizit angegeben, sondern werden durch Kürzel repräsentiert. Die Bedeutung der einzelnen Kürzel ist in den folgenden Tabellen dargestellt.

Tabelle 9. Merkmalskodierung

Merkmalstyp	Merkmal	relevante Wortarten	Bez.
Kasus	nominativ	Nomen, Adjektiv, Det/Quant, Pronomen, Präposition	n
	akkusativ		a
	dativ		d
	genitiv		g
Person	1	Det/Quant, Pronomen, Verb	1
	2		2
	3		3
	unspez.		0
Numerus	singular	Adjektiv, Det/Quant, Pronomen, Verb	e
	plural		m
Genus	fem	Adjektiv, Det/Quant, Pronomen, Nomen	F
	mask		M
	neut		N
	unspez		U
Deklination	stark	Adjektiv, Det/Quant	x
	schwach		y
	gemischt		z
	prädikativ		u
Komparation	pos	Adjektiv	p
	komp		k
	super		s
Tempus	präsens	Verb	G
	imperfekt		V
	unspez		O
Modus	indikativ	Verb	i
	konjunktiv		c
	imperativ		b
infinite Formen	partizip1	Verb	E
	parttizip2		Z
	infinitiv		I

Tabelle 10. Die Kodierung der Merkmale bei Verben

	1.Person Singular	2.Person Singular	3. Person Singular	1.Person Plural	2. Person Plural	3.Person Plural
Präsens Indikativ	1eGi	2eGi	3eGi	1mGi	2mGi	3mGi
Präsens Konjunktiv	1eGc	2eGc	3eGc	1mGc	2mGc	3mGc
Imperfekt Indikativ	1eVi	2eVi	3eVi	1mVi	2mVi	3mVi
Imperfekt Konjunktiv	1eVc	2eVc	3eVc	1mVc	2mVc	3mVc
Imperativ	eOb			mOb		
Infinitiv	OI					
Partizip I	OE					
Partizip II	OZ					

Tabelle 11. Die Kodierung der Merkmale bei Substantiven

	Nominativ	Genitiv	Dativ	Akkusativ
Maskulin Singular	neM	geM	deM	aeM
Maskulin Plural	nmM	gmM	dmM	amM
Feiminin Singular	neF	geM	deF	aeM
Feminin Plural	nmF	gmF	dmF	amF
Neutrum Singular	neN	geN	deN	aeN
Neutrum Plural	nmN	gmN	dmN	amN

Tabelle 12. Die Kodierung der Merkmale bei Pronomen, Determinatoren und Quantoren

	Nominativ	Genitiv	Dativ	Akkusativ
1. Person Singular	n1eU	g1eU	d1eU	a1eU
2. Person Singular	n2eU	g2eU	d2eU	a2eU

Tabelle 12. Die Kodierung der Merkmale bei Pronomen, Determinatoren und Quantoren

		Nominativ	Genitiv	Dativ	Akkusativ
3. Person Singular	**Maskulin**	n3eM	g3eM	d3eM	a3eM
	Feminin	n3eF	g3eF	d3eF	a3eF
	Neutrum	n3eN	g3eN	d3eN	a3eN
1. Person Plural		n1mU	g1mU	d1mU	a1mU
2. Person Plural		n2mU	g3mU	d3mU	a3mU
3. Person Plural		n3mU	g3mU	d3mU	a3mU

Falls ein Determinator oder Pronomen keine Personenunterscheidung macht, so wird an dieser Stelle *0* kodiert. Völlig unflektierbare Determinatoren und Pronomen erhalten, wie auch die entsprechenden Adjektive, das Metamerkmal *X*.

Die Adjektivflexive sind, was die Markierung morpho-syntaktischer Merkmale betrifft, hochgradig ambig, denn die 5 verschiedenen Flexive realisieren alle möglichen Merkmalsbündel der Form <Kasus, Genus, Numerus, Deklinationstyp>, das sind 4*3*2*3 = 72 theoretisch mögliche Merkmalsbündel. Beachtet man, daß im Plural keine Genusunterscheidung stattfindet, so bleiben immer noch 48 verschiedene Merkmalskombinationen, die durch 5 verschiedene Flexive realisiert werden:

Tabelle 13. Adjektivflexion: Flexive

		Maskulin Singular	Feminin Singular	Neutrum Singular	Plural
stark	**Nominativ**	er	e	es	e
	Genitiv	en	er	em	en
	Dativ	em	er	en	er
	Akkusativ	en	e	es	e
schwach	**Nominativ**	e	e	e	en
	Genitiv	en	en	en	en
	Dativ	en	en	en	en
	Akkusativ	en	e	e	en
gemischt	**Nominativ**	er	s	es	en
	Genitiv	en	en	en	en
	Dativ	en	en	en	en
	Akkusativ	en	e	es	en

Die entsprechenden Adjektivformen erhalten die der obigen Tabelle entsprechenden Kodes ergänzt durch *p* für Positiv, *k* für Komparativ und *s* für Superlativ, je nachdem um welchen Stamm es sich handelt. Die Prädikativformen sind durch *up* (Prädikativ-Positiv), *uk* (Prädikativ-Komparativ) und *us* (Prädikativ-Superlativ) gekennzeichnet.

2.4.3 Fugenformen

Neben den Flexionsformen, die frei vorkommen, enthält das FLEX-Lexikon auch die Fugenformen der flektierenden Wortarten. Als Fugenform bezeichnet man Wortformen, die als Vorderglieder in Komposita auftreten. Bei den nicht-flektierenden Wortarten sind die Fugenformen (sofern überhaupt in Komposita möglich) identisch mit den Einträgen des EF-Lexikons und müssen deshalb nicht gesondert behandelt werden. Determinatoren und Pronomen, die nur in Ausnahmefällen als Vorderglieder von Komposita auftreten können (z.B. *Ichform*), werden nicht als mögliche Kompositavorderglieder zugelassen. Die Ausnahmefälle müssen gesondert behandelt werden. Bei den Adjektiven und Verben sind die Fugenformen weitgehend aufgrund der morphologischen Klassifikation prädiktabel. Als Fugenformen der Adjektive treten die Positiv-, Komparativ- und Superlativstämme auf. Die Positiv- und Komparativstämme sind bereits Bestandteil des Flexionsparadigmas, deshalb erübrigt sich auch hier die explizite Auflistung. Einzig der Superlativstamm gehört nicht zum Flexionsparadigma des Adjektivs und muß deshalb explizit als Fugenform ins FLEX-Lexikon aufgenommen werden. Die Kodierung erfolgt analog zu den Flexionsformen, anstelle der morphosyntaktischen Merkmale erscheint bei den Fugen jedoch ein Fugenmerkmal, das bei den Superlativstämmen *AFs* lautet.

Beispiele:

%best,gut.ADJu:AFs	*bestmöglich*
%höchst,hoch.ADJu:AFS	*Höchstleistung*[1]
%weitest,weit.ADJ11:AFs	*weitestreichend*

Bei den Verben werden die Stämme als Fugenformen ins FLEX-Lexikon aufgenommen. Sie erhalten das Fugenmerkmal VF.

Beispiele:

%geh,gehen.VST1:VF	*gehbehindert*
%tauch,tauchen.VSW1:VF	*Tauchsieder*
%kletter,klettern.VSW4:VF	*Kletterseil*

Neben diesen regulären Fugen gibt es auch Ausnahmen wie beispielsweise *löse* in *Lösegeld* oder *werde* in *Werdegang*. Hierbei handelt es sich um idiosynkratische Sonderfälle, die auch ins FLEX-Lexikon mit aufgenommen werden müssen.

1. Das Beispiel *Höchstleistung* zeigt auch, daß es sich tatsächlich um Fugenformen und nicht um wortinterne Flexion handelt, die bei A-A-Komposita häufig angenommen wird.

Bei den Nomen kann im Gegensatz zu den Verben und Adjektiven die Fugenform nicht aus der Grundform und der morphologischen Klasse vorhergesagt werden. Die häufigsten Fugenformen sind die Grundform (*Hauptstadt, Bahnhof, Tellerwäscher,* usw.) und die Form mit Fugen-s (*Rechtsgelehrter, Abtreibungsgesetze, Spaltungsreaktion,* usw.). Daneben gibt es aber eine Reihe weiterer Möglichkeiten zur Bildung von Fugen:

- Subtraktive Prozesse: *Kirch* von *Kirche* in *Kirchdach*.
- Anfügung anderer Fugenelemente: *Rind***er***braten, Seide***n***schal, herz***ens***gut, Man***es***kraft, Sprinterin***nen***duell,* usw.
- Subtraktion und Anfügung von Fugenelementen: *Weihnacht***s***feier, Paradigm***en***beschreibung, myth***en***beladen, Kakt***een***zucht, Kont***en***sperrung,* usw.
- Umlautung mit weiteren Operationen: *M***ä***nn***er***chor, Werkst***ä***tt***en***besichtigung,* usw.
- Wechsel von *ss* zu *ß*: *Adre***ß***buch, Delikate***ß***geschäft.*

Die Fugenformen der Nomen sind teilweise Bestandteil des Paradigmas, teilweise aber auch Formen, die nicht frei vorkommen, wie *Liebes, Kirch* oder *Weihnachts.* Aufgrund der Nichtvorhersagbarkeit der Fugenform müssen die Nomen neben der morphologischen Klassifizierung auch noch einzeln nach ihren Möglichkeiten, Fugenformen zu bilden, klassifiziert werden. Im FLEX-Lexikon müssen auch diejenigen Fugenformen, die Bestandteil des Flexionsparadigmas sind, als Fugen kodiert werden, da es nicht aus der Grundform und der Klasse ableitbar ist, welche Formen des Paradigmas als Fugenformen auftreten können. Die Arbeit von Langer (1994) gibt einen umfassenden Überblick über die Kodierung der nominalen Fugenformen im CISLEX. Dort sind auch die insgesamt 70 verschiedenen Fugentypen ausführlich beschrieben.

Die Fugenformen werden wie die Flexionsformen aufgrund einer Paradigmenbeschreibung in Prolog generiert.

2.4.4 Die interne Repräsentation des Lexikons

Das DKL-FLEX ist neben der Textversion in verschiedenen Implementierungen verfügbar: als relationale und deduktive Datenbank, als Lexikonautomat und als vereinfachte Attribut-Wert-Struktur in Form einer DBM-Datei[1].

Die relationale bzw. deduktivc Datenbank des CISLEX ist in Eclipse implementiert. Eclipse steht für *ECRC Common Logic Programming System* und ist ein am ECRC (München) entwickeltes Prolog, das das ebenfalls am ECRC entwickelte Datenbanksystem *Megalog* enthält. In Megalog ist es möglich, Datenbanken sowohl als deduk-

1. DBM ist ein einfaches Datenbankmanagementsystem unter Unix, das einen direkten Plattenzugriff erlaubt. Die Daten sind als einfache Schlüssel-Wert-Strukturen organisiert, die intern als Hashtable realisiert sind.

tive Datenbanken in Tupelform zu benutzen, als auch mit einem relationalen Schema daraufzuzugreifen. Diese Form dient in erster Linie der Verwaltung der Daten. Eine für die Verarbeitung großer Datenmengen wesentlich effizientere Implementierung liegt in Form einer in C-implementierten Baumstruktur vor. Die Implementierung der Baumstruktur selbst und der notwendigen Operationen auf dieser Baumstruktur basiert auf dem im französischen Rechtschreibkorrektursystem *zpell* angewendeten Verfahren (Zimmermann, 1993). Mit dieser Implementierung des CISLEX ist es möglich, 8000 Look-ups pro Sekunde (auf einer NextStation mit einem mit 33MHz getakteten 68040 Prozessor) auszuführen und eine Komprimierung um den Faktor 10 gegenüber der Textversion zu erreichen. Die momentan annähernd 1 000 000 Einträge der flektierten Version des EF+-Lexikons zusammen mit den Fugenformen (das EF+-Lexikon enthält zusätzlich zum DKL-EF auch Präfigierungen von Formen des DKL) lassen sich so in einer Datei mit 1,5 MB speichern. Damit ist der Beweis für die Paktikabilität und Effizienz von Vollformenlexika (vgl. die Diskussion in Abschnitt 1.3.4 auf Seite 16) geliefert.

2.5 Bestandsanalyse des DKL-EF

In diesem Abschnitt werden einige, auch im Hinblick auf die Lemmatisierung interessante, Untersuchungen zum Bestand des deutschen Kernlexikons der einfachen Formen in seiner Grundformversion (DKL-EF) und der Vollformenversion (DKL-FLEX) mit ihren Ergebnissen vorgestellt.

2.5.1 Der Bestand des DKL-EF nach Wortklassen

Das Grundformenlexikon DKL-EF enthält momentan 70 080 Einträge, die sich folgendermaßen auf die verschiedenen Wortklassen verteilen (die Werte in Klammern beziehen sich auf das EF+-Lexikon):

Nomen	37 337 (41 869)
Adjektive	10 084
Adjektivsuffixe	215
Adjektivische Partizipien	12 707
Verben	6 465 (18 209)
Adverbien	1 687
Interjektionen	309
Konjunktionen	96
Numeralia	50
Partikeln	120
Präpositionen	157
Präpdets	14
Determinatoren	50
Pronomen	50
Verbpartikeln	9
Sonderformen	11

Den Einträgen im DKL-EF entsprechen 536 068 Einträge im FLEX-Lexikon. Die flektierenden Wortklassen im einzelnen:

Nomen	90 724 (103 059)
Adjektive	151 680
Adjektivsuffixe	3 450
Adjektivische Partizipien	215 662
Verben	71 211 (235 163)
Numeralia	419
Determinatoren	181
Pronomen	92

2.5.2 Ambiguitäten

Für die Lemmatisierung und andere Anwendungen sind besonders die homographen Einträge interessant, da sie zu zum Teil unerwünschten Ambiguitäten führen. Ambiguitäten können auf den verschiedenen linguistischen Beschreibungsebenen Morphologie, Syntax und Semantik auftreten. Semantische Ambiguitäten, seien es unterschiedliche Lesarten oder homonyme Lexeme, werden im CISLEX nur dann unterschieden, wenn sie unterschiedliches morphologisches Verhalten haben. Syntaktische Ambiguitäten sind nur dann repräsentiert, wenn sie die Wortklasse betreffen, nicht aber beispielsweise unterschiedliche Subkategorisierungseigenschaften. Im Vollformenlexikon treten zudem Ambiguitäten auf, die daraus resultieren, daß es im deutschen Flexionssystem keine eindeutige Abbildung der Morphe auf die repräsentierten morphosyntaktischen Merkmale gibt, sondern ein Morph in der Regel eine Menge verschiedener Merkmalskombinationen realisieren kann.

2.5.2.1 Ambiguitäten im Grundformenlexikon

Die 70 080 verschiedenen Einträge im DKL-EF entsprechen tatsächlich 66 961 orthographisch verschiedenen Formen. Von diesen sind 64 273 nicht ambig und 2688 mindestens 2-fach ambig:

1. Ambiguitäten der EF-Grundformen:

Gesamtzahl der Grundformen:	66 961
Formen mit nur einer Kategorie:	64 273
Ambige Formen:	2 688

Ambiguitätsgrad der EF-Grundformen:

2-fach ambig:	2 404
3-fach ambig:	197
4-fach ambig:	70
5-fach ambig:	6
6-fach ambig:	11

Ein Teil dieser Ambiguitäten betrifft rein morphologische Ambiguitäten, d.h. Ambiguitäten, die lediglich darauf beruhen, daß ein Wort verschiedenen Flexionsklassen angehört oder (im Falle von Nomen) in verschiedenen Genera auftritt. Gerade bei den Nomen führt die Genusvarianz Neutrum-Maskulinum gekoppelt mit morphologischer Varianz, die in beiden Genera auftritt, zu der hohen Ambiguitätsrate 6. Um diese Faktoren einschätzen zu können, wurde dieselbe Untersuchung nur unter Beachtung der Wortklassen nochmals durchgeführt, was folgendes Ergebnis lieferte:

2. Wortartambiguitäten der EF-Grundformen:

Gesamtzahl der Grundformen:	66 961
Formen mit nur einer Wortart:	66 008
Wortart-ambige Formen:	953

Ambiguitätsgrad der EF-Grundformen bzgl. der Wortarten:

2-fach ambig:	880
3-fach ambig:	62
4-fach ambig:	11

Obwohl gerade im Bereich der Nomen, nicht aufgrund von Genus- und Deklinationsklassenunterschieden unmittelbar auf nur morphologische Varianz geschlossen werden kann (Fälle wie *Bank* oder *der Gang/die Gang* werden bei der zweiten Auswertung nicht unterschieden), so zeigt sich doch, daß die Ambiguitätsrate wesentlich abgenommen hat. Eine relativ hohe Wortartambiguität trifft man erwartungsgemäß im Funktionswortbereich an (wenn mehr als 2 Wortarten möglich sind, so ist mindestens eine davon eine Funktionswortart). Bei den 2-fachen Wortartambiguitäten dominiert jedoch mit einer Auftretenshäufigkeit von 342 die Kombination Nomen-Adjektiv, gefolgt von der Kombination Nomen-Verb (163 mal) und der Kombination Nomen-Interjektion (75 mal). Weitere Kombinationen sind mit abnehmender Häufigkeit: Nomen-Adverb, Adverb-Partikel, Adjektiv-Verb, Adverb-Konjunktion, Adjektiv-Partikel, Adverb-Präposition, Nomen-Präposition, Nomen-Pronomen.

2.5.2.2 Ambiguitäten im FLEX-Lexikon

Zunächst werden nur Ambiguitäten betrachtet, die sich auf der Ebene der Kodierung abspielen. Die Ambiguitäten, die sich aus der Polyfunktionalität der Flexionsmorphe ergeben, werden hier nicht beachtet, da sie unabhängig vom Inventar des EF-Lexikons nur von der morphologischen Klassifikation abhängen.

1. Ambiguitäten aller FLEX-Einträge, einschließlich Fugenformen:

Gesamtzahl der Formen[1]:	363 617
Formen mit nur einer Kategorie:	282 418
Ambige Formen:	81 199

Ambiguitätsgrad der flektierten Formen:

2-fach ambig:	68 331
3-fach ambig:	6 755
4-fach ambig:	4 453

1. Im Gegensatz zur Aufstellung in Abschnitt 2.5.1 auf Seite 73 werden hier nur orthographisch verschiedene Einträge gezählt.

5-fach ambig:	810
6-fach ambig:	444
7-fach ambig:	88
8-fach ambig	218
9-fach ambig	44
10-fach ambig:	27
11-fach ambig:	6
12-fach ambig:	42

Diese Statistik ist allerdings nicht sehr aussagekräftig, da bei den Nomen ein hoher Grad an Ambiguität durch die explizit kodierten Fugenformen verursacht wird, die hier nicht als Flexivambiguität behandelt wurden. Ein zweiter Faktor sind die Doppelklassifikationen der Partizipien als Adjektive. Läßt man diese Ambiguitäten außer acht, so ergibt sich folgendes Bild:

2. Ambiguitäten der FLEX-Einträge ohne Fugenformen und Ambiguitäten zwischen verbalen Partizipien und adjektivischen Partizipien:

Ambige Formen:	48 426
2-fach ambig:	42 738
3-fach ambig	4 267
4-fach ambig	1 106
5-fach ambig:	151
6-fach ambig:	132
8-fach ambig:	32

Es zeigt sich also, daß im ersten Test die Ergebnisse durch in die in der Realität nicht auftretenden Ambiguitäten mit Fugenformen und die sowieso nur durch syntaktische Analyse auflösbaren Ambiguitäten zwischen den verbalen Partizipien und den entsprechenden Positivformen der daraus abgeleiteten Adjektive verzerrt wurden. Wie bereits im Falle der EF-Grundformen rühren die relativ hohen Ambiguitätsraten in beiden Fällen wieder von der Ausdistribuierung von Genus- und Deklinationsklassenvarianten bei Nomen her.

2.5.3 Einfache Formen als echter Bestandteil anderer einfacher Formen

Für die Segmentierung komplexer Formen sind diejenigen einfachen Formen besonders kritisch, die selbst wieder Bestandteil anderer einfacher Formen sind, bzw. die einfachen Formen, die andere einfache Formen enthalten. Es gibt 3 Fälle zu unterscheiden: (i) einfache Formen, die als Anfangsglied anderer einfacher Formen auftreten bzw. einfache Formen, die ein Anfangsglied haben, das selbst wieder eine einfache Form ist, (ii) einfache Formen, die als Endglied anderer einfacher Formen auftreten

bzw. einfache Formen, die ein Endglied haben, das selbst wieder eine einfache Form ist, und (iii) einfache Formen, die nicht randständig Bestandteil anderer einfacher Formen sind, bzw. einfache Formen, die nicht randständig eine andere einfache Form enthalten.

2.5.3.1 Einfache Formen als Präfix[1] einfacher Formen

Wörter, deren Wortanfang mit anderen Wörtern übereinstimmt, stellen vor allem für Segmentierungsalgorithmen, die eine zu analysierende Form von links nach rechts abarbeiten, ein Problem dar. Da in diesem Fall die Flexion nur eine untergeordnete Rolle spielt, sollen hier die Ergebnisse einer Untersuchung, die nur auf Grundformen basiert, ausreichen:

Untersuchung der einfachen Formen als Präfixe:

EFs, die als EF-Präfix vorkommen:	18 024
EFs, die ein EF-Präfix haben:	44 221

2.5.3.2 Einfache Formen als Suffix einfacher Formen

Am interessantesten ist der Suffix-Fall[2]. Aus diesem Grund wurde die Untersuchung sowohl für die Grundformen als auch für die Vollformen durchgeführt. In beiden Fällen wurde zusätzlich getestet, ob das EF-Suffix dieselbe Kategorie hat oder nicht. Fälle, bei denen das EF-Suffix dieselbe Kategorie hat, sind unproblematischer, da sie zwar unter Umständen zu falschen Segmentierungen führen, die Kategorie davon jedoch nicht betroffen ist.

Untersuchung der einfachen Grundformen auf Suffixe:

EFs, die als EF-Suffix vorkommen:	6 349 (9,5 %)
davon mit gleicher Kategorie:	4 663 (72,9 %)
EFs, die ein EF-Suffix haben:	33 173 (49,8 %)
davon mit gleicher Kategorie:	10 757 (32,4 %)

Das bedeutet, ungefähr ein Zehntel der EF-Grundformen kommt selbst wieder als Suffix in EF-Grundformen vor und über die Hälfte aller EF-Grundformen enthalten selbst wieder ein EF-Suffix, und immerhin in rund zwei Dritteln dieser Fälle hat das EF-Suffix eine andere Kategorie. Vergleicht man diese Ergebnisse mit denen aus 2.5.3.1

1. Die Begriffe *Präfix* und *Suffix* werden hier in einem rein formalen Sinn gebraucht und bezeichnen lediglich den Anfangs- bzw. Endstring eines gegebenen Strings.

2. Auf den ersten Blick mag es so aussehen, als ob dies ein Wiederspruch zu der Definition von *einfacher Form* ist (vgl. Abschnitt 2.1.3 auf Seite 30ff). Der Begriff Suffix bezieht sich hier jedoch auf beliebige Endstrings, während in der Definition für *einfache Form* a) eine Zerlegung, die Morphemgrenzen berücksichtigt, gefordert ist und b) nur Suffixe mit derselben morphologischen Kategorie ausgeschlossen werden. So sind *Streife*n und *Reifen* ganz klar einfache Formen, wo allerdings *Reifen* ein (formales) Suffix von *Streifen* ist.

bezüglich der Präfix-Eigenschaft, so stellt man fest, daß der Anteil der einfachen Formen, die ein EF-Präfix enthalten, wesentlich größer ist, als der Anteil derjenigen, die ein EF-Suffix enthalten (66,3% gegenüber 49,8%). Für einen Segmentierungsalgorithmus, der auf dem DKL basiert, ist also ein Vorgehen von rechts nach links vorzuziehen, da hier im ersten Schritt weniger lexikonbedingte Ambiguitäten zu erwarten sind.

2.5.4 Homonymie zwischen einfachen und komplexen Formen

Einen Sonderfall zu den in 2.5.3 behandelten Fälle stellt die Homonymie zwischen einfachen Formen und Komposita dar. Das ist immer dann der Fall, wenn sich eine einfache Form zerlegen läßt in eine Folge möglicher Vorderglieder von Komposita und einem Hinterglied[1], die alle aus dem Bereich der einfachen Formen stammen. Dies scheint auf den ersten Blick im Widerspruch zu der Definition von *einfacher Form* zu stehen, aber man muß beachten, daß bei der Definition die Einschränkung gemacht wurde, daß Zerlegungen *sinnvoll* sein müssen, also nur an tatsächlichen Morphemgrenzen erfolgen dürfen. Einige Beispiele einfacher Formen, die homonym zu Komposita sind, werden diesen Unterschied klarstellen: *Tangente - Tang + Ente, Proletarier - Prolet + Arier, Pomade - Po + Made,* usw.

Untersuchung der einfachen Grundformen auf Homonymie mit Komposita

Anzahl der EF-Grundformen:	66 961
Anzahl der nicht-zerlegbaren EF-Grundformen:	59 089
Anzahl der zerlegbaren EF-Grundformen:	7 872
davon auf mehr als eine Weise zerlegbar:	3 155

Erfreulicherweise hat also nur ein relativ geringer Anteil der EF-Grundformen diese Eigenschaft, die in jedem Fall zu Ambiguitäten bei der Textbearbeitung führt. Die relativ geringe Anzahl ermöglicht es, im Gegensatz zu den einfachen Formen, die lediglich ein EF-Suffix haben, diese Problemfälle in einer Ausnahmeliste aufzuführen.

1. Die genauen Bedingungen, denen Vorderglieder und Hinterglieder von Komposita genügen müssen, werden ebenso wie der Zerlegungsalgorithmus in Abschnitt 3.3.1.2 auf Seite 100 ausführlich dargestellt.

3 Automatische Lemmatisierung

In diesem Kapitel soll dargestellt werden, wie auf der Basis des in Kapitel 2 beschriebenen CISLEX-Lexikons die Lemmatisierung von beliebigem Fließtext geschieht. Im ersten Abschnitt werden einige grundlegende Fragen zur Lemmatisierung diskutiert. Im weiteren wird dann der Lemmatisierungsvorgang beschrieben.

Der Lemmatisierungsalgorithmus ist so ausgelegt, daß lediglich eine Segmentierung in Sätze in einem Vorverarbeitungsschritt vor der eigentlichen Lemmatisierung vorgenommen werden muß. Hierfür wurde das Satzende-Erkennungs-Programm von Schicht (vgl. Schicht 1994b) verwendet.[1] Der Lemmatisierungsalgorithmus unterscheidet sich von herkömmlichen Lemmatisierungsverfahren vor allem dadurch, daß erstens zur Identifizierung der Wortformen ein umfassendes Lexikonsystem verwendet wird und damit die Fehler beim Look-up auf das Auftreten lexikalischer Ambiguitäten beschränkt werden können, und zweitens durch eine umfassende Behandlung von Sonder- und Mischformen (numerische Ausdrücke, Ausdrücke, die Sonderzeichen enthalten usw.), die von herkömmlichen Systemen weitgehend ignoriert werden. Eine Ausnahme für den Bereich des Wortarten-Tagging stellt die Arbeit von Hippelein (1994) dar. Es geht ihr jedoch nur darum, Sonderformen mit Hilfe von regulären Ausdrücken zu erkennen und ihnen dann den Tag, der am wahrscheinlichsten ist, zuzuweisen. Eine Analyse der einzelnen Komponenten von Sonderformen, die etwa im Hinblick auf Anwendungen wie Automatische Indizierung oder Information Retrieval von Bedeutung ist, findet jedoch nicht statt. Weiterhin setzt das Verfahren von Hippelein bereits vorbearbeitete Korpora voraus, in denen kritische Formen, Eigennamen, spezielle Komposita usw. bereits markiert sind. Im Gegensatz dazu, erfordert die Lemmatisierung die Analyse auf eine “Grundform”, die unter anderem auch die Strukturinformation der Sonderformen enthält. Das oberste Ziel ist auch hier, **jedes** Element, das in einem Text zwischen zwei Blanks vorkommt, **korrekt** zu identifizieren, wobei im folgenden noch zu spezifizieren ist, was unter *korrekter Identifikation* zu verstehen ist.

Der hier vorgestellte Lemmatisierungsalgorithmus wurde anhand eines Testkorpus entwickelt, das aus den Ausgaben der Süddeutschen Zeitung vom Januar 1994 besteht. Darüberhinaus wurde ein Sonderformenkorpus erstellt, das aus mehreren Jahrgängen der Süddeutschen Zeitung sowie aus einem umfangreichen literarischen Text, Musils “Mann ohne Eigenschaften”, extrahiert wurde. Die Auswahl des Testkorpus erfolgte zum einen im Hinblick auf die Verfügbarkeit und zum anderen im Hinblick auf die auftretenden Phänomene, vor allem im Bereich der Sonder- und Mischformen.[2] Die Texte wurden unbearbeitet übernommen, lediglich Steuercodes wurden entfernt. Im

1. Es ist prinzipiell kein Problem, das Satzendeerkennungsprogramm in den Lemmatisierer einzubinden, wenn eine minimale Fehlerkennungsrate akzeptiert wird. Eine vollständig korrekte Erkennung liefert das Programm allerdings nur in einem interaktiven Modus, so daß dieses Vorgehen hier vorgezogen wurde.

2. Stichprobentests ergaben, daß bei Betrachtung anderer Zeitungstexte oder anderer Textsorten, wie z. B. Romanen, keine weiteren, wesentlich anderen Phänomene auftreten.

folgenden beziehen sich alle Aussagen über das Auftreten von speziellen Zeichen oder Elementen auf das Testkorpus und das Sonderformen-Korpus. Es ist möglich, daß in anderen, ausgefallenen Textsorten, die hier unmöglich alle berücksichtigt werden konnten, andere Sonder- und Mischformen auftreten. Der Lemmatisierungsalgorithmus ist jedoch so modular konzipiert, daß er problemlos um weitere Phänomene erweitert werden kann.

3.1 Grundlagen der Lemmatisierung

3.1.1 Auf was wird lemmatisiert?

Die Frage, was eigentlich ein Lemma ist, wird in der Literatur in zweierlei Hinsicht diskutiert: zum einen geht es um diese Frage im Kontext traditioneller Lexikographie (hier ist der Begriff *Lemma* im Zusammenhang mit Lexikonerstellung relevant und wird in der Regel synonym zu Begriffen wie *Headword* oder *Stichwort* verwendet), zum anderen wird diese Frage bei konkreten Lemmatisierungen (wie beispielsweise bei der Erstellung von Frequenzwörterbüchern) und vor allem bei der automatischen Lemmatisierung unter praktischen Gesichtspunkten erörtert. Da es in dieser Arbeit um automatische Lemmatisierung geht, soll hier nur ganz kurz anhand der Arbeiten von Wiegand (exemplarisch hier Wiegand 1983)[1] auf den traditionell-lexikographischen Lemma-Begriff eingegangen werden, insofern die dort gemachten Definitionen auch für die Textlemmatisierung relevant sind.

3.1.1.1 Der Begriff *Lemma* bei Wiegand

Nach Wiegand ist das *Lemma* ein Ausdruck, der in einem Lexikon den lexikographischen Ordnungsprinzipien unterworfen ist. Das Lemma beschränkt sich bei Wiegand ausschließlich auf die im Lexikon repräsentierte Wortform, weitere Informationen gehören nicht zum Lemma: “Mit dem Lemma alleine kann jedoch ein grammatisches Paradigma nicht (re-)präsentiert werden.” (Wiegand 1983, S. 444). Unter einem (lexikographischen) Lemmatisierungsproblem versteht Wiegand die Aufgabe “Wie kommt man von einer gegebenen Menge von Belegformen zu einer Form, die das Lemmazeichen graphematisch und damit auch graphisch repräsentiert?” (Wiegand 1983, S. 445). Das Problem der Lemmatisierung besteht bei Wiegand aus vier Teilen:

- der Lemmatisierungsaufgabe,
- der Lemmatisierungsvorschrift,
- dem Lemmatisierungsverfahren und
- dem Lemmatisierungsalgorithmus.

1. Eine weitere ausführliche, historisch orientierte Darstellung zu diesem Thema findet sich in Wolski (1989).

Diese Teile sind bei Wiegand wie folgt definiert:

1. Eine *Lemmatisierungsaufgabe* ist eine lexikographische Aufgabe, die darin besteht, entweder eine begründete Lemmatisierungsvorschrift oder ein begründetes Lemmatisierungsverfahren anzugeben.
2. Eine *Lemmatisierungsvorschrift* ist eine sprachtheoretisch begründete Vorschrift, die angibt, wie von einer Menge von Belegformen zu einer sprachlichen Form überzugehen ist, die das Lemmazeichen graphematisch (und damit auch graphisch) repräsentiert.
3. Ein *Lemmatisierungsverfahren* ist ein sprachtheoretisch begründetes, lexikographisches Verfahren (X), das aus einer Menge von geordnet anzuwendenden Lemmatisierungsvorschriften besteht.
4. Ein *Lemmatisierungalgorithmus* ist ein Lemmatisierungsverfahren, das so formal aufgeschrieben ist, daß es maschinell implementierbar ist.

(nach Wiegand 1983 S. 451f)

Um die Alternativen zum Wiegandschen Lemmabegriff und um die Lemmatisierungsvorschriften, die in der Praxis angewendet werden, soll es im folgenden gehen.

3.1.1.2 Weitere Lemmabegriffe

Nach den theoretischen Betrachtungen bei Wiegand, die sich vorwiegend auf die lexikographische Arbeit beziehen, sollen nun Lemmabegriffe, die aus dem Umfeld der Textlemmatisierung (sowohl manuell, als auch automatisch) stammen, betrachtet werden.

3.1.1.2.1 Der Lemmabegriff bei Rosengren: Frequenzwörterbuch der deutschen Zeitungssprache

Für die Arbeiten zum *Frequenzwörterbuch der deutschen Zeitungssprache* (Rosengren 1972 und Rosengren 1969) legt Rosengren einen Lemmabegriff zugrunde, der sich an ein in Allén (1967) beschriebenes Lemmatisierungsverfahren anlehnt. Allén berücksichtigt bei der Lemmatisierung nur morphologische und syntaktische Information: zu einem Lemma gehören alle Wortformen, die derselben Flexionsreihe und derselben Wortklasse angehören. Innerhalb einer Flexionsreihe kann für Allén lediglich Varianz von Formen, nicht aber Opposition von Formen auftreten. Das heißt, das Auftreten verschiedener Wortformen mit denselben morphosyntaktischen Merkmalen, die nicht in allen Kontexten ohne Denotatsänderung gegeneinander austauschbar sind, führt zu verschiedenen Flexionsreihen und damit auch zu verschiedenen Lemmata. Rosengren beschränkt sich auf den morphologischen Aspekt dieser Definition und betrachtet einzig die Flexionsreihen als grundlegend für die Ansetzung eines Lemmas. Wie Allén läßt auch Rosengren nur Variation, nicht aber Opposition sowohl der Wortformen als auch des Genus innerhalb einer Flexionsreihe zu. Die Wortklassenzugehörigkeit spielt bei ihrer Lemmatisierung nur insofern eine Rolle, als das Wortartensystem so beschaf-

fen sein muß, daß keine zwei Wortklassen Flexionsreihen desselben Typs bilden können. Wortartambiguitäten im Bereich der Funktionswörter, die keine Flexionsreihen bilden, werden durch diese Lemmatisierung nicht repräsentiert.

3.1.1.2.2 Der Lemmabegriff beim Saarbrücker Lemmatisierungsverfahren SALEM

Beim Saarbrücker Verfahren zur automatischen Lemmatisierung (Weber 1976) wird von einem Lemmabegriff ausgegangen, wie er etwa in Dietrich/Klein (1974) vorgestellt wird. Bei Dietrich/Klein besteht die Lemmainformation aus einem Paar (G,I), wobei G eine Menge von Wortlauten ist und I die Informationen zu G enthält. I selbst ist ein Tripel (S,R,P) bestehend aus den semantischen Merkmalen S, den syntaktischen Merkmalen R und pragmatischen Merkmalen P. S und P sind für alle Formen eines Lemmas gleich. R enthält die morphosyntaktischen Merkmale, die durch die jeweilige Wortform repräsentiert werden. Dieser Lemmabegriff setzt natürlich ein Lexikon voraus, das die entsprechenden Informationen S und P enthält. Damit ist dieser Lemmabegriff von vornherein nicht für regelbasierte Lemmatisierungsverfahren geeignet.

3.1.1.2.3 Der Lemmabegriff bei LEMMA

Für das LEMMA-System (Willée 1979), als in erster Linie regelbasiertes System, scheidet der Lemmabegriff von Dietrich/Klein aus den bereits genannten Gründen aus. Deshalb wurde auch in diesem System der Ansatz von Allén verfolgt, hier allerdings in einer neueren Version (Allén 1978). In diesem neueren Ansatz unterscheidet Allén zwischen *Lemma* und *Lexem*. Insgesamt sind im Zusammenhang mit Wortformen drei Typen von Information relevant:

1. Information über den Ausdruck (Wortlaut)
2. Information über die Funktion (Morphologie und Syntax)
3. Information über die Semantik

Als Schlüsselwort wird 1) bezeichnet, das Lemma wird von 1) und 2) gebildet, 3) schließlich ist das Lexem. Die Informationen über die Funktion entsprechen den bereits in Abschnitt 3.1.1.2.1 beschriebenen Informationen über Wortklasse und Flexionsreihe.

Ein großer Vorteil des Lemmabegriffs von Allén im Kontext der automatischen Lemmatisierung ist die Beschränkung auf leicht objektivierbare Informationen. Auf eine Behandlung von Polysemie und Homographie kann dadurch weitgehend verzichtet werden.

3.1.1.3 Das CISLEX-Lemma

In den vorangehenden Abschnitten wurde klar, daß die Wahl eines geeigneten Lemmabegriffs für automatische Lemmatisierung in erster Linie auch von der verfügbaren lexikalischen Information abhängt. Das hier vorgestellte Lemmatisierungsverfahren

beruht wesentlich auf dem CISLEX-Lexikonsystem. Dementsprechend kann ein Lemma nur Information, die aus dem CISLEX direkt oder mit Hilfe allgemeiner sprachlicher Gesetzmäßigkeiten abgeleitet werden kann, enthalten. Andererseits sollte es aber auch soviel Information wie möglich enthalten. Damit ist klar, was das Lemma für einfache Formen ist: der Eintrag der entsprechenden Wortform im DKL-Flex.[1] Ein Lemma ist also ein Tripel der Form:

Lemma = <Lemmaform, Lemmaklasse, morphosyntaktische Merkmale>

Die im Flex-Lexikon enthaltene Grundform ist die Lemmaform. Die syntaktische Klasse zusammen mit dem morphologischen Typ bilden die Lemmaklasse und die morpho-syntaktischen Merkmale sind analog zum Lexikon als Liste aller möglichen, durch die betreffende Form repräsentierbaren Merkmalskombinationen definiert. Damit unterscheidet sich die Lemmadefinition von der in Wiegand (1983) vorgestellten durch einen größeren Informationsgehalt: anhand des Lemmas lassen sich alle Formen des morphologischen Paradigmas rekonstruieren. Im Gegensatz zu Rosengren (1969, 1972) ist das Lemma jedoch nicht als Repräsentant aller Formen des Paradigmas zu verstehen, da durch die morphosyntaktischen Merkmale die spezielle Form eindeutig bestimmt ist. Im Hinblick auf die anderen vorgestellten Lemmadefinitionen verhält sich die hier verwendete eher neutral, indem die Beschränkung auf die morphologischen und groben Distributionseigenschaften pragmatisch durch das zugrundeliegende Lexikon bedingt ist und eine Erweiterung um andere Informationstypen von einer Erweiterung des DKL abhängt. Die Frage nach den Flexionsreihen ist bereits durch die Kodierung im CISLEX beantwortet. Im Unterschied zu Alléns und Rosengrens Begriff der Flexionsreihe, führen im CISLEX nur systematische Variationen innerhalb eines Paradigmas zu einer morphologischen Klasse. Nicht systematische Varianz der Wortformen, auch wenn diese in allen Kontexten austauschbar sind, wurden durch Mehrfachklassifikation behandelt.

Da die komplexen Formen auf der Basis der einfachen Formen kodiert werden, überträgt sich die Definition direkt auf komplexe Formen. Zusätzlich enthält die Lemmaform hier noch die entsprechende Segmentierung. Die Frage nach dem Lemma bei Formen, die nicht in den Bereich des DKL fallen, wie Namen, Zahlen, Abkürzungen und andere Sonderformen, wird bei der Behandlung dieser Einheiten in Abschnitt 3.4 auf Seite 109ff jeweils separat diskutiert.

Die oben angeführte Definition von Lemma ist in bezug auf einige Auswahl- und Kodierkriterien des DKL problematisch:

- die Ableitung bestimmter Kategorien durch Regeln, die auf Einträgen des DKL operieren

1. Ich gehe hier von dem in Abschnitt 2.1 auf Seite 28 beschriebenen, auf morphosyntaktische Informationen eingeschränkten DKL aus. Die in Arbeit befindlichen Erweiterungen um syntaktische und semantische Kategorien sind hier nicht berücksichtigt. Diese Merkmale sind aber nach vollständiger Kodierung dem Lemma ebenfalls anzufügen.

- die Kodierung orthographischer Varianten
- die Kodierung ambiger Grundformen

Auf die dadurch implizierten Probleme soll im folgenden eingegangen werden.

3.1.1.3.1 Lemmatisierung systematischer Ableitungen und Konversionen

Für die Lemmatisierung von Wörtern, die durch eine generelle Regel von anderen Lexikoneinträgen abgeleitet werden können, bieten sich verschiedene Alternativen an:

1. Man kann diese Wörter auf ihre Basis, von der sie ableitbar sind, lemmatisieren. Die Lemmaform ist dann die Grundform der Basis (auch wenn diese einer anderen Wortart angehört), die Lemmaklasse die Klasse der Basis und die morpho-syntaktischen Merkmale sind ebenfalls die für die Klasse der Basis spezifischen Merkmale.
2. Das Lexikon kann für die Zwecke der Lemmatisierung um Einträge für die entsprechenden Wörter erweitert werden. Die Regeln, die die systematische Ableitung dieser Wörter beschreiben, wären dann sogenannte “once-and-only”-Regeln, die nur zur automatischen Lexikonerweiterung angewendet werden. Die Lemmaklasse für diese Wörter ist dann die Grundform der Wortklasse der Ableitung, Lemmaklasse und morphosyntaktische Merkmale sind ebenfalls die der Ableitung.
3. Das Lemma der Ableitung wird ähnlich wie in 2) durch eine Regel aus den Lexikoneinträgen abgeleitet, ohne daß die Ableitung selbst Element des Lexikons wird.

Das erste Verfahren hat den Vorteil, daß keine unnötige Aufblähung des Lexikons notwendig ist und das Lemma nur Informationen enthält, die tatsächlich auch im Lexikon sind. Es führt allerdings auch häufig zu abwegigen Lemmatisierungen, beispielsweise wenn partizipiale Adjektive auf den entsprechenden Verbinfinitiv lemmatisiert werden. Hier liefern das zweite und dritte Verfahren eindeutig befriedigendere Lösungen, allerdings beim zweiten Verfahren auf Kosten einer erheblichen Redundanz im Lexikon. Beim dritten Verfahren ist, wie bei allen regelbasierten Verfahren, die Gefahr von Fehlanalysen und Generierung nicht existenter Lemmata gegeben. Diese Gefahr besteht prinzipiell auch beim zweiten Verfahren, hier können jedoch kritische Fälle im Lexikon manuell nacheditiert werden. Es gibt für dieses Problem also keine universelle Lösung und man wird in der Regel auch nicht für alle Ableitungstypen dasselbe Verfahren wählen. Welches dieser Verfahren gewählt wird, hängt von mehreren Faktoren ab: zum einen von der Anzahl der betroffenen Elemente, von der Wahrscheinlichkeit von Fehlanalysen bei einer Ableitung des Lemmas durch Regeln und nicht zuletzt auch von der intendierten Anwendung. Für manche Anwendungen (zum Beispiel die automatische Indizierung) kann es durchaus sinnvoll sein, wie im obigen Beispiel etwa die partizipialen Adjektive auf das entsprechende Verb zu lemmatisieren, während dies für andere Anwendungen keinerlei Sinn macht. Hieraus darf man jedoch nicht den Schluß ziehen, daß man für jede Anwendung ein spezielles Lemmatisierungsverfahren entwickeln sollte. Vielmehr sollte das Lemmatisierungsverfahren so beschaffen sein, daß aus den gefundenen Lemmata je nach Anwendung die gewünschte Informa-

tion extrahiert werden kann. Das Lemmatisierungsverfahren muß den größtmöglichen gemeinsamen Nenner darstellen, und alle notwendige Information repräsentieren.

Welches Verfahren bei den verschiedenen Ableitungstypen aufgrund dieser Kriterien gewählt wurde, wird in den nächsten Abschnitten erläutert.

3.1.1.3.1.1 Partizipiale Adjektive

Sowohl das Partizip Präsens als auch das Partizip Perfekt der Verben kann (mit mehr oder weniger großer Akzeptabilität) auch mit entsprechender Deklination in adjektivischer Funktion gebraucht werden. Je nach Grad der Lexikalisierung ist der prädikative Gebrauch neben dem attributiven Gebrauch auch möglich (*das Aussehen wurde verändert - das veränderte Aussehen - das Aussehen ist verändert*), wobei tendentiell in beiden Positionen die vom Partizip Präsens abgeleiteten Formen akzeptabler sind.[1] Rosengren (1972) lemmatisiert beispielsweise alle Präsenspartizipien und deren adjektivisch flektierte Formen als Adjektive, im Falle des Partizip Perfekt werden nur solche von intransitiven Verben, die als Auxiliar “sein” selegieren, als Adjektive lemmatisiert. Alle anderen nicht adjektivisch flektierten Partizip-Perfekt-Formen werden als Verb auf den entsprechenden Infinitiv lemmatisiert. In LEMMA2 (Schulze & Willée 1983) und beim Saarbrücker Lemmatisierungsverfahren (Weber 1976) werden die Partizipien schon grundsätzlich als separate Wortart behandelt, so daß sich bei unflektierten Partizipien die Frage nach der Grundform, auf die lemmatisiert werden soll, nicht stellt.

Die Annahme einer speziellen Wortklasse *Partizip* erscheint mir sowohl im Hinblick auf die in Abschnitt 2.2 auf Seite 34 aufgeführten Kriterien zur Wortartenklassifizierung im CISLEX als auch im Hinblick auf die Lemmatisierung nicht adäquat. Die Partizipien sind Elemente des Verbparadigmas. Daß sie in unflektierter Form sowohl in verbaler Funktion als auch in adjektivischer Funktion auftreten können, ist meiner Meinung nach kein hinreichendes Kriterium für die Annahme einer separaten Wortklasse. Die Annahme einer solchen Wortklasse mag zwar die Lemmatisierung vereinfachen, entspricht jedoch nicht der eigentlichen Distribution. In seiner verbalen Funktion verhält sich das Partizip (etwa hinsichtlich Selektionsrestriktionen, Stellung, usw.) wie alle anderen Formen des Verbparadigmas. Deshalb wird man beispielsweise im Bereich der automatischen Indizierung solche Vorkommen nicht von denen anderer Verbformen unterscheiden wollen und nach dem gemeinsamen Infinitiv indizieren. Dazu kommt, daß die “zweite Funktion” als Adjektiv, die ja bei der unflektierten Form eine prädikative Verwendung sein müßte, nicht generell möglich ist. Damit bleiben für die Lemmatisierung der Partizipien nur noch die beiden Möglichkeiten der Lemmatisierung auf den verbalen Infinitiv oder die Lemmatisierung auf die Lemmaform Partizip und die Lemmaklasse Adjektiv.

1. Dies kann natürlich daran liegen, daß es eine syntaktische Regel der Gerundivbildung gibt. Diese Regel überführt eine Verbalphrase in eine (meist das Subjekt des Verbs modifizierende) Adjektivphrase, deren Kopf die adjektivische Form des Partizip Präsens des Verbs ist.

Hier spricht schon aus der Sicht des CISLEX einiges dafür, die Partizipien trotz ihrer relativ großen Anzahl über eine Regel als Adjektive ins Lexikon zu integrieren. Zwar ist das Flexionsverhalten dieser partizipialen Adjektive vorhersagbar, nicht aber die Möglichkeiten der Komparation und der prädikativen Verwendung. Für die Lemmatisierung kann daher von einem eigenen (durch Regeln erzeugten) Lexikoneintrag ausgegangen werden.[1] Bei den unflektierten Formen muß dann gegebenenfalls noch zwischen dem Verb- und dem Adjektivlemma disambiguiert werden. Ein weiterer Vorteil der Kodierung als Adjektiv ergibt sich im Hinblick auf die Behandlung von Nomen mit adjektivischer Flexion (siehe “Nominalisierte Adjektive - Nomen mit adjektivischer Deklination” auf Seite 87). Partizipiale Adjektive treten sehr häufig auch als stark lexikalisierte deadjektivische Nomen auf (*Verwandte, Angestellte, Behinderte,* usw.). Die Kodierung der Partizipien als Adjektiv erlaubt die direkte Ableitung dieser Nomen.

Der Vollständigkeit halber sei hier noch auf die sogenannten Pseudopartizipien hingewiesen. Unter Pseudopartizipien versteht man Adjektive, die der Form nach den Partizipien entsprechen, zu denen jedoch keine verbale Basis im Lexikon existiert (*beknackt, bedeppert, berühmt,* usw.). Diese Pseudopartizipien sind von Anfang an nur als normale Adjektive im Lexikon kodiert und bereiten hier keine Probleme.

3.1.1.3.1.2 Nomen in anderer Funktion

Es gibt eine kleine Menge von Nomen, die auch in anderen syntaktischen Funktionen, z.B. als Präposition (*angesichts*, *teils*, usw.) oder in adverbialer Funktion auftreten können. Bei den Fällen, in denen Nomen (in der Regel in der Genitiv Singular Form) als Präposition auftreten, handelt es sich eindeutig um Idiosynkrasien, die im Lexikon kodiert sein müssen. Eine Disambiguierung kann dann aufgrund der Groß- bzw. Kleinschreibung erfolgen. Anders bei den Tageszeiten, die, wenn es sich um bestimmte Daten handelt, kleingeschrieben vorkommen. Hier wird eine spezielle Liste derjenigen Nomen geführt, die kleingeschrieben auftreten können.[2]

3.1.1.3.1.3 Unflektierte Adjektive in adverbialer Verwendung

Ein Teil der Adjektive kann in der unflektierten Form auch adverbial gebraucht werden. Es stellt sich bei der Lemmatisierung hier also die Frage, ob auf die Lemmaklasse Adjektiv oder Adverb lemmatisiert werden soll. Da die Lemmaform in beiden Fällen identisch ist, ist diese Frage jedoch nicht von großer Relevanz. Der Einfachheit halber werden alle Vorkommen als Adjektiv lemmatisiert. Falls für bestimmte Anwendungen eine Unterscheidung zwischen prädikativer und adverbialer Funktion notwendig ist,

1. Falls eine Lemmatisierung der partizipialen Adjektive auf den verbalen Infinitiv gewünscht wird, muß die Adjektivgrundform als Partizip-Präsens oder Partizip-Perfekt nochmals im Lexikon nachgeschlagen werden, um den entsprechenden Infinitv zu erhalten.

2. Diese Liste ist unabhängig von solchen Fällen für einen Teil der quantifizierenden Nomen (siehe Abschnitt 2.2.1 auf Seite 36) wie *bißchen* oder *paar* notwendig.

kann dann mit einem geeigneten Disambiguierungsverfahren das Lemma entsprechend verfeinert werden.

3.1.1.3.1.4 Nominalisierte Adjektive - Nomen mit adjektivischer Deklination

Nominalisierte Adjektive und Nomen mit adjektivischer Deklination unterscheiden sich in erster Linie durch den Lexikalisierungsgrad. Prinzipiell kann jedes Adjektiv nominal verwendet werden, wenn die Eigenschaft als solche bzw. ein Träger der Eigenschaft bezeichnet werden soll (*das Schöne, der Gute,* usw.). Im Gegensatz dazu gibt es lexikalisierte Ableitungen, die sich (bis auf die Deklination) wie Nomen verhalten (*Verwandte, Angestellte,* usw.). Diese Lexeme werden als Nomen empfunden, was sich vor allem daran zeigt, daß sie als Kopf von nominalen Komposita auftreten, die als Adjektiv ungebräuchlich sind (*Verwaltungsangestellter, Firmenangehöriger,* usw.). Spezielle Fälle wie *Beamter*, wo überhaupt keine adjektivische Basis existiert, müssen natürlich in jedem Fall als spezielle Nomenklasse ins Lexikon. Zwar wäre auch für die stark lexikalisierten Fälle ein Lexikoneintrag wünschenswert, doch der Übergang zwischen den nominalisierten Adjektiven und den Nomen mit adjektivischer Deklination ist fließend, eine Abgrenzung ist in jedem Fall problematisch. Betrachtet man zudem noch die relativ geringe Anzahl dieser stark lexikalisierten Fälle im Vergleich zu den anderen Adjektiven, so erscheint es unplausibel aufgrund dieser geringen Anzahl, das Lexikon um Nomeneinträge für alle Adjektive zu erweitern, da sich außer der Lemmaklasse nichts ändert. Die Lemmaklasse *adjektivisches Nomen* wird deshalb durch eine Regel aufgrund der Groß- bzw. Kleinschreibung von Adjektiven hergeleitet.

3.1.1.3.1.5 Pronomen und Artikel

Wie in Abschnitt 2.2.4 auf Seite 38 bereits dargestellt, wird im CISLEX davon ausgegangen, daß jeder Determinator auch pronominal verwendet werden kann. Lediglich pronominale Formen von Determinatoren, die nicht zum Determinatorparadigma gehören, sind explizit kodiert. Im Hinblick auf die geringe Anzahl dieser Elemente, wäre für die Zwecke der Lemmatisierung auch die Mehrfachklassifikation denkbar. Da der gesamte Bereich der Determinatoren und Pronomen bislang noch weitgehend unstrukturiert ist[1], erscheint jedoch hier eine Unterscheidung unnütz, so daß für die Lemmatisierung von einer einheitlichen Klasse *DET bzw. PRO* ausgegangen wird.

3.1.1.3.1.6 Nominalisierte Infinitive

Die nominalisierten Infinitive werden aufgrund ihrer hohen Regularität nicht im Lexikon aufgeführt. Erstens sind alle Verbinfinitive nominalisiert verwendbar und zweitens ist die Form der abgeleiteten Nomen vollständig prädiktabel. Nominalisierte Infinitive werden über eine Regel von den entsprechenden Verbeinträgen abgeleitet.

1. Eine feinere Subklassifizierung nach syntaktischen Kriterien ist Thema weiterer Untersuchungen.

3.1.1.3.1.7 Verben mit abtrennbarer Partikel

Ein weiteres Problem stellt die Lemmatisierung von Verben mit abtrennbarer Partikel dar. Diese Verben werden in Verberst- und Verbletztstellung zusammengeschrieben, bei Verbzweitstellung mit nicht-zusammengesetzten Tempusformen jedoch bildet die in diesem Fall abgetrennte Verbpartikel die rechte Satzklammer. In diesem Fall sollte es auf jeden Fall das Ziel sein, finites Verb und abgetrennte Verbpartikel zu einer (in diesem Fall diskontinuierlichen) Lemmaform zusammenzuführen. Zu diesem Zweck enthält das CISLEX eine Liste aller einfachen Verben zusammen mit den bei jedem Verb gebräuchlichen Verbpartikeln. Eine eindeutige Zuordnung ist jedoch häufig erst nach einer umfassenden Kontextanalyse möglich, wie dies beispielsweise in Weber (1976) beschrieben wird.

3.1.2 Überblick über Lemmatisierungsmethoden

Relativ unabhängig von der Frage, auf was lemmatisiert wird, ist die Frage nach dem Wie. Die Lemmatisierung läßt sich, auch wenn dies beim Algorithmus nicht explizit zum Ausdruck kommt, in 3 Teilaufgaben gliedern:

1. Die Identifizierung der relevanten Texteinheiten, auch Tokenisation oder Tokenisierung genannt.
2. Die Analyse der bei der Tokenisierung gewonnenen Einheiten hinsichtlich aller möglichen Informationen, die durch diese Formen repräsentiert werden können.
3. Die Disambiguierung von Mehrfachanalysen.

Auf das Problem der Tokenisierung wird in Abschnitt 3.2 auf Seite 93 ausführlich eingegangen. In den folgenden Abschnitten soll ein kurzer Überblick über die verschiedenen Methoden gegeben werden, die zur Lösung der unter 2) und 3) auftretenden Probleme in bisherigen Systemen angewendet wurden. In der Literatur wird die Tokenisierung als Bestandteil der automatischen Lemmatisierung in der Regel vernachlässigt. Es wird sich aber zeigen, daß eine “intelligente” Tokenisierung die unter 2 und 3 anfallenden Probleme wesentlich vereinfachen kann.

3.1.2.1 Analyse der Wortformen

Bei der Analyse der Wortformen gibt es einen fließenden Übergang zwischen vorwiegend regelbasierten Verfahren und vorwiegend lexikonorientierten Verfahren.

Bei den regelbasierten Verfahren werden die Wortformen mit Hilfe einer umfangreichen Morphologiekomponente auf mögliche Grundformen reduziert, während beim lexikonorientierten Verfahren soviel Information wie möglich aus dem Lexikon genommen wird und die Regelkomponente minimal gehalten ist. Dazwischen gibt es Mischformen, die versuchen mit einem Minimallexikon (Schott 1978) oder einem reinen Stammlexikon und einer nicht ganz so umfangreichen Regelkomponente

(SALEM, Eggers 1980) auszukommen. Die Aufgabe der Wortformenanalyse ist die Reduktion der Wortform auf alle möglichen Lemmata.

3.1.2.1.1 Regelbasierte Wortformenanalyse

Das Problem bei der regelbasierten Wortformenanalyse ist die hochgradige Mehrdeutigkeit der deutschen Flexionsmorpheme (ein Problem, das sich unabhängig davon stellt, ob nach dem “longest-match”-Verfahren oder dem “shortest-match”-Verfahren vorgegangen wird). Zur Verbesserung der Ergebnisse werden deshalb häufig auch Wortbildungsmorpheme zur Analyse mit herangezogen: so läßt beispielsweise das Suffix *keiten* auf ein Nomen schließen (vgl. Willée 1979).

Bei den flektierenden Wortklassen sind allenfalls Ausnahmen in einem Lexikon verzeichnet. Da das Verfahren für einen Großteil der Funktionswörter aufgrund fehlender Flexive nicht anwendbar ist, müssen diese über eine sogenannte Stopliste, die alle Funktionswörter enthält, abgefangen werden. Im CONDOR-System werden diese Elemente völlig ignoriert, bei LEMMA ist die Stopliste als Lexikon der Funktionswörter angelegt, das die notwendige Lemmatisierungsinformation enthält. Aufgrund des Fehlens umfangreicher Lexika wurde in der Vergangenheit trotz der offensichtlichen Mängel und der Vielzahl von künstlichen Ambiguitäten die durch die Regeln erzeugt werden, vor allem dieses Verfahren angewendet.

3.1.2.1.2 Mischformen

Als Alternative zu ausschließlich regelbasierten Systemen wurde versucht, die Nachteile dieses Verfahrens durch den Einsatz relativ kleiner Lexika zu umgehen (Schott 1978, Eggers 1980). Im Saarbrücker System beispielsweise wird ein relativ umfangreiches Stammlexikon verwendet, anhand dessen die nach dem longest-match Verfahren aus den Wortformen ermittelten Stämme auf ihre Existenz hin überprüft werden. Eine etwas andere Strategie wird in Schott (1978) verfolgt. Sie benutzt ein sogenanntes Minimalwörterbuch, das neben den Funktionswörtern auch flektierte Wortformen enthält. Bei diesen Wortformen handelt es sich um Formen, die in bezug auf den Gesamtwortschatz ein hohes Verhältnis an Überschneidungen von Flexionsmorphemen und Wortbildungsmorphemen aufweisen und deshalb in rein regelbasierten Systemen extreme Probleme bereiten.

3.1.2.1.3 Lexikonbasierte Wortformenanalyse

Im Gegensatz zu den bisher beschriebenen Verfahren ist die Wortformenanalyse bei lexikonbasierten Verfahren auf das Nachschlagen der gesuchten Wortform im Lexikon beschränkt. Lediglich im Bereich der komplexen Wörter wird sich auch in einem umfangreichen Lexikon keine Vollständigkeit erzielen lassen. Deshalb ist bei komplexen Wörtern, die nicht im Lexikon enthalten sind, eine Segmentierung in Elemente des Lexikons notwendig. Als Nachteil des lexikonbasierten Verfahrens wird häufig der hohe Speicherplatzbedarf angeführt. In diesem Bereich kann jedoch auf neuere Ent-

wicklungen zur speicherplatzeffizienten internen Datendarstellung von großen Datenmengen verwiesen werden (vgl. beispielsweise Gonnet 1984), so daß aufgrund der wesentlich größeren Korrektheit und der damit verbundenen Reduzierung des Aufwands bei der Disambiguierung eine Wortanalyse mit einem möglichst umfangreichen Lexikon in jedem Fall vorzuziehen ist.

3.1.2.2 Disambiguierung

Je nach dem gewählten Verfahren zur Wortformenanalyse, hat man es bei der Disambiguierung entweder nur mit echten, formbedingten Ambiguitäten (lexikonbasiertes Verfahren) oder zusätzlich auch noch mit künstlichen Ambiguitäten (falsche Analysen beim regelbasierten Verfahren) zu tun. Bei der Disambiguierung unterscheidet man je nach dem zur Disambiguierung verwendeten Kontext zwischen wortbezogenen Verfahren und satzbezogenen Verfahren. Für die Zukunft ist auch die Berücksichtigung größerer Kontexte als die des Satzes denkbar, im Moment ist jedoch die Satzebene der maximale Kontext, der von Systemen verwendet wird.

3.1.2.2.1 Wortbezogene Lemmatisierung

Ist der Kontext, der zur Disambiguierung verwendet wird, auf das Wort beschränkt, so bieten sich für eine Disambiguierung kaum Hinweise. Lediglich die Groß- bzw. Kleinschreibung bietet eventuell einen Anhaltspunkt. Eine andere Möglichkeit ist die statistisch-basierte Disambiguierung. In diesem Fall wird aus der Menge der aufgrund der Formanalyse möglichen Lemmata dasjenige ausgewählt, das in einem manuell bearbeiteten Referenzkorpus die größte Häufigkeit hat. Ein solches Verfahren wurde beispielsweise im Rahmen des Saarbrücker Sonderforschungsbereichs (SFB 100) implementiert. Auch der Algorithmus von Schott (1978) arbeitet wortbezogen, indem bei der Wortformenanalyse erst gar keine Ambiguitäten zugelassen werden.

3.1.2.2.2 Satzbezogene Lemmatisierung

Die Lemmatisierung auf Satzebene ist natürlich der Lemmatisierung auf Wortebene weit überlegen.[1] Welche Informationen zur Disambiguierung herangezogen werden hängt von der Analysetiefe der Satzanalyse ab. Der Idealfall wäre natürlich eine vollständige syntaktische und semantische Analyse des Satzes. Die Möglichkeit einer solchen Analyse liegt für beliebige Sätze aus großen Korpora jedoch noch in weiter Ferne. Die Analysetiefe schwankt bei den verschiedenen Systemen erheblich: während beispielsweise bei LEMMA (Willée 1979) nur eine Umgebungsanalyse zur Dis-

1. Wenngleich auch auf der Satzebene nicht alle Ambiguitäten aufgelöst werden können, wie das folgende Beispiel zeigt: “umfuhr” in “Daß er den Schutzmann umfuhr, hatte Folgen.” muß korrekterweise sowohl auf “umfahren” als VST1s, als auch auf VSTT1s/2, also auf ein Verb mit nicht-abtrennbarem Präfix und auf ein Verb mit abtrennbarer Partikel, lemmatisiert werden. Diese Ambiguität kann auf Satzebene ohne weitere Zusatzinformationen selbst von einem kompetenten Sprecher nicht aufgelöst werden.

ambiguierung stark ambiger Endungen durchgeführt wird, wird im Saarbrücker satzbezogenen Verfahren eine Homographenanalyse, die Zusammenführung getrennter Verbformen, die Identifikation fester Syntagmen sowie eine Homographenauflösung angestrebt.

Neben Verfahren, die zur Disambiguierung aus einer Umgebungsanalyse gewonnene linguistische Informationen verwenden, können zur reinen Wortartdisambiguierung auch statistische Informationen herangezogen werden, wie dies beispielsweise beim automatischen Wortklassentagging (Church 1980, DeRose 1988, Feldweg 1993, u.a.) häufig der Fall ist. Diese Verfahren beruhen im wesentlichen darauf, daß in einem manuell getaggten Korpus die Häufigkeiten von Abfolgen bestimmter Tags ermittelt werden. Mithilfe solcher N-Gramm-Wahrscheinlichkeiten von Tags kann aus einer Menge ambiger Wortklassentags dann der wahrscheinlichste Tag ausgewählt werden.

3.1.3 Anwendungen

Abschließend soll in diesem Abschnitt noch auf die Anwendungen von Lemmatisierung bzw. von lemmatisierten Texten eingegangen werden. Unter linguistischer Perspektive sind automatisch lemmatisierte Korpora oder auch Konkordanzen auf Lemmabasis für die verschiedensten Fragestellungen eine wichtige Datengrundlage. Beispiele für solche Anwendungen sind etwa Untersuchungen zu Verbrahmen, Verhalten von Komposita, Extraktion von Mehrwortlexemen, usw. Aber auch für Untersuchungen zum Wortschatz und zur Frequenz bestimmter Lexeme sind lemmatisierte Korpora unerläßlich. Neben solchen linguistisch motivierten Anwendungen gibt es im Bereich der maschinellen Sprachverarbeitung weitere wichtige Anwendungsgebiete:

1. Tagging
   Die Anforderungen bei der Lemmatisierung gehen über die Anforderungen, die an einen Wortklassen-Tagging-Algorithmus gestellt werden, weit hinaus, was bei vielen Tagging-Algorithmen zu einer Vernachlässigung der verfügbaren linguistischen Information geführt hat. Die in der Regel mit statistischen Methoden arbeitenden Tagger haben zwar eine verhältnismäßig hohe Erfolgsquote (häufig über 90%), die verbleibenden Fehler lassen sich bei diesem Ansatz allerdings kaum beheben. Mit Hilfe einer lexikonbasierten Lemmatisierung der Formen kann hier eine erhebliche Steigerung der Performanz erreicht werden. Tapnainen und Voutilainen (1994) zeigen, daß schon der Einsatz einer einigermaßen elaborierten Morphologiekomponente beim Wortklassen-Tagging zusammen mit linguistisch motivierten Disambiguierungsregeln dem rein statistischen Wortklassen-Tagging deutlich überlegen ist.
2. Alignment
   Unter Alignment versteht man die Parallelisierung mehrsprachiger Korpora. Bislang erreichen die gängigen Alignment-Algorithmen (siehe z.B. Gale/Church 1991, Kay/Röscheisen 1988) die Parallelisierung entsprechender Abschnitte und Sätze. Die Identifikation sich entsprechender Einheiten geschieht mit statistischen Mitteln. Wallner (1994) zeigt, daß durch die Anwendung dieser Alignment-Techniken auf

zuvor lemmatisierte Texte eine deutliche Verbesserung der Algorithmen erreicht werden kann. Für die Zukunft verspricht dieses Verfahren im Hinblick auf die automatische Extraktion mehrprachiger Terminologien viel Erfolg.

3. Information Retrieval
   Herkömmliche Verfahren zum Information Retrieval aus Volltext-Datenbanken benutzen zur Recherche lediglich die Suche nach bestimmten Strings. Um auch Flexionsformen miterfassen zu können, ist häufig auch noch eine Trunkierung bzw. die Suche nach einem Stamm mit beliebiger Endung möglich. Das hat zur Folge, daß teilweise zu viel und falsche Stichwörter aufgesucht werden, wenn sie zufällig mit demselben String wie der angegebene Stamm beginnen. Andere Formen, die irregulär gebildet werden oder allomorphe Stämme haben (im Deutschen beispielsweise häufig bei Nomen, wenn im Plural Umlautung stattfindet), können gar nicht gefunden werden. Auch Vorkommen in Komposita oder anderen komplexen Formen lassen sich kaum identifizieren. In diesem Bereich ermöglicht die Lemmatisierung eine große Verbesserung der Erfolgsquote.
4. Automatische Indizierung
   Was fürs Information Retrieval bereits gesagt wurde, gilt im großen und ganzen auch für die automatische Indizierung von Texten. Auch hier ist es von großer Bedeutung, alle relevanten Vorkommen von Begriffen, sei es in flektierter Form oder in komplexen Formen, zu erkennen.

Neben diesen Hauptanwendungen gibt es noch weitere Anwendungen, in denen zumindest einzelne Module eines Lemmatisierers eine wichtige Rolle spielen, wie etwa die maschinelle Übersetzung oder eine “intelligente” automatische Rechtschreibkorrektur im OCR-Bereich.

3.2 Tokenisierung

Der erste Schritt bei der automatischen Textlemmatisierung besteht im Einlesen des Textes und dessen Aufsplittung in die relevanten Einheiten, die dann von der Wortformanalyse weiterverarbeitet werden. Dazu werden die relevanten Texteinheiten extrahiert, die Schreibung normalisiert, und gleichzeitig werden die Texteinheiten nach ihrer Struktur typisiert, um so sofort an die geeignete Prozedur der Wortformenanalyse übergeben zu werden. Ein weiteres Ziel bei der Tokenisierung ist die Annotierung von lemmatisierungsrelevanten Kontextmerkmalen, wie etwa die Position im Satz, Informationen über Vorgänger und Nachfolger, Informationen zur tatsächlichen Orthographie im Text, usw.

Einfachere Verfahren zur Tokenisierung splitten den Eingabesatz an Leerzeichen und an einer Menge von vordefinierten Separatoren. Separatoren sind typischerweise die üblichen Satzzeichen (**.,:;!?** ...) und Klammerzeichen (**()[]{}”**...). Ein solches Verfahren führt jedoch zu zahlreichen inkorrekten Segmentierungen:

- Zahlen mit Dezimalkomma oder -punkt werden getrennt: 1,05 -> [1] [05]
- Datums- oder Verhältnisangaben werden getrennt: 1:4 -> [1] [4], 11.11.1999 -> [11] [11] [1999]
- Es ist keine Unterscheidung zwischen Punkt als Satzendezeichen und Abkürzungspunkt möglich: u.a. -> [u] [a]
- Wortinterne Klammerungen können nicht erkannt werden: (Un-)Sinn -> [Un] [Sinn]

usw.

Um der Polyfunktionalität der einzelnen Zeichen Rechnung zu tragen, ist hier eine detailliertere Analyse notwendig.

Prinzipiell werden drei Typen von Zeichen unterschieden:

1. **Buchstaben**:ABCDEFGHIJKLMNOPQRSTUVWXYZ
   abcdefghijklmnopqrstuvwxyz
   ÀÁÂÃÄÅÇÈÉÊËÌÍÎÏÑÒÓÔÕÖÙÚÛÜ
   àáâãäåçèéêëìÆíîïñŒòóôõöæùúûüøœßÿ
2. **Ziffern**: 0123456789
3. **(Sonder-)Zeichen**: .,:;-_#+*!~”§$<%>&|/{([)]=}?\`´@’ usw.

3.2.1 Identifikation von Satzzeichen und Klammern

Ausgangspunkt ist, wie bereits erwähnt, ein Text der mittels eines Satzendeerkennungsprogramms (Schicht 1994b) in einzelne Sätze zerlegt ist, so daß der Satzendepunkt als identifiziert vorausgesetzt werden kann (auf die Problematik der Satzendeerken-

nung wird in Abschnitt 3.2.2 noch näher eingegangen). Die Eingabe zur Tokenisierung ist also ein Satz, der in einem ersten Schritt an allen Spacezeichen (Blanks, Tabs usw.) in die einzelnen **Textformen** aufgesplittet wird. Textformen sind alle Einheiten eines Textes, die durch Leerzeichen separiert werden. Textformen können Wörter, Zahlen, einzelne Zeichen oder beliebige Mischungen sein. Hört eine Textform mit einem Satzzeichen, das eindeutig als solches identifizierbar ist oder mit einem Klammerzeichen, das für die Wortstruktur irrelevant ist[1], auf, so muß dieses im ersten Verarbeitungsschritt isoliert werden. Elemente, die keine solchen Zeichen mehr enthalten, bzw. nur aus solchen Zeichen bestehen werden im folgenden als **A-Formen** (für analysierbare Formen) bezeichnet. Das Ziel bei diesem Arbeitsschritt ist die Zerlegung einer Textform in eine Folge (auch der Länge 1) von A-Formen:

```
Textform -> A-Form+ [2]
```

Bei der Analyse der Textformen werden bei einer Form t folgende Fälle unterschieden (t2 ist wiederum eine Textform):

1. t hat weder am Anfang noch am Ende Satz- oder Klammerzeichen
2. t hört mit einem Satzzeichen auf: t = t2 [,?!;]
3. t beginnt und endet mit einem Klammerzeichen: t = [“’([{] t2 [“’)]}]
4. t endet mit einer schließenden Klammer: t = t2 [“’)]}]
5. t beginnt mit einer öffnenden Klammer: t = [“’([{] t2

Im ersten Fall wird die Textform t direkt als neue A-Form in die Liste der zu analysierenden Formen übernommen. Im zweiten Fall wird das Satzzeichen als A-Form abgetrennt und in die Liste der zu analysierenden Formen übernommen, dann wird die verbleibende Textform t2 derselben Textformanalyse erneut unterzogen, da es sich bei t ja um eine Textform wie *tatsächlich(!)*, handeln könnte, die außer dem Komma noch weitere Zeichen enthält, die extrahiert werden müssen. Das Ergebnis der Analyse von t2 wird vor dem zuerst abgetrennten Satzzeichen in die Liste eingefügt.

Falls t am Anfang oder am Ende ein Klammerzeichen hat, muß unterschieden werden, ob im Inneren das duale Zeichen vorkommt oder nicht. Falls im Inneren das duale Zeichen vorkommt, und nicht nur Zeichen von dieser Klammerstruktur umschlossen werden, kann davon ausgegangen werden, daß die Klammer für die Lemmatisierung der A-Form relevant ist, d.h., daß es sich um Formen von Typ *(Un-)sinn* handelt, bei denen die Klammern nicht isoliert werden dürfen. Im Fall 4 wird dann die gesamte Textform als A-Form übergeben (der Fall, daß im Anfangsglied nochmals Klammerstrukturen auftreten, kann hier vernachlässigt werden, da er im gesamten Testkorpus nicht vorkam.). Im Fall 5 muß das nach der schließenden Klammer verbleibende Restglied noch weiter analysiert werden. Und im Fall 3 schließlich kann das nicht paarweise auf-

1. Als für die Wortstruktur relevant betrachte ich Beispiele wie (Un-)Sinn.
2. Mit + bzw. * bezeichne ich im folgenden die für reguläre Ausdrücke typischen Operationen: x+ bedeutet ein oder beliebig viele x, x* bedeutet kein oder beliebig viele x.

tretende Klammerzeichen isoliert werden und dann wie in Fall 4 bzw. 5 weiterverfahren werden. Durch die Forderung, daß das duale Zeichen im Inneren vorkommt, werden einfache, randständige Klammerungen hier nicht erfaßt, da bei ihnen ja auch davon ausgegangen werden kann, daß sie für die reine Formanalyse irrelevant sind. Bei asymmetrischen Klammerungen bzw. Einklammerungen der gesamten Textform können die Klammerzeichen als A-Formen isoliert werden und die verbleibenden Reste weiter analysiert werden. Dieses Verfahren liefert aus dem Eingabesatz eine Liste von A-Formen, die im nächsten Verarbeitungsschritt typisiert, orthographisch normalisiert und mit Kontextmerkmalen versehen werden.

3.2.2 Satzendeerkennung

Um den eigentlichen Lemmatisierungsalgorithmus nicht unnötig zu überfrachten, wurde wie bereits erwähnt, ein am CIS entwickelter Satzendeerkenner als Präprozessor verwendet. In diesem Abschnitt soll deshalb nur kurz auf die Probleme, die bei der Satzendeerkennung auftreten, eingegangen werden. Das Problem besteht in erster Linie darin, den Satzendepunkt zu erkennen. Die Satzendemarkierung durch Frage- oder Ausrufezeichen ist vergleichsweise einfach, da ein Frage- oder Ausrufezeichen am Ende einer Form in den meisten Fällen ein Satzende markiert, und sei es nur das Ende eines eingeschobenen Satzes. Frage- oder Ausrufezeichen, die sich nur auf eine Form beziehen, sind in der Regel eingeklammert und damit nicht am Ende einer Textform. Die einzige Ausnahme sind Ausrufezeichen nach Interjektionen, die aber selbst Satzstatus haben.

```
Wenn der Fraktionsvorsitzende der CDU, Wolfgang Schäuble,
fordert, die Arbeit, die es gibt, in höherem Maße durch
deutsche Arbeitskräfte erledigen zu lassen (SZ vom 22. 12.),
fördert er fahrlässig (?) den Ungeist, der mit dem Ruf
Deutschland den Deutschen Ausländer drangsaliert.

Was soll man davon halten, wenn ein Parteichef moniert, die
Stellungnahme des Justiz(!)-Ministeriums zu Herrn Gauweilers
Mandantenpachtvertrag könne nicht als Stellungnahme der
Bundesregierung bezeichnet werden, da der Bundeskanzler dazu
noch keine Meinung abgegeben hat?

Hat der Willi den Kragen denn immer noch nicht voll? fragt
Siegfried Axtmann, der Präsident von Bundesliga-Schlußlicht
VfB Leipzig.

[...] bis auf das häufig wiederholte Alaaf! und Helau!, das
wir inzwischen kannten [...]

Also greifen Sie zum Hörer! sagt die SPD und verweist auf das
verfassungsmäßig garantierte Recht, sich mit Bitten und Sorgen
an den Landtag wenden zu können.
```

Problematischer dagegen ist die Erkennung des Satzendepunkts. Der Punkt am Ende einer Textform kann die Funktion eines Abkürzungspunkts, eines Satzendepunkts und nach Ziffern die Funktion eines Ordinalzahlpunkts haben. Dieses Problem stellt selbst im englischen Sprachraum ein Problem dar, obwohl im Englischen die Großschreibung der folgenden Form ein wesentlich besseres Indiz für ein Satzende ist, als im Deutschen. Garside/Leech/Sampson (1987) beispielsweise schließen den Punkt nach Formen, die nur aus Großbuchstaben bestehen, nach abgekürzten Titeln, akademischen Graden und Anreden als Satzendemarkierer aus, selbst wenn die darauffolgenden Formen großgeschrieben sind, da diese dann in der Regel Eigennamen sind.

Prinzipiell kommen auch im Deutschen nur Punkte deren folgende Form großgeschrieben ist, als Satzendepunkt in Frage. Beginnt die folgende Form mit einer Ziffer oder einem öffnenden Klammerzeichen, kommt ein Punkt ebenfalls als Satzendepunkt in Frage. Zur genauen Bestimmung der Funktion müssen folgende Fragen geklärt werden:

1. Welche Kategorie hat die folgende Form? Falls sich die Kategorie eindeutig bestimmen läßt und es sich nicht um eine nominale Kategorie handelt, ist ein Satzende anzunehmen.
2. Bildet die Punktform eine Abkürzung, die im Abkürzungslexikon mit Punkt verzeichnet ist? In diesem Fall ist der Punkt mit einiger Sicherheit ein Abkürzungspunkt. Problematisch sind Abkürzungspunkte, die gleichzeitig als Satzendepunkt dienen.
3. Stehen unmittelbar vor dem Punkt nur Ziffern? Dann ist die Form daraufhin zu untersuchen, ob es sich um eine Ordinalzahl handelt. Typischerweise steht vor Ordinalzahlen (außer Datumsangaben, die gesondert überprüft werden) ein Determinator. Ordinalzahlen innerhalb von Datumsangaben stehen entweder vor Monatsnamen, vor einer Jahreszahl, vor einer weiteren Ordinalzahl oder nach einer Ordinalzahl.

Das hier verwendete Programm testet diese Fälle systematisch mit Hilfe des Lexikons. Im Zweifelsfall wird ein Satzende angenommen. Dies ist insbesondere im Hinblick auf die Lemmatisierung nützlich, da hier eventuelle Fehler abgefangen werden können, indem nicht erkannte Formen vor Punkt am Satzende auf Abkürzung mit Punkt getestet werden können.

3.2.3 Typen von A-Formen

Die Typisierung der A-Formen dient der Steuerung der Formanalyse. Es findet aber auch eine Zusammenführung von Zahlen statt, die im Text als durch Leerzeichen getrennte Zahlentripel dargestellt sind, aber eine Einheit bilden (1 000 wird zu 1000). Durch die Typisierung soll erreicht werden, daß unsinnige Lemmatisierungsversuche von vornherein erst gar nicht unternommen werden. Es würde beispielsweise wenig Sinn machen, eine A-Form, die andere Elemente als Buchstaben enthält, direkt im DKL nachzuschlagen. Dies ist (abgesehen von Abkürzungen) nur dann sinnvoll, wenn

die Form nur Buchstaben enthält. Diese Formen werden im folgenden als **B-Formen** (für Buchstaben-Formen) bezeichnet. Alle anderen Formen werden als **X-Formen** bezeichnet, die nach ihrer Zusammensetzung noch weiter unterschieden werden.

```
A-Form -> B-Form

A-Form -> X-Form
```

Im Unterschied zu den A-Formen, die abgesehen von der Segmentierung dieselbe Form wie im Text haben, sind die B- und X-Formen normalisiert, das heißt, sie sind in der Regel kleingeschrieben. B- und X-Formen haben eine komplexe Struktur: sie bestehen aus der orthographisch normalisierten A-Form zusammen mit einem Merkmalsvektor, der Wortstrukturbeschreibung, in dem unter anderem die orthographischen Eigenschaften der A-Form kodiert sind. Diese Wortstrukturbeschreibung wird im Laufe der Analyse um weitere Informationen ergänzt. Sehr wichtig für die Bearbeitung wortübergreifender Phänomene ist beispielsweise die Kodierung der möglichen lexikalischen Typen, als die die Form analysiert werden konnte. Die relevanten lexikalischen Typen sind:

EF	einfache Form
KF	komplexe Form
BW	Bindestrich-Wort
NUM	jede Art von Numeral-Konstruktionen (außer Ordinalzahlen)
ORD	Ordinalzahlen
AK	Abkürzungen
EN	Eigennamen
CC	einzelne Zeichen oder Buchstaben
UK	unbekannte Elemente

Bei der Tokenisierung werden bereits bestimmte Merkmale der Wortstrukturbeschreibung definiert, die dann über das weitere Vorgehen bei der Lemmatisierung entscheiden. So wird neben der groben Unterscheidung nach B- und X-Formen eine feinere Typisierung der Formen vorgenommen, je nachdem welche Art von Zeichen die Form enthält (die relevanten Typen sind in Abschnitt 3.3 auf Seite 98 für die B-Formen und in Abschnitt 3.4 auf Seite 109 für die X-Formen detailliert beschrieben). Als weitere Merkmale werden die Originalorthographie und die Länge der Form in der Wortstrukturbeschreibung aufgenommen. Sie dienen der Bewertung von alternativen Lemmatisierungen, insbesondere bei Abkürzungen und komplexen Formen. Zusätzlich enthält die Strukturbeschreibung Merkmale, die über die wortinitiale Groß- bzw. Kleinschreibung Auskunft geben. Bei der initialen Großschreibung wird unterschieden nach Großschreibung in fragwürdigen Kontexten (Satzanfang, nach Gedankenstrich oder Doppelpunkt, nach IX-Typen, usw.) und Großschreibung in klaren Kontexten, d.h. im Satzinnern und nicht nach Formen, die einen fragwürdigen Kontext bilden. Bei komplexen Formen, wie Bindestrichwörtern, bei denen die initiale Großschreibung nicht notwendig auf die Kategorie der gesamten Form schließen läßt, wird zusätzlich die Groß- bzw. Kleinschreibung der einzelnen Komponenten vermerkt. Auf weitere Wort-

strukturmerkmale wird im Rahmen der Disambiguierung (vgl. Abschnitt 3.5) eingegangen.

3.3 Wortformenanalyse

Die Wortformenanalyse ist zuständig für alle Elemente, die nach der Tokenisierung nur Buchstaben enthalten, die sogenannten B-Formen. Es werden prinzipiell 4 Typen von B-Formen unterschieden:

- ("normale") Wortformen (**WB-Formen**): das sind solche Buchstabenketten, die auf den ersten Blick aussehen wie ein normales Wort. Das heißt, sie müssen aus mehr als nur einem Buchstaben bestehen, mindestens einen Vokal enthalten und allenfalls der erste Buchstabe kann im Originaltext großgeschrieben sein. Normale Wortformen sind das typische Erscheinungsbild von einfachen Formen, Komposita oder Eigennamen. Aber auch Abkürzungen und Akronyme können in dieser Form auftreten.
- Einzelne Buchstaben und Konsonantenfolgen (**CB-Formen**), die in der Regel als Abkürzungen lemmatisiert werden müssen (die einzige Ausnahme hier stellt die Präposition *à* dar).
- Wortformen, die nur aus Großbuchstaben bestehen (**GB-Formen**). Es kann sich bei diesen Wortformen sowohl um normale Wortformen handeln, die im Text aus bestimmten Gründen durch Großschreibung repräsentiert werden, als auch um Abkürzungen oder Akronyme.
- Wortformen, die sowohl Klein- als auch Großbuchstaben enthalten, wobei Großbuchstaben auch in der Wortmitte auftreten müssen (**MB-Formen**). Bei diesen Formen kann davon ausgegangen werden, daß es sich nicht um einfache oder komplexe Formen handelt, sondern daß es sich entweder um Namen (z.B. Firmennamen), Abkürzungen oder Akronyme handelt.

Der Analyseprozeß sieht die Erzeugung aller in Frage kommender Lemmatisierungen vor.

3.3.1 Die Analyse der WB-Formen

Um bei den WB-Formen alle Lemmatisierungsmöglichkeiten zu erhalten, werden diese Formen unabhängig hinsichtlich der folgenden Kategorien getestet:

- Einfache Form: Look-up im CISLEX-EF
- Komplexe Form: Kompositaanalyse
- Eigenname: Look-up im CISLEX-EN
- Abkürzung und Akronym: Look-up im CISLEX-AK

```
WB-Form -> einfacheForm
```

```
WB-Form -> komplexeForm

WB-Form -> Eigenname

WB-Form -> Abkürzung
```

Die Ergebnisse dieser Tests werden in einer Liste gesammelt und der Disambiguierung übergeben.

3.3.1.1 Look-up im CISLEX-EF

Die Lemmatisierung einfacher Formen ist durch das zugrundeliegende CISLEX-FLEX Lexikon weitgehend auf den Look-up im Lexikon reduziert (Beispiele aus dem CISLEX-FLEX sind in Anhang B auf Seite 200ff angeführt). Es müssen jedoch einige Sonderfälle beachtet werden, die nicht explizit im Lexikon aufgeführt sind (vgl. dazu die Diskussion in Abschnitt 3.1.1.3.1 auf Seite 84). Zusätzlich zum Look-up im FLEX-Lexikon müssen noch folgende Fälle überprüft werden:

- Nominalisierte Adjektive:
  Konnte eine Wortform als flektiertes (d.h. hinsichtlich Kasus, Genus, Numerus und Deklination spezifiziertes) Adjektiv identifiziert werden, so besteht prinzipiell die Möglichkeit, daß es sich auch um ein nominalisiertes Adjektiv handelt. Es wird demzufolge ein weiteres Lemma generiert, dessen Lemmaform der Stamm des entsprechenden Adjektivs mit dem Flexiv *e* ist (Positiv-, Komparativ- und Superlativformen werden bei den nominalisierten Adjektiven im Gegensatz zu den eigentlichen Adjektiven als verschiedene Lemmaformen behandelt). Die Lemmaklasse ist NA für Nomen mit adjektivischer Flexion und die morphosyntaktischen Merkmale entsprechen bis auf das Komparationsmerkmal denen des Adjektivs.

  ```
  Weite:
  weit.ADJ11:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp
    :aeFzp
  weite.NA:neFx:nmUx:aeFx:amUx:neUy:aeFy:aeNy:neFz:aeFz
  ```

- Nominalisierte Infinitive:
  Bei den nominalisierten Infinitiven sind zwei Fälle zu unterscheiden: die Nominativ-, Dativ- und Akkusativform, die mit dem verbalen Infinitiv identisch ist und die Genitivform, die aus dem verbalen Infinitiv und Genitiv-s besteht. Im ersten Fall kann die Wortform im EF-FLEX als Verb im Infinitiv identifiziert werden. Es muß dann nur noch ein weiteres Lemma generiert werden, dessen Lemmaform mit dem Infinitiv übereinstimmt, das als Lemmaklasse die entsprechende Nomenklasse hat und als morphosyntaktische Merkmale Nominativ-, Akkusativ- und Dativ-Singular-Neutrum. Falls es sich bei der Wortform um eine Form handelt, die auf *ns* endet, so muß überprüft werden, ob die Wortform ohne *s* am Ende auf einen verbalen Infinitiv lemmatisiert werden kann. Ist dies der Fall, so wird ein neues Lemma generiert, das als Lemmaform die Form ohne *s* hat, als Lemmaklasse wie im vorigen Fall die entsprechende Nomenklasse und als morphosyntaktisches Merkmal Genitiv-Singular-Neutrum.

```
Laufen:
laufen.VST1:OI:1mGi:3mGi:1mGc:3mGc
laufen.neut(NS2,NPSG):neN:deN:aeN

Laufens:
  laufen.neut(NS2,NPSG):geN
```

Die Menge möglicher Lemmata setzt sich dann aus den Lemmata, die im FLEX-Lexikon unter der entsprechenden Form verzeichnet sind, und den eventuell über Regeln generierten Lemmata zusammen. Lediglich unfreie Elemente wie Fugen oder Affixe müssen ausgeschlossen werden.

3.3.1.2 Kompositaanalyse

Der hier vorgestellte Zerlegungsalgorithmus benutzt lediglich das Lexikon der einfachen Formen, da nach dem Prinzip der Generierung aller möglichen Lemmatisierungen auch alle möglichen Segmentierungen einer komplexen Form erzeugt werden sollen.[1] Das Problem der Segmentierungsrichtung (von links oder von rechts) ist bei der Zerlegung ebenso wie die Frage, ob nach dem “longest match”-Verfahren oder nach dem “shortest match”-Verfahren vorgegangen werden soll, irrelevant. An dieser Stelle werden durch ein erzwungenes Backtracking alle möglichen Segmentierungen erzeugt. “longest match”-Erwägungen unter Beachtung der Richtung spielen erst bei der Disambiguierung eine Rolle.

Bei der Segmentierung wird versucht, die Wortform in eine Struktur der Form [Vorderglied][Vorderglied]*[Hinterglied] zu zerlegen. Ein Hinterglied kann ein Eigenname oder eine einfache Form sein, die die Bedingung für Hinterglieder erfüllt. Nach dieser Bedingung muß ein Hinterglied einer der folgenden Klassen angehören:

Nomen oder nominalisierte Adjektive oder nominalisierte Infinitive außer Fugenformen

Verben außer den als Fugenformen kodierten Verbstämmen und Imperativformen

Adjektive oder Adverben

Suffixe

Als Vorderglieder kommen neben Eigennamen, die gesondert behandelt werden, folgende Klassen in Frage:[2]

Funktionswörter

1. Das in Abschnitt 2 auf Seite 26 erwähnte Lexikon der komplexen Formen, DKL-KF war zum Zeitpunkt dieser Arbeiten noch nicht vollständig, so daß auf seinen Einsatz hier verzichtet wurde.. Das KF-Lexikon kann jedoch bei der Disambiguierung zur Auswahl der wahrscheinlichsten Segmentierung benutzt werden.

2. Das Vorkommen von Eigennamen in Komposita ist ausgesprochen selten. Der Look-up im EN-Lexikon wird deshalb nur dann ausgeführt, wenn alle anderen Analyseversuche gescheitert sind.

Prädikativformen von Adjektiven und Superlativstämme als Fugenformen

Verbale Infinitve und Verbstämme

Fugenformen von Nomen

Präfixe

Fugenformen der Suffixe (jedoch nicht am Wortanfang)

Der Segmentierungsalgorithmus versucht zuerst ein Suffix abzutrennen, das mittels Look-up im FLEX-Lexikon als mögliches Hinterglied in Frage kommt. Ist ein solches Suffix gefunden, wird versucht den verbleibenden Anfang entweder als Vorderglied oder als Folge von Vordergliedern zu identifizieren. Bei der Segmentierung muß beachtet werden, daß bei der Komposition dreifache Konsonanz vor Vokalen nicht erlaubt ist und ein Konsonant getilgt wird (*Still+Leben = Stilleben, Stoff + Fetzen = Stoffetzen*, aber: *Stoff + Flecken = Stoffflecken*). Immer wenn bei einer Segmentierung links und rechts einer Segmentgrenze derselbe Konsonant auftritt und rechts an zweiter Stelle ein Vokal folgt, muß zusätzlich versucht werden, auch eine Segmentierung des linken Teils als Vorderglied oder Folge von Vordergliedern zu finden, wenn der auslautende Konsonant verdoppelt wird. Dieses Verfahren wird für alle möglichen Segmentierungen der Wortform wiederholt.

Aus den Segmentierungen werden die möglichen Lemmata generiert: die Lemmaklasse und morphosyntaktischen Merkmale werden vom Hinterglied übernommen. Die Lemmaform ergibt sich durch Konkatenierung der Vorderglieder mit der Lemmaform des Hinterglieds, wobei an allen Segmenttrennstellen zusätzlich ein Separatorzeichen (“|”) eingefügt wird. Lediglich die zu-Infinitive bei komplexen Verben müssen anders behandelt werden, da als Lemmaform der normale Infinitiv des komplexen Verbs dienen soll. Falls eine komplexe Form segmentiert wird in eine Folge von Vordergliedern, deren letztes Element *zu* ist, und das Hinterglied ein verbaler Infinitiv ist, muß bei der Bildung der Lemmaform das *zu* getilgt werden.[1]

Dieses Verfahren liefert außer im Fall von Zahlausdrücken auch die gewünschten Lemmatisierungen. Betrachtet man komplexe Formen, die Zahlausdrücke enthalten, so liefert dieses Verfahren zwar formal korrekte Segmentierungen, die Lemmata sind jedoch für Anwendungen nicht brauchbar. Bei der komplexen Form *einundzwanzig* wünscht man sich doch etwas mehr Information als die Lemmaform *ein|und|zwanzig* und die Lemmaklasse *Adjektiv*. Ebenso sollte bei hochproduktiven Kombinationen aus Zahlwort und entsprechenden Affixen wie *-malig, -fach, -jährig, -achser, -ender, -tonner* usw., die sowohl mit Zahlausdrücken als auch mit Zahlen vorkommen, kein Unterschied zwischen der Kombination mit dem Zahlwort und der Kombination mit einer Zahl bei der Lemmatisierung gemacht werden. Aus diesem Grund enthält der Segmen-

1. Genauer muß überprüft werden, ob das dem *zu* vorangehende Vorderglied nicht in einer Liste von Präfixen enthalten ist, die zusammen mit *zu* ein komplexes Präfix bilden, wie beispielsweise *hin*. Denn Formen wie *hinzukommen* sind ambig: zu-Infinitiv von *hin|kommen* oder normaler Infinitiv von *hinzu|kommen*. In diesem Fall müssen natürlich beide Lemmatisierungen berücksichtigt werden.

tierungsalgorithmus für komplexe Formen eine spezielle Grammatik für komplexe Zahlformen, die im folgenden Abschnitt separat beschrieben wird.

```
komplexeForm -> Vorderglied+ Hinterglied

komplexeForm -> ZahlForm

Vorderglied -> FugenForm|Präfix|Eigenname (Vorderglied)

Hinterglied -> Nomen|Verb|Adjektiv|Adverb|Suffix
```

Bislang wurde davon ausgegangen, daß bei komplexen Formen alle Bestandteile entweder in einem der CISLEX-Wörterbücher (dazu gehört auch die Liste der Präfixe) aufgelistet sind oder aber durch einfache Regeln als einfache Formen daraus abgeleitet werden können. Komplexe Formen können häufig aber auch Vorderglieder aus dem Fremdwortbereich oder Namensbereich haben, die möglicherweise nicht im Lexikon sind. In diesem Fall werden Segmentierungsversuche, bei denen nur ein Suffix identifiziert werden konnte, zwischengespeichert und nur für den Fall, daß die Wortform weder als einfache Form noch als komplexe Form noch als Eigenname oder Abkürzung bzw. Akronym identifiziert werden konnte, mit einer entsprechenden Kennzeichnung ausgegeben.

3.3.1.3 Analyse komplexer Zahlformen

Zahlformen können sowohl Numeralia sein, die morphologisch komplex sind, wie *dreizehn* oder komplexe Formen, in denen Numeralia auftreten, wie *Sechzehnender, zweijährig, Neunziger, verfünffachen*, usw. Es sollen hier allerdings nur solche Formen untersucht werden, die produktiv sind in dem Sinn, daß sie mit beliebigen Numeralformen gebildet werden können. Die Analyse der komplexen Zahlformen (innerhalb der B-Formen) zerfällt in zwei Probleme: erstens das Problem, bei der Kompositazerlegung zu erkennen, daß es sich um eine komplexe Zahlform handelt und zweitens die eigentliche Analyse der Zahlform. Ziel der Analyse ist es, Numeralia als die Zahl (Ziffernfolge), die sie darstellen, zu analysieren und komplexe Ausdrücke mit Numeralkomponente lexikalisch anhand einer Grammatik für diese Formen zu analysieren. Der Hauptgrund für die Unterscheidung der komplexen Zahlausdrücke von anderen komplexen B-Formen ist das Ziel, Ausdrücke wie *25* und *fünfundzwanzig* oder *10jährig* und *zehnjährig* auch auf Lemmaebene jeweils in Beziehung setzen zu können.

```
ZahlForm -> ZahlWort ZSuffix

ZahlForm -> ZPräfix ZahlWort

ZahlForm -> ZZirkumfix_anfang ZahlWort ZZirkumfix_ende
```

Die Erkennung von komplexen Zahlausdrücken und ihre Zerlegung geschieht im Rahmen der Kompositaanalyse. Wenn während der Zerlegung nur Numeralia, die Konjunktion *und* und als solche gekennzeichnete Numeralaffixe auftreten, so wird die

segmentierte Form als Zahlform analysiert. (Beispiele für einfache Zahlwörter und Z-Affixe sind Anhang D auf Seite 206 zu entnehmen.)

3.3.1.3.1 Analyse der Zahlen

Die Aufgabe einer Zahlengrammatik ist sowohl die Überprüfung einer Zahlform auf ihre Wohlgeformtheit, als auch die Übersetzung in die entsprechende numerische Darstellung. Da in diesem Zusammenhang von einer wohlgeformten Eingabe ausgegangen werden kann, werden Einzelheiten, wie etwa die korrekte Position der Konjunktion *und* in einem Zahlausdruck, nicht überprüft. Ist ein Zahlausdruck aus anderen Gründen als diesem nicht wohlgeformt, so scheitert die Übersetzung in den numerischen Ausdruck. Als Zahlformen in Frage kommen Kardinalzahlen (*eins, einhundertvierundfünfzig,* usw.), Ordinalzahlen (*erstens, neuntem, zwanzigster,* usw.) und Bruchzahlen (*achtel, viereinhalb,* usw.). Die Bruchzahlen und Ordinalzahlen kann man als Ableitungen der Kardinalzahlen betrachten, so daß die Analyse der Kardinalzahlen hier im Vordergrund steht.

```
ZahlWort -> KardinalzahlWort

ZahlWort -> OrdinalzahlWort

ZahlWort -> BruchzahlWort

KardinalzahlWort -> (einfachesZahlwort+ “und”
  einfachesZahlwort+)+[1]
```

Die Basis der Zahlenanalyse ist das Lexikon der Zahlausdrücke. Es enthält alle morphologisch einfachen Zahlausdrücke mit der Kategorie NUM. Als Merkmal enthält jedes NUM den Koeffizienten und den Exponenten der höchsten Zehnerpotenz ungleich 0, die die Zahl enthält. Zur Übersetzung der Zahlausdrücke wird der Ausdruck zuerst in eine Liste der Paare der Koeffizienten und Exponenten umgewandelt, wobei Einer und Zehner in die korrekte Reihenfolge gebracht werden. Dann wird durch Addition der jeweils maximalen Exponenten der tatsächliche Exponent zu den Koeffizienten ermittelt. Der numerische Wert ergibt sich dann durch einfaches Ausrechnen (Summe über alle Produkte aus Koeffizient und zugehöriger Zehnerpotenz).

Bei den Ordinalzahlen werden im Lexikon nur die morphologisch einfachen Formen kodiert. Das sind die Formen mit Exponent 0 und irreguläre Formen. Sie erhalten die Kategorie ORD und als Merkmale analog zu den einfachen Kardinalzahlen den Koeffizienten und entsprechenden Exponenten. Komplexe Ordinalzahlen setzen sich zusammen aus einem Kardinalzahlausdruck und einer einfachen Ordinalzahl. Die numerische Darstellung wird dann genauso wie im Falle der Kardinalzahlen berechnet. Prinzipiell dasselbe gilt auch für die Bruchzahlen. Allerdings wird bei Bruchzah-

1. Wie bereits oben erwähnt, lassen die Regeln zur Analyse von komplexen Zahlwörtern auch nicht wohlgeformte Ausdrücke zu, beispielsweise falsche Positionen von *und*. Hier ist für die Zukunft die Entwicklung einer vollständigen Zahlengrammatik notwendig.

len, deren Zähler größer als 1 ist, in der Praxis, wenn überhaupt die ausbuchstabierte Schreibweise gewählt wird, die Getrenntschreibung vorgezogen. Ausnahmen sind gemischte einfache Brüche (d.h. gemischte Brüche deren Zähler eins ist, wie *dreieinhalb*) und häufig vorkommende Bruchzahlen (*dreiviertel*). Hier ist es am sinnvollsten, die häufig vorkommenden Bruchzahlen alle explizit zu kodieren, während die einfachen Fälle der gemischten Brüche mit einfachen Heuristiken zu bewältigen sind.

3.3.1.3.2 Lexikalische Ableitungen mit Zahlen

Die Zahlformen (außer den Numeralia), die produktive Muster bilden, werden durch eine eigene Grammatik beschrieben. Die in Frage kommenden Affixe sind im Lexikon mit einer speziellen Markierung versehen, die Auskunft darüber gibt, mit welchem Typ von Numeralia Kombinationen möglich sind. Das nominale Suffix[1] *klässler* z. B. kann nur mit Ordinalzahlen kombiniert werden, während das Suffix *tonner* mit Kardinalzahlen, Dezimalzahlen und Bruchzahlen kombiniert werden kann. Die hier behandelten Konstruktionen bestehen alle aus einem Affix das im Lexikon entsprechend markiert ist und einer Numeralform. Es kann sich bei den Affixen sowohl um Suffixe, Präfixe als auch um Zirkumfixe (*ver*-n-*fachen*) handeln.

3.3.1.4 Look-up im CISLEX-EN

Das CISLEX-Eigennamenlexikon stellt einen sehr wichtigen Bestandteil des CISLEX-Systems dar. Es ist nach verschiedenen Arten und Bereichen der Eigennamen unterteilt. Die Teile sind:

- Personennamen: hierunter fallen Vornamen und Nachnamen allgemein sowie Persönlichkeiten
- Geographische Namen und ihre Ableitungen. Die geographischen Namen sind unterteilt in UNO-Länder, nicht-UNO-Länder und Gebiete, Städte, Straßennamen und sonstige geographische Namen (Flüsse, Berge, Stämme usw.).
- Firmennamen mit einer Grammatik zur Erkennung komplexer Firmennamen.
- Sonstige Eigennamen

Die Eigennamen sind, falls sie flektieren mit einer morphologischen Kodierung, die der Kodierung der einfachen Formen entspricht, versehen. Darüber hinaus sind weitere typenspezifische Informationen kodiert, wie etwa bei Vornamen die Herkunft, das Geschlecht und die kanonische Form.

Das hier vorgestellte Lemmatisierungsverfahren macht von diesen Zusatzinformationen jedoch keinen Gebrauch, so daß für den Look-up ein vereinfachtes Eigennamenlexikon, das nicht nach den verschiedenen Bereichen untergliedert ist, verwendet werden kann (Beispiele befinden sich in Anhang C auf Seite 205). Das vereinfacht

1. Affix (entsprechend Suffix, Präfix, usw.) wird hier nur in einem rein formalen Sinn verstanden.

zum einen den Look-up, zum anderen ist die Ambiguitätsrate geringer. Falls für bestimmte Anwendungen genauere Informationen zu Eigennamen relevant sind (z. B. Information Retrieval, automatische Indizierung), so ist neben der im CISLEX-EN kodierten Information eine separate Komponente zur Erkennung von Eigennamen notwendig, da Eigennamen häufig als Mehrwortlexeme auftreten. Auf diese Problematik kann in diesem Rahmen jedoch nicht eingegangen werden.

Für die flektierenden Eigennamen oder auch deren Ableitungen (hier sind insbesondere die Ländernamen, deren adjektivische Ableitungen sowie die Bezeichnung der Einwohner und Einwohnerinnen betroffen) wurde dasselbe Format wie im DKL-FLEX angenommen, der Look-up geschieht nach denselben Regeln wie in Abschnitt 3.3.1.1 auf Seite 99 beschrieben. Für die anderen Eigennamen wird als Default angenommen, daß sie eine Genitiv- oder Pluralmarkierung in Form von *s* tragen können. Beim Look-up wird zuerst überprüft, ob die Form in der Eigennamenliste enthalten ist. Falls die Form auf *s* endet wird auch überprüft, ob die Form ohne *s* in der Liste enthalten ist. Als Lemmaform gilt in allen Fällen die im Eigennamenlexikon bzw. der Eigennamenliste verzeichnete Form. Als Lemmakategorie wird im Fall der flektierenden Eigennamen der Eintrag des Lexikons gewählt, im anderen Fall nur die Kategorie *EN*.

3.3.1.5 Look-up im CISLEX-AK

Der Look-up im Abkürzungslexikon wird hier zwar im Rahmen der WB-Formen beschrieben, gilt jedoch für eine Vielzahl der in der Folge näher erläuterten Formtypen, insbesondere auch für die anderen B-Formen. Aus diesem Grund werden hier alle Erscheinungsformen von Abkürzungen und deren Look-up beschrieben, auch wenn sie für die WB-Formen eigentlich nicht zutreffen. Die Grenzen zwischen den verschiedenen Formtypen lassen sich im Bereich der Abkürzungen ohnehin nicht so deutlich ziehen, da es bei den Abkürzungen eine große Varianz bezüglich Schreibweise mit und ohne Punkt und bezüglich der Groß- und Kleinschreibung gibt.

Das Abkürzungslexikon liegt momentan als Liste vor, die über 20 000 Abkürzungen und Akronyme aus den verschiedensten Bereichen enthält. Das Ziel ist es, jede Abkürzung mit ihren konventionalisierten Bedeutungen zu versehen, um so eine Abkürzung auf ihre eigentliche Bedeutung zurückführen zu können. Dieses Lexikon lag zum Zeitpunkt dieser Untersuchungen noch nicht vor, so daß es lediglich möglich ist, anzugeben, ob eine Form in der Liste der Abkürzungen verzeichnet ist. (Anhang C auf Seite 205 enthält einige Beispieleinträge aus dem verwendeten Abkürzungslexikon.)

Abkürzungen treten in verschiedenen Formen auf: es kann sich um einzelne Buchstaben handeln (*m* für *Meter*), einzelne Buchstaben mit Punkt (*u.* für *und*), Buchstabenfolgen in beliebiger Schreibweise, sowohl mit als auch ohne Punkt am Ende (*DDR, ca., vgl., Gebr., GmbH, StVO, Verz.*, usw.), oder auch mit Punkten zwischen den einzelnen Zeichen (*u.a., z.B.*, usw.). Daneben kommen auch Mischformen vor, die spezielle Zeichen enthalten (*B&B,* usw.). Bedauerlicherweise treten viele Abkürzungen in verschiedenen orthographischen Varianten auf. So ist beispielsweise der Punkt am

Ende oft optional (*min / min.*), teilweise werden auch Punkte im Inneren an den zugrundeliegenden Wortgrenzen optional gesetzt (*FDP/F.D.P.*), besonders uneinheitlich ist jedoch die Groß-/Kleinschreibung. Bei den schon stark lexikalisierten Akronymen trifft man häufig die normale Schreibweise neben der ursprünglichen reinen Großschreibung an (*Aids / AIDS*). Besonders problematisch ist die Varianz zwischen Getrennt- und Zusammenschreibung (*u.a. / u. a.*), da im einen Fall die Abkürzung als eine zu analysierende A-Form und im anderen Fall als zwei unabhängige A-Formen zum Look-up kommt. Taucht bei der Analyse des Satzes eine Folge von Abkürzungen auf, so muß also zusätzlich überprüft werden, ob die gesamte Folge der Abkürzungen als Abkürzung im Lexikon zu finden ist.

Neben dem unflektierten Vorkommen von Abkürzungen kommen auch “flektierte” Abkürzungen mit einem *s* zur Markierung von Plural oder Genitiv in Texten vor (*CDs*/ *CD’s*). Auch in diesem Fall ist die Orthographie nicht einheitlich: Formen in denen das *s* direkt an die Abkürzung angehängt wurde, stehen neben Formen bei denen *’s* angefügt wurde. Die apostrophierten Fälle werden im Rahmen der Behandlung von Apostroph-Formen in Abschnitt 3.4.8.2 auf Seite 122 besprochen. In manchen Fällen wird auch die Flexionsendung des letzten Elements der zugrundeliegenden Form angefügt (*AGen* für *Aktiengesellschaften)*. In diesen Fällen ist momentan keine Analyse möglich, da die zugrundeliegenden Formen nicht rekonstruiert werden können. Wenn dies allerdings aus dem AK-Lexikon abgeleitet werden kann, kann anhand des EF-Lexikons überprüft werden, ob es sich um eine korrekte Flexionsendung handelt.

Aufgrund der uneinheitlichen Schreibweise wurden die Abkürzungen für den Look-up beim Lemmatisieren so aufbereitet, daß als Eintrag im Lexikon eine normalisierte Schreibweise (alle Zeichen klein, ohne Punkte) steht. Als Grund- und Lemmaform wird beim Look-up die konventionalisierte Schreibweise zurückgegeben. Dieses Verfahren führt natürlich zu einer großen Rate von Fehlanalysen, insbesondere da auch auf Flexions-*s* am Ende geprüft wird. Aus diesem Grund dürfen nur bestimmte Abweichungen von der konventionalisierten Schreibweise erlaubt werden. Es ist allerdings sehr schwierig in diesem Bereich Verallgemeinerungen über die Freiheitsgrade in der Orthographie zu formulieren. Es ist die Aufgabe des Abkürzungslexikons, die verschiedenen orthographischen Varianten zu jeder Abkürzung zu verzeichnen. Diese Fehlanalysen lassen sich momentan nur mit Hilfe von Heuristiken einschränken (vgl. dazu auch die Ausführungen in Abschnitt 3.6.3.2 auf Seite 139).

3.3.2 Die Analyse der CB-Formen

Einzelne Buchstaben oder Folgen von Konsonanten tauchen im Text meistens als Abkürzung auf (*m* für Meter, *h* für Stunde, *C* für Celsius, usw.). Sie werden wie in Abschnitt 3.3.1.5 auf Seite 105 beschrieben, lemmatisiert. Im CISLEX ist momentan nur ein einbuchstabiges Element, die Präposition *à*, bekannt, so daß auf den look-up im EF-Lexikon in diesem Fall verzichtet werden kann, da dieser Fall explizit bei der Analyse der CB-Form überprüft wird. Im Falle von Konsonantenfolgen ist auch die Möglichkeit einer lautmalenden Interjektion (*hmm, grr, psst, sst,* usw.) gegeben. Im

Bereich der Lautmalereien diesen Typs ist die Produktivität relativ hoch, so daß es wenig Sinn machen würde, alle Möglichkeiten im Lexikon zu verzeichnen. Interjektionen sind jedoch meistens kontextuell leicht zu identifizieren, da sie in der Regel vor Ausrufezeichen stehen. Weitere Vorkommen von Einzelbuchstaben sind normalerweise wörtliche Vorkommen, auch in der Bedeutung von Notennamen. Eine Unterscheidung der verschiedenen Vorkommen von CB-Formen ist in der Regel ohne genaue Kontextanalyse nicht möglich. Die CB-Formen werden daher, wenn möglich als Abkürzung lemmatisiert. Können sie nicht als Abkürzung identifiziert werden, so erhalten sie die Lemmaform *CC*(<char>).

```
CB-Form -> “à”

CB-Form -> Abkürzung

CB-Form -> Interjektion

CB-Form -> Buchstabe
```

3.3.3 Die Analyse der GB-Formen

Wörter können in Texten aus verschiedenen Gründen in Großbuchstaben auftauchen. Bei Akronymen, Abkürzungen und Eigennamen ist häufig die kanonische Form die in Großbuchstaben. Daneben kann aber nahezu jedes Wort aus Gründen der Hervorhebung oder bestimmter Formatierungskonventionen als GB-Form auftreten. Der Lookup der GB-Formen unterscheidet sich nur insofern von der Analyse der WB-Formen, als in diesem Fall Eigennamen und Abkürzungen bzw. Akronyme präferiert werden sollten. Die Präferenz von Eigennamen macht aus zweierlei Gründen Sinn: zum einen gibt es Eigennamen, die in dieser Form konventionalisiert sind (*ARAL*, *ESSO*, usw.), zum anderen zeigen die Belege des Korpus, daß gerade Eigennamen häufig zur Hervorhebung oder aus anderen typographischen Gründen in Großschreibung auftauchen.

```
GB-Form -> Eigenname

GB-Form -> Abkürzung

GB-Form -> WB-Form
```

3.3.4 Die Analyse der MB-Formen

Die gemischten B-Formen sind entweder Abkürzungen oder Eigennamen. Alle anderen Fälle von WB-Formen im Korpus sind auf Fehler (entweder fehlerhafte Großschreibung im Wortinneren (*FÜr* am Satzanfang) oder das Fehlen von Blanks (*dieLeiche*)) zurückzuführen. Bei den Eigennamen sind in erster Linie Firmennamen in gemischter Schreibweise anzutreffen. Dabei handelt es sich sowohl um Zusammensetzung aus verschiedenen Wortanfängen (*BayWa*), als auch um die Großschreibung von Namensbestandteilen wie *AG* (*BayerAG*). Typischerweise tauchen in dieser Gruppe auch Nachnamen mit dem Präfix *Mc* (*McGuiness*) auf. Bei den Abkürzungen

kann die gemischte Schreibweise auf die verschiedenen Bestandteile der Abkürzung zurückzuführen sein (*GmbH*), es kann sich aber auch um ein Genitiv- oder Plural-s handeln, das an eine großgeschriebene Abkürzung angefügt wurde (*IMs*). Die Eigennamen und Abkürzungen werden, wie in Abschnitt 3.3.1.4 auf Seite 104 bzw. Abschnitt 3.3.1.5 auf Seite 105 bereits beschrieben, lemmatisiert. Ein Spezialfall der MB-Formen sind Personenbezeichnungen, bei denen das Suffix *in* oder eine Flexionsform davon großgeschrieben angefügt wurde, um die Unspezifiziertheit bezüglich des natürlichen Geschlechts anzuzeigen (*LeserInnen*). In diesen Fällen wird sowohl auf die männliche als auch auf die weibliche Personenbezeichnung lemmatisiert.

Falls diese Suche keinen Erfolg hat und die MB-Form auch durch einen fehlenden Blank entstanden sein könnte (das heißt, wenn die Form keine aufeinanderfolgenden Großbuchstaben enthält), wird versucht, die mit Großbuchstaben beginnenden Teile als WB-Formen zu analysieren.

```
MB-Form -> Eigenname

MB-Form -> Abkürzung

MB-Form -> einfacheForm “In”|“Innen”

MB-Form -> Fehler: WB-Form+
```

3.4 Analyse von Sonderformen und Mischformen

Sonderformen und Mischformen sind alle Ketten, die nach der Tokenisierung Zeichen aus dem Nicht-Buchstabenbereich enthalten, die sogenannten X-Formen. Die X-Formen stellen eine sehr inhomogene Menge dar, die sowohl Ziffernfolgen, Bindestrichwörter, Abkürzungen, die verschiedensten numerischen Ausdrücke usw. enthalten. Je nach der Weiterverarbeitung wird hier aufgrund der bei der Tokenisierung erstellten Wortstrukturmerkmale eine ganze Reihe von Fällen unterschieden:

- Reine Ziffernfolgen oder einzelne Ziffern (**ZX-Formen**), die in der Regel als Zahl lemmatisiert werden.
- Reine Folgen von Zeichen (Zeichen außer Buchstaben und Ziffern)[1] oder einzelne Zeichen (**SX-Formen**). Es kann sich bei den SX-Formen sowohl um lexikalische Symbole handeln, wie etwa *$, %* usw. als auch um bei der Tokenisierung extrahierte Satzzeichen oder bestimmte Zeichenfolgen, wie "...".
- Mischformen aus Ziffern und Zeichen (**ZSX-Formen**), bei denen es sich in der Regel um numerische Ausdrücke, wie Dezimalzahlen, Verhältnisangaben, Bruchzahlen oder Maßangaben usw. handelt.
- Mischformen aus Buchstaben und Zeichen (**BSX-Formen**). Diese Gruppe besteht aus sehr unterschiedlichen Formen, die fast alle eine gesonderte Behandlung erfordern: Bindestrichwörter, Abkürzungen, apostrophierte Formen, wortinterne Klammerungen, komplexe Namen usw.
- Mischformen aus Buchstaben und Ziffern (**BZX-Formen**). Die BZX-Formen enthalten einerseits produktive Wortbildungsmuster vom Typ *10jährig* oder *12ender*, die kompositionell analysiert werden müssen und andererseits idiosynkratische Namen und Typenbezeichnungen wie *Ö1, 3Sat, Audi100*, die als Eigennamen in einer speziellen Liste geführt werden.
- Mischformen aus Buchstaben, Ziffern und Zeichen (**BSZX-Formen**). In dieser Rubrik hat man es mit Namen wie *'60er* zu tun, daneben gibt es produktive Bildungen mit lexikalischen Zeichen wie *100%ig.* Den Hauptanteil bilden aber Formen, bei denen es sich um Zusammensetzungen aus anderen Typen handelt, wie etwa Bindestrichwörter, deren einzelne Komponenten anderen Formtypen angehören, wie *100-m-Marke*.
- Numerierungsformen (**IX-Formen**) wie *1), ii), a),* usw., die, obwohl sie der Form nach in die Klassen der BSX- bzw. der ZX-Formen gehören, separat behandelt werden, da die folgenden Formen optional großgeschrieben sein können.

1. Im folgenden werden die Sonderzeichen, wenn keine Verwechslungsgefahr besteht, nur als Zeichen bezeichnet.

3.4.1 Analyse von ZX-Formen

Reine Ziffernfolgen sind unproblematisch, sie erhalten die Kategorie *NUM*. Das einzige Problem, das bei Ziffernfolgen auftreten kann, ist die Abtrennung von Zifferntripeln durch Blanks bei großen Zahlen. Die Zusammenführung solcher Fälle zu einer zusammenhängenden Ziffernfolge erfolgt bereits bei der Tokenisierung. Über die Funktion einer Ziffernfolge (Jahreszahl, Zahl,...) kann ohne weiteren Kontext nichts ausgesagt werden. Als Lemmaform wird bei Ziffernfolgen (im Hinblick auf die Analyse der ZSX-Formen) der Bezeichner *Zahl* gefolgt von der Ziffernfolge in runden Klammern gewählt, wenn die Ziffernfolge die Form [1-9][0-9]* hat. Beginnt die Ziffernfolge mit *0*, so lautet der Bezeichner *ziffern*.

3.4.2 Analyse von SX-Formen

Die Analyse der SX-Formen erschöpft sich in einem einfachen Look-up im Verzeichnis der Zeichen und Zeichenkombinationen. Häufig vorkommende Zeichenkombinationen (+/-, ... usw.) werden dabei als eine Einheit betrachtet. Ist eine SX-Form nicht als ganzes verzeichnet, so wird versucht, sie in Teile, die im Zeichenlexikon aufgeführt sind, zu zerlegen. Diese Fälle werden allerdings durch die Tokenisierung schon weitgehend ausgeschlossen. Als Lemmaform wird, wie bereits bei den nicht anderweitig analysierbaren CB-Formen CC(<zeichen(name)>) gesetzt.

3.4.3 Analyse von ZSX-Formen

Bei den Mischformen aus Ziffern und Zeichen handelt es sich zum einen um die verschiedensten Zahlformen (Dezimalzahlen, Bruchzahlen, Zahlen mit Vorzeichen), zum anderen um komplexere Konstruktionen, deren Bestandteile Zahlformen und spezielle Zeichen sind. Die komplexen Konstruktionen sind häufig in ihrer Funktion schon relativ festgelegt. Als Oberkategorie für Zahlformen wurde *signZahl* für Zahlformen mit bzw. ohne Vorzeichen gewählt:

```
signZahl -> “+” zahl

signZahl -> “-” zahl

signZahl -> zahl

zahl -> integerZahl

zahl -> dezimalZahl

integerZahl -> [1-9][0-9]*

dezimalZahl -> [1-9][0-9]*[,.][0-9]+

dezimalZahl -> 0[,.][0-9]+

bruchZahl -> integerZahl“/”integerZahl
```

```
signbruchZahl -> “”|“+”|“-” bruchZahl[1]
```

Es ist zu beachten, daß die Form `integerZahl“/”integerZahl` den Typ Bruchzahl nicht eindeutig bestimmt. In dieser Form kommen auch Jahresfolgen (*1994/95*) vor, außerdem kann das Zeichen “/” auch als Trenner fungieren, beispielsweise um im Schneelagebericht die Schneehöhe im Tal und die Schneehöhe auf dem Berg zu trennen (*20/180*). In der Trennerfunktion kann “/” auch mehr als einmal innerhalb einer ZSX-Einheit vorkommen.

Mit Hilfe der verschiedenen Zahlformen lassen sich weitere Formen beschreiben:

```
ordinalZahl -> integerZahl “.”

prozentZahl -> signZahl “%”

verhaeltnisZahl -> signZahl “:” signZahl

gradZahl –> integerZahl “:” integerZahl (“:” integerZahl)
```

Die Verhältniszahlen kommen häufig als Sportergebnisse (*3:4*) aber auch als Verhältnisangabe (*50:50*) oder mit Dezimalzahlen (*1,5:2,5*) vor. Gradzahlen sind Angaben mit Maßeinheiten, die nicht auf einer dezimalen Skala basieren, wie die Winkelgrade oder Zeitangaben. Die Zeitangaben kommen jedoch auch mit dem Trennsymbol “.” vor. Ein Sonderfall zu Ordinalzahlen sind Datumsangaben:

```
datum -> [1-9][0-9]* “.” [1-9][0-9]* (“.” {[1-9][0-9]}[0-9][0-
  9])
```

Es muß in diesem Fall zusätzlich überprüft werden, ob die erste Ziffernfolge zwischen *1* und *31* und die zweite zwischen *1* und *12* für deutsches Datum (bzw. umgekehrt für amerikanisches Datum) liegt. Nur amerikanisches Format kann bei folgender Schreibweise auftreten:

```
datum -> [1-9][0-9]* “/” [1-9][0-9]* “/” {[1-9][0-9]}[0-9][0-
  9]
```

Von einer Jahresangabe kann man in der Regel auch ausgehen, wenn es sich um Formen der Art *’94* handelt, obwohl im Prinzip durch die Apostrophierung auch die Auslassung anderer kontextuell erschließbarer Zahlen oder auch Buchstaben realisiert werden kann. Dieser Fall trat jedoch im Testkorpus nie auf, so daß hier aufgrund der großen Häufigkeit der Typ *jahresZahl* angenommen wird. Eine andere Form, bei der Jahreszahlen eine Rolle spielen, sind die bereits erwähnten Reihungen vom Typ *1975/76*, bei denen eine vierstellige Zahl von einer zweistelligen Zahl, die um eins größer ist als der Einer-Zehner-Anteil der ersten Zahl, abgetrennt wird.

1. Da für die Analyse umfangreicherer Formeln ohnehin ein spezielles Modul notwendig ist, genügt es hier, sich auf die im intendierten Anwendungsbereich tatsächlich vorkommenden Formen zu beschränken.

```
jahresZahl -> “’” [0-9][0-9]

jahresZahlFolge -> [1-9][0-9][0-9][0-9] “/” [0-9][0-9]
```

Weitere Folgen sind mit ganzen Zahlen (*Seite 5/6*) ebenso möglich, wie mit Ordinalzahlen (*am 5./6. Januar*), wenn die jeweils zweite Zahl um eins größer ist als die erste.

```
integerZahlFolge -> integerZahl “/” integerZahl

ordinalZahlFolge -> ordinalZahl “/” ordinalZahl
```

Enthält eine ZSX-Form einen Bindestrich in nicht initialer Stellung, so handelt es sich aller Wahrscheinlichkeit nach um eine Differenzangabe, bei der im Gegensatz zu einer Subtraktion der kleinere Wert vorn steht (*1973-1975,* aber auch *1973-75, Seiten 125-167, 10.-15. Juli,* usw.).

```
differenzZahl -> zahl “-” zahl

differenzOrdinal -> ordinalZahl “-” ordinalZahl
```

Die Zeichen, die innerhalb von Ziffernfolgen auftreten, sind, wie die obige Aufstellung gezeigt hat, bezüglich ihrer Funktion in der Regel nicht eindeutig. Neben klaren Fällen, wie Vorzeichen, Dezimalkomma oder %, gibt es Kombinationen, die mit relativ großer Wahrscheinlichkeit richtig identifiziert werden können, wie apostrophierte Jahreszahlen, Dezimalpunkt, Ordinalzahlpunkt, Datumspunkt und Doppelpunkt. Der Slash (“/”) erweist sich jedoch als sehr ambig, und die Bedingungen, die an die verschiedenen Formen geknüpft sind, erweisen sich häufig als relativ schwache Heuristiken, insbesondere was die Abgrenzung zwischen der Funktion als reiner Trenner und anderen Funktionen betrifft. Hier muß entweder eine kontextabhängige Disambiguierung stattfinden oder in unklaren Fällen die relativ neutrale Trennerfunktion gewählt werden.

```
ZSX-Form -> ZSX-Form (“/” ZSX-Form)+
```

Die Unterscheidung zwischen Koordinationsbindestrich und Minuszeichen am Anfang einer ZSX-Form ist ohne weitere Information nicht möglich. Allerdings tritt der Fall des Koordinationsbindestrichs verglichen mit der Häufigkeit des Minuszeichens so selten auf, daß dieser Fall wirklich nur dann erwogen werden sollte, wenn eine der Vorgängerformen eine Bindestrichform ist, deren letzte Komponente eine analoge ZSX- oder ZX-Form ist (vgl. dazu auch die Ausführungen in Abschnitt 3.4.7.2.1 auf Seite 118).

Die Analyse der ZSX-Formen liefert als Ergebnis sowohl den Typ der gesamten Form, als auch die interne Struktur mit ihren Typen. Als Lemmaform wird die Struktur der ZSX-Form angesetzt:

```
['94]:ApoJahr(NN94).NUM

[2:1]:Verhältnis(Integerzahl(2),Integerzahl(1)).NUM
```

```
[12/45]:Folge(Integerzahl(12),Integerzahl(45)).NUM

[1913/14]:Jahreszahlfolge(1913,14).NUM

[60.]:Ord(Integerzahl(60)).ORD

[13,9%]:Prozentzahl(Dezimalzahl(13,9)).NUM

[+21,9%]:Prozentzahl(signDezimalzahl(21,9)).NUM

[08/28/94]:Datum(08,28,94).NUM
```

Als Lemmakategorie wird bei den Ordinalzahltypen ORD und bei den anderen Typen NUM ausgegeben.

3.4.4 Analyse von BSX-Formen

Formen, die nur aus Buchstaben und Zeichen bestehen, trifft man im Korpus in erster Linie als Bindestrichwörter, apostrophierte Formen und Abkürzungen mit Punkt an. Daneben gibt es lexikalisierte Vorkommen bei Eigennamen (*M&M, C&A,* usw.) und Reihungen, die durch "/" abgetrennt sind (*afp/ap/dpa/Reuter, Halifax/Tokio,* usw.). Der Bindestrich wird bei der Analyse mit höherer Priorität vor dem Apostroph behandelt. Durch Bindestrich können beliebige andere BSX-Formen (ohne Bindestrich) verknüpft werden, die dann als Bestandteile des Bindestrichworts bzw. als ein Koordinationsglied im Falle des Koordinations-Bindestrichs analysiert werden. Da sich die Phänomene der Bindestrichformen bei BSX- Formen und BZSX-Formen nicht unterscheiden, und die Bindestrichkonstruktionen eine derart häufige Form darstellen, werden diese in Abschnitt 3.4.7 auf Seite 116 gesondert behandelt. Nach dem Bindestrich folgt in der Priorität der Apostroph. Auch hier macht es keinen Sinn Apostroph-Formen getrennt nach BSX- und BZSX-Formen zu behandeln. Eine ausführliche Darstellung der Apostroph-Formen wird in Abschnitt 3.4.8 auf Seite 120 erfolgen.

```
BSX-Form -> BindestrichForm

BSX-Form -> ApostrophForm

BSX-Form –> SlashForm

BSX-Form -> Abkürzung

BSX-Form -> Eigenname
```

Die Abkürzungen und Eigennamen werden wie die entsprechenden B-Formen (vgl. Abschnitt 3.3.1.4 auf Seite 104 bzw. Abschnitt 3.3.1.5 auf Seite 105) im Eigennamen- bzw. Abkürzungslexikon nachgeschlagen und entsprechend lemmatisiert. Ist die BSX-Form eine B-Form mit Punkt am Ende, so kann man mit ziemlicher Sicherheit davon ausgehen, daß es sich um eine Abkürzung handelt. Bleibt in diesem Fall der Look-up im Abkürzungslexikon erfolglos, so muß überprüft werden, ob es sich um eine komplexe Form handelt, deren Hinterglied abgekürzt wurde (*Verlagsverz., Rumfordstr., Ver-*

triebsges., usw.). In diesem Fall wird versucht die Abkürzung am Ende zu identifizieren. Handelt es sich um eine MB-Form, so gibt die Groß-/Kleinschreibung schon einige Hinweise auf den möglichen Beginn der Abkürzung. Handelt es sich um eine WB-Form, so wird von rechts nach dem shortest-match-Verfahren vorgegangen. Konnte eine Abkürzung am Ende gefunden werden, so wird das Vorderglied wie bereits bei der Kompositaanalyse (vgl. Abschnitt 3.3.1.2 auf Seite 100) beschrieben, analysiert. Wie das Beispiel *Rumfordstr.* zeigt, kann es sich beim Vorderglied auch um einen Eigennamen handeln. Es muß also beim Vorderglied auch der Look-up im Eigennamen-Lexikon erfolgen. Typischerweise tauchen Konstruktionen der Form *Eigenname + Abkürzung* mit ganz bestimmten Abkürzungen wie *Str.* (Straße), *Pl.* (Platz), *W.* (Weg) usw. auf.

```
BSX-Form -> B-Form Abkürzung
```

Der Slash (“/”) zur Kennzeichnung von Reihungen kann mit den unterschiedlichsten Elementen (wie im vorigen Abschnitt bereits für Ziffernfolgen beschrieben) vorkommen (*CDU/CSU, 1,79 m/62 kg, Matti Nykänen/Finnland, Mülheim/Ruhr, Neue Pinakothek/Graphische Sammlung,* usw.). Ohne genauere Kontextanalyse ist aufgrund dieser Vielfalt von Reihungsmöglichkeiten in der Regel keine Spezifikation der Trennerfunktion möglich. Wie die obigen Beispiele zeigen, können durch “/” sowohl einzelne Formen als Elemente der Folge abgetrennt werden (*afp/ap/dpa/Reuter, CDU/CSU*), als auch Einheiten, die aus mehr als einer Form bestehen (*1,79 m/62 kg, Neue Pinakothek/Graphische Sammlung*). Im zweiten Fall hat der Slash innerhalb der durch Blanks begrenzten Form selbst keine Funktion, hier muß eine Aufspaltung in verschiedene A-Formen erfolgen, die aber in diesem Rahmen ohne genauere Kontextanalyse nicht möglich ist. Folgen, die lediglich einzelne Formen enthalten, können selbst wieder Bestandteile komplexerer Formen sein (*CDU/CSU-Fraktion*), eine Spaltung in mehrere A-Formen würde hier zu falschen Ergebnissen führen. Um jedoch nicht völlig unsinnige Kombinationen wie *Pinakothek/Graphische* aus dem vorigen Beispiel als eine Form zu lemmatisieren, werden Slash-Konstruktionen im nicht numerischen Bereich nur dann als eine Form lemmatisiert, wenn sie Bestandteil komplexerer Formen sind, das heißt, wenn sie außer “/” noch weitere Zeichen enthalten. Wenn dies nicht der Fall ist, wird bereits bei der Tokenisierung getrennt. Das hat zwar zur Folge, daß Konstruktionen wie *CDU/CSU* je nach Kontext als eine oder als mehrere Formen analysiert werden, das ist jedoch einer Analyse unsinniger Kombinationen als eine Form vorzuziehen. Für Fälle, wie den eben erwähnten, in denen es sich um eine stark lexikalisierte Reihung handelt, ist auch die explizite Auflistung in einem Ausnahmelexikon möglich.

Bei Slash-Konstruktionen, die in komplexeren Formen vorkommen, werden die Komponenten der Folge ähnlich wie bei Bindestrich-Wörtern einzeln lemmatisiert. Als Bezeichner für die Lemmaform wird, wie bereits bei den ZS-Formen auch die neutrale Bezeichnung *Folge* gewählt.

3.4.5 Analyse von BZX-Formen

Bei der Analyse der BZX-Formen, der Formen, die aus Ziffern und Buchstaben bestehen, gibt es zwei prinzipiell verschiedene Fälle zu beachten. Zum einen die idiosynkratischen Konstruktionen, die nur mit einer bestimmten Zahl auftreten können, und zum anderen die produktiven Konstruktionen, die mit beliebigen Zahlen, häufig auch mit den entsprechenden Zahlwörtern, auftreten können. Zu den idiosynkratischen Konstruktionen gehören Namen wie *Pro7, Audi100, K2* aber auch Abkürzungen wie *3D*, G*7*. Bei den Abkürzungen gibt es neben den völlig feststehenden Ausdrücken auch halbproduktive Konstruktionen, deren Ziffernkomponente aus einer bestimmten Menge oder aus einem bestimmten Zahlbereich gewählt werden kann, wie *A<n>*, für Autobahnen, wo <n> die Nummer einer Autobahn sein muß, entsprechendes für Bundesstraßen und *B* außerdem kann *A[0-6]* für Papierformate stehen oder *B[1-5]* für den entsprechenden Sender des Bayerischen Rundfunks. Diese Beispiele zeigen, daß die Bereiche, aus denen die Zahlen gewählt werden können, unterschiedlich umfangreich sein können. Falls die Menge der möglichen Zahlen relativ klein ist, wie im Fall der zweiten Beispiele, empfiehlt sich die explizite Auflistung der Kombinationen als Eigennamen (hier sind vor allem auch Typennamen wie *R65, R80, R100* betroffen) oder Abkürzungen. Ist der Bereich relativ groß, so wird die Basis wie bei den produktiven Formen behandelt und zusätzlich die Bereichsrestriktion im Lexikon angegeben.

```
BZX-Form -> Eigenname

BZX-Form -> Abkürzung
```

Die produktiven Formen sind in der Regel Kombinationen mit Affixen, die eine numerische Basis erfordern, wie sie für den Fall von Zahlwörtern schon in Abschnitt 3.3.1.3 auf Seite 102 beschrieben wurden (*100prozentig, 12ender, ver12fachen*, usw.). Für diese Fälle gelten in der Regel dieselben Gesetzmäßigkeiten, wie sie bereits für die Zahlwörter beschrieben wurden.

```
BZX-Form -> NUM ZSuffix

BZX-Form -> ZPräfix NUM

BZX-Form -> ZZirkumfix_anfang NUM ZZirkumfix_ende
```

3.4.6 Analyse von BSZX-Formen

Den größten Anteil unter den BSZX-Formen, also den Formen, die sowohl Buchstaben, Zahlen als auch Zeichen enthalten, stellen die Bindestrichformen: *5000-Seelen-Dorf, 0:3-Rückstand, 90-m-Sprung, DIN-A4-Blatt, PRO-7-Produkt, 16er-Liga*, usw., die in Abschnitt 3.4.7 auf Seite 116 ausführlich dargestellt werden. Daneben gibt es eine Reihe von Apostroph-Formen vom Typ *ISPO'94*, die ebenfalls gesondert in Abschnitt 3.4.8 auf Seite 120 behandelt werden. Neben diesen Konstruktionen tauchen vereinzelt Eigennamen (*Auto'94&Greger*) auf, auch Abkürzungen sind denkbar, im Korpus allerdings nicht belegt. Die einzige produktive Konstruktion diesen Typs,

neben Bindestrich- und Apostroph-Formen, die im Korpus belegt ist, bezieht sich auf das Suffix *-prozentig*, indem *prozent* durch das Prozentzeichen *%* ersetzt werden kann (*20%ig*). In diesem Fall wird *-%ig* wie die entsprechenden Suffixe bei den Zahlkonstruktionen (Abschnitt 3.4.5 auf Seite 115) behandelt.

```
BSZX-Form -> BindestrichForm

BSZX-Form -> ApostrophForm

BSZX-Form –> SlashForm

BSZX-Form -> NUM ZSuffix

BSZX-Form -> Eigenname

BSZX-Form -> Abkürzung
```

3.4.7 Bindestrich-Formen

Der Bindestrich kommt neben den bereits in Abschnitt 3.4.3 auf Seite 110 beschriebenen Funktionen als Vorzeichen und in Differenzangaben, vor allem in zwei Funktionen vor: im Inneren von BSX- und BSZX-Formen als Konnektor der verschiedenen Bestandteile von Bindestrichwörtern, und am linken bzw. rechten Rand von BSX- und BSZX-Formen (in seltenen Fällen auch von ZSX-Formen) als Koordinations-Bindestrich um Gapping bzw. Tilgung von Wortbestandteilen bei Koordination anzuzeigen. Bei der Koordination von Bindestrichwörtern mit Tilgung fallen diese beiden Funktionen zusammen. Neben der Koordination kann in seltenen Fällen der Bindestrich auch in Konstruktionen wie *[...] das -ova in ihrem Namen [...]* oder *-zigfach* auftreten, wo es ein Suffix als solches kennzeichnet. Diese Fälle können jedoch zunächst vernachlässigt werden.

```
BindestrichForm -> BindestrichWort

BindestrichForm -> RKoordForm

BindestrichForm -> LKoordForm

Bindestrichform -> “-” Suffix
```

3.4.7.1 Bindestrich im Innern

Der Bindestrich im Wortinnern wird zur Verknüpfung verschiedener Elemente zu einer komplexen Form verwendet. Es kann sich dabei um die Verknüpfung gleichartiger Formen (z.B. WB-Formen) oder auch um die Verknüpfung verschiedener Formtypen zu einem heterogenen Komplex handeln. Ein Spezialfall für Bindestrichkonstruktionen sind nominalisierte Phrasen oder nominalisierte Infinitiv-Konstruktionen (*in-den-April-schicken*, *das-weiß-blaue-Nase-im-Wind-ahh-und-ohh-*

Sonne-und-Seeblick-auf-dem-Starnberger-und-Ammersee-Schiffs-Vergnügen, usw.). Bei den WB-Bindestrichformen handelt es sich meistens um mehrgliedrige Komposita, die der Übersichtlichkeit halber oder wegen des Zusammentreffens von drei gleichen Vokalen mit Bindestrich geschrieben werden (*See-Elefant, Märtyrer-Blutkult, Bundesbank-Schätzung, Tante-Emma-Laden,* usw.). Stark vertreten sind auch die Eigennamen, bei denen die einzelnen Namensbestandteile durch Bindestrich verbunden werden (*Römisch-Germanisches-Museum, Kiesselbach-Platz, Alexandre-Collectio, Dow-Jones-Index,* usw.) oder Aneinanderreihungen von Eigennamen (*Salzburg-München*). Eine Besonderheit stellen die Buchstabierformen (*s-u-p-e-r*) dar, deren Bestandteile zwar keine WB-Formen sind, die aber auf eine WB-Form zurückgehen.

Bei den inhomogenen Zusammensetzungen werden durch den Bindestrich in der Regel verschiedene Formtypen miteinander verbunden (*Rathaus-CSU, Stasi-RAF, S-Klasse, Max-II-Denkmal, 100-m-Lauf, Lloyd's-Schiffahrtsdienstes, 10%-Punkte, CDU/CSU-Fraktion,* usw.). Die Komponenten solch inhomogener Bindestrichwörter können beliebigen Typs sein und beliebige Sonderzeichen (bis auf den Bindestrich selbst) enthalten. Eine ausführliche Beschreibung und Klassifikation der hier möglichen Strukturen findet sich in Schicht (1994a).

Das Ziel bei der Lemmatisierung der Bindestrichformen ist die Identifikation aller Bestandteile. Aus diesem Grund werden bei der Analyse der Bindestrichformen alle Bestandteile erneut einem Typen-Check unterworfen und entsprechend lemmatisiert, so als ob sie isoliert vorkämen. Der erste Schritt bei der Analyse ist die Aufsplittung der Bindestrichform in ihre Bestandteile. Wie bei den Komposita wird auch bei den Bindestrichformen für das letzte Element ein anderes Look-up-Verfahren angewendet wie für die Vorderglieder. Da sich WB-Bindestrichformen bis auf die Phrasen- und Satzkonstruktionen, die jedoch die Ausnahme darstellen, wie normale Komposita verhalten wird hier nach demselben Verfahren vorgegangen. Der einzige Unterschied ist, daß an jeder Position auch Komposita auftreten können, auf die sich dann die entsprechenden Bedingungen für Vorder- bzw. Hinterglieder übertragen. Bei nicht-WB-Bestandteilen von Bindestrichwörtern entfallen diese Tests. Die Lemmaform für Bindestrichwörter ist die Bindestrichverknüpfung der Lemmaformen der Bestandteile. Die Lemmakategorie ist die Kategorie des Hinterglieds, dazu wird noch die kategoriale Zerlegung annotiert. Unter kategorialer Zerlegung sind die Wortarten der einzelnen Bestandteile zu verstehen. Ist ein Bestandteil Wortart-ambig, so werden in der Annotation die verschiedenen Möglichkeiten durch “+” getrennt. Ist ein Bestandteil komplex, so wird dessen kategoriale Struktur als Wortart eingesetzt.

```
LLoyd's-Versicherungsgesellschaft:
lloyd's-versicherungs||gesellschaft
  .fem(NS0,NP3):neF:geF:deF:aeF##N,N###EN-N,N
```

Falls bei diesem Verfahren ein Bindestrichwort nicht erkannt wird, so wird unterschieden, ob die Bestandteile WB-Formen oder andere Formen sind. Bei WB-Formen wird versucht, die nicht identifizierbaren Bestandteile ohne Vorder- bzw. Hintergliedbe-

schränkung als WB-Form zu lemmatisieren, da es sich dann auch um eine Satz- oder Phrasenbindestrichform handeln kann. Bei anderen Formen wird bei Vordergliedern die Kategorie UK in der Zerlegung angenommen. Die oben erwähnten Buchstabierformen sind aufgrund ihrer charakteristischen Form (/.(-.)*/) leicht von normalen Bindestrichformen zu unterscheiden und lassen sich schon vor der eigentlichen Analyse bestimmen.

Neben diesen kompositionell analysierbaren Formen, bei denen es hinsichtlich der Kombinierbarkeit kaum Restriktionen gibt, existieren auch lexikalisierte Bindestrichwörter, deren Einzelbestandteile keine sinnvollen Wörter sind und die demzufolge auch in keinem Lexikonmodul nachzuschlagen sind, wie *Tam-Tam*. Für solche Formen muß auch hier wieder ein Ausnahmelexikon geführt werden. Weitere lexikalisierte Bindestrichformen sind Namen und Abkürzungen, die wie in Abschnitt 3.3.1.4 auf Seite 104 und Abschnitt 3.3.1.5 auf Seite 105 beschrieben, analysiert werden.

3.4.7.2 Koordinations-Bindestrich

Hat der Bindestrich die Funktion, eine Tilgung bei Koordination anzuzeigen (er wird in dieser Funktion auch Ergänzungsstrich genannt), so ist das Ziel der Formanalyse die Rekonstruktion der getilgten Bestandteile aus den Elementen des Kontexts. Je nach dem, ob es sich um einen linksseitigen oder rechtsseitigen Bindestrich handelt, muß das entsprechende Koordinationsglied vor bzw. nach der Bindestrichform gesucht werden. Bei Koordination mehrgliedriger Komposita tauchen auch Fälle beidseitiger Tilgung auf, wie in *Vogelbrut- und -rastgebiet.*

3.4.7.2.1 Linksseitiger Koordinations-Bindestrich

Die Auflösung eines linksseitigen Koordinations-Bindestrichs ist vergleichsweise einfacher als der rechtsseitige Fall, da die entsprechende Bezugsform schon vorher im Satz stehen muß und deshalb deren Analyse bereits zur Verfügung steht. Die Belege im Korpus haben gezeigt, daß die Bezugsform oder eine weitere Form mit linksseitigem Koordinationsbindestrich in den meisten Fällen innerhalb eines Fensters der Größe 5 vom aktuellen Wort an rückwärts gefunden werden kann. Die Bezugsform kann sowohl ein komplexes Wort als auch ein Bindestrichwort sein. Ist die Bezugsform ein Bindestrichwort, so kann davon ausgegangen werden, daß die zu rekonstruierende Form mit einer Bindestrich-Komponente des Bindestrichworts übereinstimmt, wobei es keine prinzipiellen Beschränkungen über den Typ der zu rekonstruierenden Form gibt. Ist die Bezugsform eine komplexe Form, so ist die zu rekonstruierende Form eine Komponente der komplexen Form, die nicht notwendig Wortstatus haben muß.

Beispiele:

Europa-Skepsis und -Verdrossenheit
lebenswertem und -unwertem Leben

Mineralientausch und -bestimmung
Kriegsopferversorgung und -fürsorge
Geschwisterspott und -neid und -spiele
DDR-Richter und -Postkontrolleure
Schicki-Micki-Lokalen und -Geschäften

Die Analyse linksseitiger Bindestrichformen erfolgt in zwei Schritten: zuerst wird die Bindestrichform (ohne den Bindestrich) nach den Regeln für den entsprechenden Formtyp analysiert. Handelt es sich um eine WB-Form oder ist zumindest die letzte Komponente der Analyse eine WB-Form, so wird wie im Fall komplexer Formen überprüft, ob es sich um ein mögliches Hinterglied handelt. Im zweiten Schritt, dem Rekonstruktionsschritt, wird in der Liste der bisher analysierten Formen zurückgegangen, bis ein Eintrag gefunden wird, der entweder als komplexe Form oder als Bindestrich-Wort analysiert werden konnte, maximal jedoch fünf Schritte.[1] Bei dieser Form wird überprüft, ob die Kategorie mit der der Bindestrichform kompatibel ist. Es ist zwar prinzipiell möglich, daß bei der Koordination die Koordinationsglieder unterschiedlichen Wortarten angehören, dieser Fall ist hier jedoch wegen seiner geringen Häufigkeit nicht berücksichtigt. Handelt es sich bei der Bezugsform um ein zweigliedriges Bindestrichwort oder Kompositum, so wird als Vorderglied der Bindestrichform das Erstglied der Bezugsform rekonstruiert, auch wenn es sich um ein Bindestrich-Wort mit komplexem Erstglied handelt. Handelt es sich bei der Bezugsform um ein mehrgliedriges Bindestrichwort oder Kompositum, so läßt sich in der Regel ohne weitere Informationen nicht entscheiden, welche Anfangssequenz eine korrekte Rekonstruktion darstellt. In diesem Fall werden alle Anfangssequenzen als mögliche Vorderglieder rekonstruiert.

3.4.7.2.2 Rechtsseitiger Koordinations-Bindestrich

Der rechtsseitige Koordinations-Bindestrich ist insofern schwieriger zu behandeln, als die Bezugsform erst in der Folge analysiert werden kann und zum Zeitpunkt der Analyse der defektiven Form noch nicht bekannt ist. Weiterhin kommt hinzu, daß die kategoriale Übereinstimmung der Hinterglieder bei Tilgung mit Koordinations-Bindestrich weitaus stärker ist als die Übereinstimmung der Vorderglieder. Das erweist sich bei der Auswahl des Bezugselements als Nachteil, da unter Umständen eine falsche Form als Bezugselement gewählt wird. Ansonsten gilt für die rechtsseitigen Koordinations-Bindestrich-Konstruktionen dasselbe wie für die linksseitigen. Während jedoch bei den linksseitigen Koordinationstilgungen die verbleibenden Formen in der Regel Wortstatus haben, ist der Fall, daß es sich um Affixe handelt, bei der rechtsseitigen Tilgung wesentlich gebräuchlicher (...*diesen neuen Belag be- oder abnutzt*...). Dieser Unterschied spielt jedoch für die Analyse keine Rolle.

1. Der Algorithmus geht davon aus, daß die erste komplexe Form bzw. das erste Bindestrichwort auch die wahrscheinlichste Quelle zur Rekonstruktion ist. Nur wenn die Kategorien völlig unverträglich sind, wird weitergesucht.

Die Grundidee bei der Analyse von Formen mit rechtsseitigem Koordinations-Bindestrich ist, zuerst die Form (ohne Berücksichtigung des Bindestrichs) als Vorderglied zu analysieren. Diese Analyse wird auf jeden Fall in die Liste der Analysen der A-Formen eingetragen. Dazuhin kommt diese Analyse zusammen mit dem Index im Satz auf eine Warteliste. Wird in der Folge eine Form als komplexe Form oder als Bindestrich-Wort analysiert, so werden die Elemente der Warteliste nacheinander auf die Rekonstruierbarkeit der fehlenden Elemente im Hinblick auf die komplexe oder Bindestrich-Form überprüft und falls die Rekonstruktion möglich ist, wird der Eintrag in der Liste der Analysen der A-Formen am entsprechenden Index revidiert und der Eintrag in der Warteliste gelöscht.

3.4.7.2.3 Mehrfache Tilgung bei Koordinations-Bindestrich

Vergleichsweise selten kommen auch mehrfache Tilgungen bei Koordination vor. Das erste Koordinationsglied weist dann einen rechtsseitigen Koordinations-Bindestrich auf und das letzte Koordinationsglied einen linksseitigen (*Vogelbrut- und -rastgebiet, Rover-600- und -800-Modellen*, usw.). Es kann sich dabei sowohl um zu rekonstruierende Bindestrich-Formen mit beliebigen Konstituententypen, als auch um Komposita handeln. In diesem Fall müssen beide Rekonstruktionsverfahren angewendet werden. Es ist lediglich darauf zu achten, daß die rekonstruierten Formen nur Bestandteile der jeweiligen Bindestrich-Form und nicht der bereits rekonstruierten Form sein dürfen, um Rekonstruktionen der Art *Rover-600-800-Modellen* oder *Vogelbrutrastgebiet* zu verhindern.

3.4.8 Apostroph-Formen

Unter den Mischformen bilden die Formen mit Apostroph eine weitere Gruppe, die in diesem Rahmen einer genaueren Betrachtung unterzogen werden. Es soll in diesem Abschnitt nur um solche Vorkommen des Apostrophs gehen, wo durch Apostrophierung eine Auslassung oder spezielle Verknüpfung realisiert wird. Die zweite Funktion von Apostrophen oder Hochkommas im Text ist die Hervorhebung oder Zitierform bestimmter Satzteile oder auch ganzer Sätze. Auf diese Fälle soll hier nicht eingegangen werden. Formen, die beidseitig durch Apostrophen begrenzt sind, werden bereits bei der Tokenisierung als Klammerstrukturen erkannt. Werden mehrere Formen durch Apostroph “geklammert”, werden die Apostrophen bereits während der Tokenisierung mit einer speziellen Klammerungsauflösungsroutine isoliert. Es kann also davon ausgegangen werden, daß Hochkommas in X-Formen tatsächlich Apostroph-Funktion haben. Apostrophierte Formen können selbst wieder Bestandteil komplexer BSZX- oder BSX-Formen sein, beispielsweise in Bindestrichkonstruktionen. Apostrophierte Formen selbst enthalten in der Regel keine weiteren Zeichen als Buchstaben oder Ziffern. Eine Ausnahme stellen lediglich Abkürzungen und Eigennamen mit Abkürzungen dar, bei denen aber immer ein ganz bestimmter Apostrophiertyp realisiert ist. Die Apostrophierung von Zahlen beschränkt sich, abgesehen von feststehenden Namen, auf Jahresangaben (*’93, Bündnis’90*), die in Abschnitt 3.4.3 auf Seite 110 beschrieben sind. So kann im folgenden davon ausgegangen werden, daß es sich bei Apostroph-

Formen um BSX-Formen handelt. Es können folgende Apostroph-Funktionen auftreten:

1. Auslassung von Zeichen sowohl am Wortanfang, in der Wortmitte als auch am Wortende: *'nauf, G'müs, heil'ge, geh'*, usw.
2. Verbindung zweier Worte unter Auslassung von Zeichen (in diesem Fall ist der Apostroph notwendigerweise nicht am Rand der Form): *weil's, Rock'n'roll, auf'n,* usw.
3. Markierung morphosyntaktischer Funktionen (Genitiv oder Plural) bei Eigennamen, Fremdwörtern und Abkürzungen bzw. bei der Derivation von Eigennamen: *CD's, Boris', Beck'sche,* usw.

Die Lemmatisierung der Fälle unter 3 ist relativ einfach. Prinzipiell kann der 3. Fall nur dann eintreten, wenn der Apostroph nicht am Formanfang steht. Die morphosyntaktische Funktion tritt in drei verschiedenen Fällen auf: Eigenname oder Abkürzung, die auf Sibilant enden und Apostroph am Wortende (*Boris', PDS'*), Eigenname und Abkürzung gefolgt von Apostroph, gefolgt von "*s*", selten auch *"en"*, (*AG's, Lloyd's*) und im letzten Fall in der Konstruktion Eigenname, Apostroph und eine Flexionsform des EN-Suffix *sch* (*Beck'sche, Weimar'sche)*.

Der zweite Fall beschränkt sich mit Ausnahme lexikalisierter Formen wie *Rock'n'roll* auf Kombinationen von Verben, Konjunktionen und Präpositionen mit *das, es, sie, den* (abgekürzt dann als *s* bzw. *n*), wobei auch der Fall eintreten kann, daß das Vorderglied abgekürzt wird (*wär's, seh's,...*).

Der erste Fall betrifft teilweise lexikalisierte Formen wie *'nauf* wo mehr als ein Buchstabe getilgt wird. Diese Formen müssen explizit aufgelistet werden: zusammen mit der ausgeschriebenen Version und deren Kodierung bilden sie ein Ausnahmelexikon. In anderen Fällen kann in der Regel davon ausgegangen werden, daß es sich um u- oder e-Tilgung handelt (*über'n, über's, gäb's, G'müs, z'sammen*, usw.). Im Bereich der Eigennamen und Fremdwörter kommen abgekürzte Namen der Form *L'Ermitage, L'Humanité, D'Annunzio, O'Connor,* usw. vor, die auch unter diese Rubrik fallen.

Beim Look-up der Apostroph-Formen wird nach den Stellungsvarianten Anfang, Mitte und Ende unterschieden.

```
ApostrophForm -> LApostrophForm|MApostrophForm|
  RApostrophForm
```

3.4.8.1 Apostroph am Anfang

Zuerst wird überprüft, ob es sich bei der Form um eine lexikalisierte Form (*'naus, 'tschuldigung*, usw.) handelt oder ob es ein Eigenname ist. Ist dies nicht der Fall, so muß es sich um eine komplexe Form handeln, deren erste Komponente eine lexikalisierte Form ist.

```
LApostrophForm -> lexikalisierteApostrophForm

LApostrophForm -> Eigenname

LApostrophForm -> lexikalisierteApostrophForm
  einfacheForm|komplexeForm
```

Lemmaform ist im Fall der Konstruktionen mit lexikalisierten Apostroph-Formen die vollständige Form, die im Ausnahmelexikon verzeichnet ist. Bei Namen ist die Lemmaform identisch mit der Apostroph-Form.

3.4.8.2 Apostroph in der Mitte

Auch in diesem Fall wird zuerst überprüft, ob es sich bei der Apostroph-Form um eine lexikalisierte Form handelt (*Wies'n, Rock'n'Roll,* usw.). Außerdem muß der Fall von Eigennamen mit *D', L',* O', *Dell'* und *All'* überprüft werden. Das heißt, es wird sowohl geprüft, ob die Apostroph-Form als ganzes oder nur die Komponente nach dem Apostroph im Eigennamenlexikon ist. Der Fall, daß es sich bei der Apostroph-Form um einen Eigennamen oder um eine Abkürzung handelt, wird auch bei beliebiger Anfangskomponente in jedem Fall überprüft. Bei Namen findet sich auch häufig die Kombination mit Jahreszahlen (*Boot'90, ISPO'94,* usw.). Sie sind daran zu erkennen, daß der Apostroph an drittletzter Stelle steht, gefolgt von zwei Ziffern. Da es sich hierbei oft um jährlich wiederkehrende Ereignisse handelt (Ausnahme: *Bündnis'90*), macht es wenig Sinn, diese Kombinationen explizit im Namenslexikon aufzunehmen. Stattdessen werden die Vorderglieder (zusammen mit dem Apostroph) als EN-Präfixe zu Zahlen, mit der Beschränkung auf zwei Ziffern, kodiert. Als Lemmaform gilt in diesen Fällen, wenn im Lexikon nicht explizit eine andere Form (*heil'ge* zu *heilige*) kodiert ist, die Apostroph-Form.

Bei Eigennamen und Abkürzungen muß auch der Fall der Apostrophierung bei der Bildung der Genitiv- oder Pluralform mit *s* überprüft werden. Als Lemmaform wird in diesem Fall die Grundform angesetzt, und die Lemmaklasse um ein Merkmal für "Genitiv oder Plural" ergänzt.

Außer den lexikalisierten Fällen (als Apostroph-Form lexikalisiert oder Eigenname bzw. Abkürzung), gilt es in erster Linie festzustellen, ob es sich um die Verknüpfung von zwei Formen oder um eine zugrundeliegende Form handelt. Es kann aufgrund des Korpusmaterials davon ausgegangen werden, daß diese Fälle nur dann auftreten, wenn die Form außer dem Apostroph nur Buchstaben enthält. Wenn es sich nicht um die Verknüpfung zweier Formen handelt, so kann es sich nur um die Auslassung von Buchstaben handeln. Systematische Auslassung betrifft *e* oder *u*. Die Auslasssung anderer Buchstaben ist nur in lexikalisierten Fällen wie *heil'ge* anzutreffen. Die Auslassung von *u* ist eingeschränkt auf das Präfix *zu*. Formen mit Apostroph im Wortinnern werden auf die Möglichkeit eines ausgelassenen *e* und, falls der Apostroph nach *z* kommt auch auf die Möglichkeit eines ausgelassenen *u* hin überprüft. Dazu wird bei der Apostroph-Form anstelle des Apostrophs ein *e* bzw. *u* eingefügt, und die so ent-

standene WB-Form nach dem in Abschnitt 3.3.1 auf Seite 98 beschriebenen Verfahren lemmatisiert, wobei der Fall des Eigennamens oder der Abkürzung in der Regel ausgeschlossen werden kann. Die Lemmaform ist die Lemmaform, die die Lemmatisierung der WB-Form liefert. Bei der Auslassung von *u* ist der Übergang zum Apostroph, der zwei Formen trennt, manchmal fließend: *z'sammen, z'viel, z'wenig, z'schnell.*

Handelt es sich um die (nicht lexikalisierte) Verknüpfung zweier Wörter, so ist die Komponente hinter dem Apostroph *s* oder *n* (selten auch *ne, nen, nem, ner*). Die erste Komponente kann entweder ein Verb oder ein Funktionswort sein. Im Falle von Präpositionen ist für *s* nur die Interpretation als definiter Artikel *das* möglich. In den anderen Fällen kann *s* sowohl als Pronomen *es* oder *sie* analysiert werden. Für *n* bietet sich die Analyse als definiter Artikel *den* an. Theoretisch ist auch die Interpretation als *ein* denkbar, dieser Fall ist im Korpus jedoch nicht belegt. Bei den Endungen *ne, nen, nem, ner*, die hier aufgrund mangelnder Belege nicht berücksichtigt werden, handelt es sich um die entsprechenden Flexionsformen von *ein*.

```
MApostrophForm -> lexikalisierteApostrophForm

MApostrophForm -> {D'|L'|O'|Dell'|All'} Eigenname

MApostrophForm -> EN_NUM_Präfix jahresZahl

MApostrophForm -> Eigenname|Abkürzung "'s"

MApostrophForm –> einfacheForm|komplexeForm

MApostrophForm –> Präposition "den"

MApostrophForm –> Verb|Konjunktion|Adverb|DetPro
  "es"|"sie"|"den"
```

3.4.8.3 Apostroph am Ende

Am Wortende tritt ein Apostroph bei Eigennamen und Abkürzungen auf, die auf Sibilanten enden um den Genitiv (evtl. auch Plural) zu markieren.[1] Hier wird aus phonologischen Gründen kein *s* realisiert. Es muß also in diesem Fall lediglich überprüft werden, ob das Wort vor dem Apostroph ein Eigenname oder eine Abkürzung der entsprechenden Form ist. Als Lemmaform wird die entsprechende Grundform angesetzt, und die morphosyntaktischen Merkmale (im unmarkierten Fall nur *X*) durch ein Genitiv-Merkmal ersetzt. Eine weitere Analyse für Formen, die mit Apostroph enden ist die

1. Der Fall der lexikalisierten Apostroph-Formen kann hier auf Namen und Abkürzungen beschränkt werden, da im Korpus keine anderen Vorkommen belegt sind. Die Benutzung des Apostroph zur Markierung des Genitivs (angelehnt an die englische Schreibweise) wird jedoch auch im Deutschen in zunehmenden Maße (fälschlicherweise) üblich. Diese Fälle werden bislang noch nicht berücksichtigt. Bei Texten, in denen diese Schreibweise häufig auftaucht, empfiehlt sich aber auch eine Berücksichtigung dieser Fälle.

Auslassung von *e*. Zu diesem Zweck wird auch bei Apostroph am Ende wieder überprüft, ob eine Ersetzung des Apostrophs durch *e* eine Analyse als einfache oder komplexe Form erlaubt.

```
RApostrophForm -> Eigenname|Abkürzung

RApostrophForm -> einfacheForm|komplexeForm
```

3.5 Disambiguierungsstrategien

Wie die Ausführungen in Abschnitt 3.3 und Abschnitt 3.4 gezeigt haben, können alle Typen von A-Formen auf verschiedene Analysemöglichkeiten überprüft werden. Werden bei allen Formen alle Möglichkeiten ohne Restriktionen überprüft, so führt das bezogen auf das Testkorpus zu durchschnittlich 2,3 möglichen Analysen pro Form (bezogen auf die A-Formen), wobei 62% der A-Formen mehr als eine Analyse erhalten und immerhin 6,8% der Formen 10 oder mehr mögliche Analysen haben. Diese Menge von (zum Teil auch unsinnigen) Analysen gilt es soweit wie möglich einzuschränken, und das wenn möglich bereits während der Wortformenanalyse. In diesem Kapitel werden zuerst gängige Disambiguierungsstrategien vorgestellt, danach folgt eine Analyse der im Testkorpus auftretenden Ambiguitäten, aufgrund derer dann im folgenden verschiedene Möglichkeiten der Disambiguierung diskutiert werden.

Bei den Disambiguierungsstrategien kann man einerseits linguistisch motivierte Strategien (auch als regelbasierte Verfahren bezeichnet) von statistisch orientierten Verfahren unterscheiden und andererseits Verfahren, die wortbezogen vorgehen und solche die kontextbezogen vorgehen (auch als satzbezogene Verfahren bezeichnet). Des weiteren können Verfahren, die die Menge der Möglichkeiten soweit wie möglich einschränken, von solchen unterschieden werden, die nur eine Analyse zulassen.

3.5.1 Disambiguierung im System SALEM

Von den in Abschnitt 3.1.2 auf Seite 88 vorgestellten Verfahren, hat lediglich das Saarbrücker Lemmatisierungsverfahren SALEM eine umfangreichere Disambiguierungskomponente. Zur Homographenauflösung gibt es beim Saarbrücker Verfahren zwei verschiedene Varianten : eine linguistisch motivierte Auflösung aufgrund des Kontexts (vgl. Weber 1976) und eine statistisch orientierte Methode (vgl. Eggers 1980). In beiden Fällen werden lediglich durch den Look-up im Lexikon verursachte Mehrdeutigkeiten aufgelöst. Da das Verfahren keine Kompositazerlegung vornimmt und auch keine Analyse von Sonderformen stattfindet, die häufig mehrere Interpretationen zulassen, sind derartige Mehrdeutigkeiten im Saarbrücker Verfahren von vornherein ausgeschlossen.

3.5.1.1 Der kontextbezogene Ansatz

Beim kontextbezogenen Ansatz werden die Mehrdeutigkeiten, die vom Programm behandelt werden können, nach verschiedenen Homographietypen (Homographie mit satzeinleitenden Wortklassen, mit Auxiliaren, mit Modifikatoren, mit verbalen Flexionsformen und mit nominalen Flexionsformen) getrennt behandelt. Das Ziel ist es, die Menge der möglichen Analysen einzuschränken, wobei nur mit Sicherheit als falsch erkannte Lösungen ausgeschlossen werden. Homographien, die nicht zu einem der obigen Typen gehören, werden nicht behandelt.

3.5.1.2 Der statistische Ansatz

Der in Eggers (1980) beschriebene Ansatz hat eine Gewichtung aller möglichen Ketten von Lemmata, die dem Satz zugeordnet werden können zum Ziel. Diese Gewichtung geschieht durch Aufsummierung der Gewichte der einzelnen Kategorienpaare. Die Gewichte für die Abfolge von Kategorienpaaren wurden durch die Auswertung eines Referenzkorpus mit ca. 5000 Sätzen gewonnen. Für einige hochgradig mehrdeutige Wortformen (z. B. *bis*) gibt es spezielle Tests, die aufgrund bestimmter lexikalischer Elemente im Kontext eine Disambiguierung vornehmen. Die gewichteten Satzketten sind die Eingabe für die syntaktische Analyse, wobei die Kette mit der höchsten Gewichtung zuerst analysiert wird. Das Verfahren bricht ab, sobald einer Kette eine syntaktische Analyse zugeordnet werden kann. Die Gewichtung der Ketten stellt also lediglich eine Vorsortierung dar.

3.5.2 Linguistisch motivierte Verfahren - Lokale Grammatiken

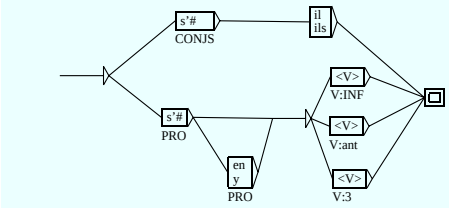
Als Beispiel für ein linguistisch motiviertes Verfahren zur Disambiguierung bei der Lemmatisierung soll hier der Ansatz von Silberztein (1989) fürs Französische vorgestellt werden. Dieser Ansatz basiert auf der Repräsentation des zur Disambiguierung relevanten Kontexts einzelner Homographen oder aber auch ambiger Klassen durch endliche Automaten (genauer: Transduktoren), die lokale Grammatiken repräsentieren. Diese lokalen Grammatiken finden neben der Disambiguierung in den verschiedensten Bereichen Anwendung: Erkennung von Komposita und Mehrwortlexemen, Repräsentation orthographischer Varianten im Lexikon, Rechtschreibkorrektur sowie Überprüfung von Kongruenz, Identifikation spezieller Sinnelemente des Texts, wie zum Beispiel zur Identifikation von Zeitangaben in den verschiedensten Schreibvarianten oder allen synonymen Bezeichnungsmöglichkeiten für bestimmte Begriffe (bei Silberztein gibt es beispielsweise eine umfangreiche lokale Grammatik, die alle Bezeichnungsmöglichkeiten für *Kriminalroman* enthält), usw.

Zur Disambiguierung gibt es verschiedene Möglichkeiten:

- Transduktoren, die speziell zur Disambiguierung einer Form definiert sind. Sie enthalten zu jeder möglichen Analyse der Form die Kontextbeschreibung. Paßt der Analysekontext auf einen in der lokalen Grammatik beschriebenen Kontext, so wird als Ergebnis die entprechende Klassifizierung der Form ausgegeben. Eine weitere Disambiguierung dieser Form erübrigt sich dann. Dieses Verfahren ist allerdings relativ aufwendig und kann deshalb nicht für alle Ambiguitäten angewendet werden. Derartig umfangreiche lokale Grammatiken existieren nur für häufig auftretende lexikalische Ambiguitäten und nur unter der Voraussetzung, daß sich die zur Disambiguierung notwendigen Kontexte bestimmen lassen. Das folgende Beispiel zeigt den Automaten zur Disambiguierung der Form *s'*, die sowohl als Pronomen (*se*) als auch als Konjunktion (*si*) interpretiert werden kann.[1]

1. Die Kategorien, die in den Beispielautomaten verwendet werden sind die in Abschnitt 1.4.3 auf Seite 24ff beschriebenen Kategorien und Merkmale des DELAF.

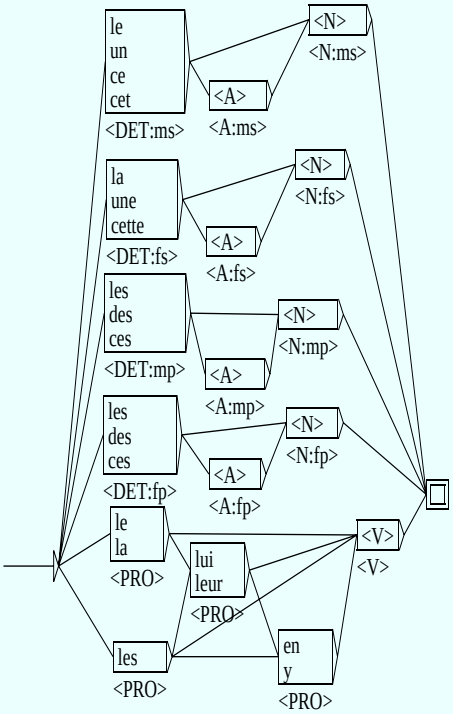
Abbildung 2. Automat zur Disambiguierung von s’



Silberztein (1989), S. 26

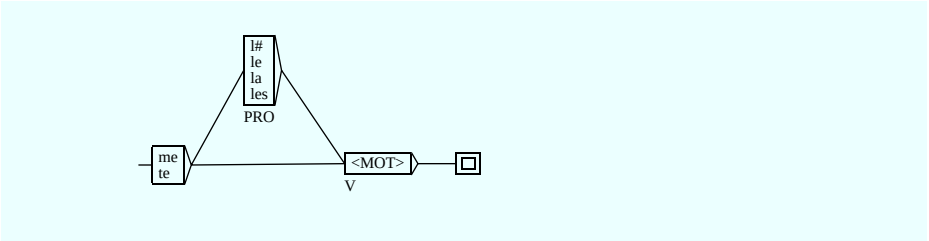
- Der Ausschluß bestimmter Analysen indem zum Beispiel auf Kongruenz überprüft wird. Die Abbildung unten zeigt ein Beispiel für einen Automaten, der die Kongruenz zwischen Determinator, Adjektiv und Nomen im Französischen beschreibt.

Abbildung 3. Automat zur Überprüfung von Kongruenz im pränominalen Bereich



- Die Auswahl bestimmter Analysen, wenn diese von einer lokalen Grammtik akzeptiert werden, ungeachtet der anderen Analysen. In diesem Fall werden nicht ambige Formen als Ausgangspunkt genommen. Falls eine ambige Form mit ihrem Kontext auf einen solchen Automaten paßt, so ist sie in diesem Kontext nicht mehr ambig. Das Beispiel zeigt den Automaten für die nicht ambigen Pronomen *me* und *te*, die es erlauben die Formen *l', le, la,* und *les* eindeutig als Pronomen und nicht als Determinatoren zu identifizieren:

Abbildung 4. Automat der Pronomen l', le, la, les



Silberztein (1989), S. 165

3.5.3 Statistische Methoden

Neben dem oben vorgestellten statistischen Ansatz zur Disambiguierung im Rahmen der Lemmatisierung gibt es in neuerer Zeit eine Vielzahl von Disambiguierungsalgorithmen, die vor allem im Kontext der Erkennung und Synthese gesprochener Sprache entwickelt wurden. Diese Systeme sind zum Teil sehr speziell auf diese Anwendung zugeschnitten, und können im Kontext der Lemmatisierung nicht eingesetzt werden. Für die Lemmatisierung interessant sind in erster Linie die Wortartentagger, die zur Disambiguierung von Wortartambiguitäten verwendet werden können.

Beim statistisch basierten Wortartentagging[1] werden unterschiedliche Informationen zur Disambiguierung benutzt. Die einfachste Möglichkeit besteht darin, nur die relative Wahrscheinlichkeit eines Tags zu betrachten und im Zweifelsfall die Kategorie mit der größten Auftretenshäufigkeit im Referenzkorpus auszuwählen. Dieses Verfahren wird jedoch in neueren Arbeiten nicht mehr als alleiniges Kriterium verwendet. Elaboriertere Verfahren betrachten nicht das isolierte Auftreten eines Elements sondern n-Gramm-Wahrscheinlichkeiten, wobei n in der Regel als 2 oder 3 gewählt wird. (DeRose 1988, Wothke u.a. 1993, u.v.m). Eine Kombination dieser beiden Strategien wird in Church (1988) vorgeschlagen. Daneben kommen auch Hidden Markov-Modelle zum Einsatz (Cutting u.a. 1992). DeRose und in Anlehnung daran auch

1. Bei dem Begriff *Wortartentagging*, der hier etwas mißverständlich sein mag, handelt es sich um die Übersetzung des englischen *part-of-speech-tagging* oder kurz *POS-tagging*. Die als *POS-Tags* verwendeten Elemente enthalten in der Regel wesentlich mehr morphosyntaktische Information als der Begriff Wortart suggeriert. Die Tagger legen teilweise ein Tagset mit mehreren Hundert verschiedenen Tags zugrunde. Diese Tags enthalten sowohl die Wortklasse, als auch die morphosyntaktischen Informationen in etwa demselben Ausmaß wie sie auch im FLEX-Lexikon verzeichnet sind.

Wothke nehmen aufgrund der n-Gramm-Wahrscheinlichkeiten eine Bewertung aller möglichen Ketten, die dem Satz entsprechen vor und wählen aus diesem Netz (ähnlich wie bereits Eggers 1980) den wahrscheinlichsten Pfad aus.

Insgesamt erweist sich der statistisch basierte Ansatz als problematisch, da:

1. umfangreiche, von Hand bearbeitete Referenzkorpora zur Verfügung stehen müssen.[1]
2. für selten auftretende Phänomene auch bei umfangreichen Referenzkorpora im Stile des Brown- oder LOB-Korpus das Belegmaterial nicht ausreicht.
3. Konfigurationen, die im Trainingskorpus nicht auftreten, überhaupt nicht behandelt werden können.
4. die durch die Trainingskorpora gewonnenen Daten stark von der Zusammenstellung des Trainingskorpus abhängen, was besonders bei der lexikalischen Disambiguierung zum Tragen kommt.
5. die Referenzkorpora im günstigen Fall Ergebnisse liefern, die mit der linguistischen Intuition übereinstimmen, also auch durch eine systematische Analyse der Ambiguitäten ohne die genannten Unsicherheitsfaktoren hätten erzielt werden können.
6. unter Umständen korrekte Analysen aufgrund der obengenannten Referenzkorpus-Problematik ausgeschlossen werden.

3.5.4 Gemischte Verfahren

Ein sehr vielversprechendes Verfahren zur Disambiguierung wurde von Tapainen und Voutilainen (1994) vorgeschlagen. Sie benutzen ein kombiniertes Verfahren aus einer Regelkomponente und einer statistischen Komponente. Die Wortformanalyse wird von dem System ENGTWOL, einem Morphologiesystem, das auf dem 2-Level-Formalismus (Koskenniemi 1983) basiert, ausgeführt. Die Ergebnisse dieser Analyse werden mit Hilfe von grammatischen Regeln gefiltert. Verbleibende Ambiguitäten wurden mit der Disambiguierungskomponente des von XEROX entwickelten Taggers XT (Cutting u.a. 1992) weiter bearbeitet. Tapainen und Voutilainen konnten durch dieses kombinierte Verfahren eine erhebliche Verbesserung der Ergebnisse gegenüber der rein statistischen Methode erreichen. Bezogen auf ein 26.711-Wort-Korpus wird durch die morphologische Analyse mit ENGTWOL eine Fehlerrate von nur 0,09 % erreicht, allerdings bei 37,6 % ambigen Wörtern. Die linguistische Disambiguierung reduziert die Ambiguitäten auf 3,6% bei einer Fehlerrate von 0,35 %. Im Gegensatz dazu verbleiben bei der Anwendung des XT-Taggers mit rein statistischer Disambiguierung nur 0,7% der Wörter ambig, allerdings bei einer Fehlerrate von 6,38 %. Das kombinierte Verfahren, bei dem zuerst mit Hilfe grammatischer Regeln disambiguiert wird,

1. Dieses Referenzmaterial steht fürs Englische mit dem getaggten Brown-Korpus (Francis/Kucera 1982) und dem getaggten LOB-Korpus (Garside/Leech/Sampson 1987) in gewissem Ausmaß zur Verfügung, fürs Deutsche fehlt ein derartig umfangreiches Referenzkorpus völlig.

bevor der XT-Tagger die verbleibenden Ambiguitäten auflöst, kommt auf eine Fehlerrate von 1,46 % bei 0,0 % verbleibenden ambigen Formen.

3.5.5 Grundannahmen bei der CISLEX-Lemmatisierung

Bei der Disambiguierung stehen sich zwei in der Praxis häufig nicht miteinander zu vereinbarende Forderungen gegenüber. Zum einen die Forderung, wirklich nur (in dem betreffenden Kontext) korrekte Analysen zuzulassen gegenüber der Forderung, nur definitiv als falsch oder nicht adäquat erkannte Analysen auszuschließen. Ein Verfahren, das beiden Forderungen voll gerecht wird, schließt meines Erachtens eine vollständige syntaktische und semantische Analyse des Satzes ein. Eine vollständige syntaktische und semantische Satzanalyse ist jedoch nach dem derzeitigen Stand der Forschung nicht machbar. So gilt es bei der Disambiguierung einen Kompromiß zwischen diesen beiden Forderungen zu finden. Die Forderung nach weitgehender Korrektheit wird dabei in unserem Fall sehr hoch bewertet, so daß rein statistische Methoden aufgrund der oben erwähnten Probleme als alleiniges Verfahren zur Disambiguierung ausscheiden. Andererseits ist eine regelbasierte Disambiguierung mit den hier zur Verfügung stehenden Mitteln nicht in der Lage, die Menge der Alternativen in allen Fällen hinreichend einzuschränken, was für eine Vielzahl von Anwendungen aber notwendig ist. Demzufolge wird hier eine ähnliche Strategie wie in (Tapainen/ Voutilainen 1994) verfolgt. Die Menge der möglichen Analysen wird zuerst mit Hilfe linguistisch motivierter Regeln soweit wie möglich eingeschränkt. Bei den verbleibenden Mehrdeutigkeiten wird versucht, diese mit Hilfe von Präferenzregeln weiter einzuschränken. So ergibt sich insgesamt eine Hierarchie von Filtern, die nacheinander angewendet werden. Durch diese Beschränkung, zusammen mit der Bedingung, daß der Filterungsprozeß abgebrochen wird, sobald keine Mehrdeutigkeiten mehr vorliegen, wird gewährleistet, daß nie alle Analyseergebnisse ausgeschlossen werden. Das Ziel sollte sein, durch eine ständige Weiterentwicklung, basierend auf der Analyse der auftretenden Ambiguitäten, den Einsatzbereich der reinen Präferenzregeln soweit wie möglich einzuschränken.

3.6 Auftretende Ambiguitäten und ihre Disambiguierung

Nach der Betrachtung und Bewertung der gängigen Disambiguierungsstrategien, sollen nun die bei der Lemmatisierung auftretenden Mehrdeutigkeiten genauer analysiert werden. Im Anschluß daran wird gezeigt, wie ein Teil dieser Mehrdeutigkeiten mit Hilfe von linguistisch motivierten Regeln eingeschränkt werden kann, und welche Heuristiken zu einer weiteren Restriktion der Menge möglicher Lemmata führen. Abschließend werden die Möglichkeiten diskutiert, das Verfahren zu verbessern und zu erweitern.

3.6.1 Auftretende Ambiguitäten

Durch den in Abschnitt 3.3 und Abschnitt 3.4 vorgestellten Algorithmus zur Analyse der Formen und die Prinzipien der Kodierung im CISLEX-DKL (vgl. Abschnitt 2.1 auf Seite 28) ergeben sich verschiedene Möglichkeiten von Mehrdeutigkeiten:

- Lexikon-bedingte Homographien
- Segmentierungs- und Rekonstruktionsambiguitäten
- Ambiguitäten durch verschiedene Look-up-Alternativen
- spezielle funktionale Ambiguitäten bei Sonderformen

Die Probleme, die mit diesen Typen von Mehrdeutigkeiten verbunden sind, sollen im folgenden anhand von Beispielen dargestellt werden.

3.6.1.1 Lexikon-bedingte Homographien

Wie bereits die Darstellungen in Abschnitt 2.5 auf Seite 73 zeigten, ist ein beträchtlicher Teil der Formen des FLEX-Lexikons ebenso wie die Formen des Grundformenlexikons ambig. Das schlägt sich natürlich in einer hohen Rate an Mehrdeutigkeiten beim Lemmatisieren nieder. Man muß dabei wiederum verschiedene Arten von Mehrdeutigkeiten unterscheiden:

- Wortartambiguitäten, die sowohl zufällig sein können, als auch systematisch, wie etwa die Ambiguität zwischen dem verbalen Partizip und der Prädikativform des abgeleiteten Adjektivs oder zwischen dem verbalen Infinitiv und dem nominalisierten Infinitiv. Im Bereich der Funktionswörter sind durch die funktionale Wortartklassifikation sehr viele Wortartambiguitäten anzutreffen:[1]

1. Die Kodierungen entsprechen den Tabellen in Kapitel 2.4 Das EF-FLEX-Lexikon. Die Wortarten sind mit NS und NP für Nomen, ADJ für Adjektive, Vxxx für Verben, DET für Determinatoren usw. bezeichnet. Die genauen Beschreibungen der Wortarten sind in Kapitel Abschnitt 2.2 auf Seite 34ff nachzulesen.

```
eine:
  ein.DET2:amM:nmM:aeF:neF
  einen.VSW1:1eGi:1eGc:3eGc
geäst:
  geäst.neut(NS1,NPSG):neN:deN:aeN
  äsen.VSW3:OZ
runde:
  rund.ADJ11
:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp
  rund.neut(NS1,NP2):deN:nmN:gmN:amN
  runde.fem(NS0,NP4):neF:geF:deF:aeF
  runden.VSW2:1eGi:1eGc:3eGc
verändert:
  verändert.ADJ0:up
  verändern.VSW4s:3eGi:2mGc:OZ:2mGi
sehen:
  sehen.VST1:OI:1mGi:3mGi:1mGc:3mGc
  sehen.neut(NS2,NPSG):neN:deN:aeN
doch:
  doch.PART
  doch.KONJ
  doch.ADV
```

- Homographien bei gleicher Wortart aber unterschiedlicher Grundform. Diese Mehrdeutigkeiten können durch orthographische Varianten zur Grundform, die im CISLEX separate Einträge erhalten, bedingt sein (*Typen* zu *Typ* oder *Typus* oder *Krisen* zu *Krisis/Krise*). Sie können aber auch auf tatsächlich verschiedene Lexeme zurückzuführen sein, wie die weitere mögliche Grundform *Type* im Beispiel zeigt:

  ```
  Typen:
    typ.mask(NS2,NP3):nmM:gmM:dmM:amM
    type.fem(NS0,NP4):nmF:gmF:dmF:amF
    typus.mask(NS0,NP24):nmM:gmM:dmM:amM
  ```

- Homographie der Grundformen bei unterschiedlicher morphologischer Klasse. Es kann sich in diesem Fall um rein morphologische Varianz oder Genusvarianz bei einem Lexem handeln (*der/das Teil, Wort - Worte* / *Wort - Wörter*) oder um tatsächliche Polysemie (*Bank:* Geldinstitut, Sitzgelegenheit, ...). Rein semantische Ambiguitäten können nach dem derzeitigen Stand des DKL nicht auftreten, da sie bei der morphosyntaktischen Kodierung nicht berücksichtigt werden.

```
Teil:
  teil.neut(NS1,NP2):neN:deN:aeN
  teil.mask(NS1,NP2):neM:deM:aeM
Bank:
  bank.fem(NS0,NP12):neF:geF:deF:aeF (Sitzgelegenheit)
  bank.fem(NS0,NP3):neF:geF:deF:aeF (Geldinstitut)
Gang:
  gang.fem(NS0,NP6):neF:geF:deF:aeF (Bande)
  gang.mask(NS1,NP12):neM:deM:aeM (Flur/Bewegungsart/...)
```

- Homographie von verschiedenen Flexionsformen innerhalb eines Paradigmas:

```
schöne:
  schön.ADJ0
:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp
laufen:
  laufen.VST1:OI:1mGi:3mGi:1mGc:3mGc
Fonds:
  fonds.mask(NS0,NP0):neM:geM:deM:aeM:nmM:gmM:dmM:amM
```

- Homographie durch Anwendung lexikalischer Regeln (vgl. Abschnitt 3.1.1.3.1 auf Seite 84):

```
Weise:
  weise.fem(NS0,NP4):neF:geF:deF:aeF (Art und Weise/ Lied,..)
  weise.NA:neFx:nmUx:aeFx:amUx:neUy:aeFy:aeNy:neFz:aeFz
(von ADJ: weise)
Reisen:
  reise.fem(NS0,NP4):nmF:gmF:dmF:amF
  reisen.neut(NS2,NPSG):neN:deN:aeN (vom Verb reisen)
```

  Diese Homographien sind teilweise auch dadurch bedingt, daß die nominalisierten Infinitive bereits lexikalisiert sind, so daß sie Plural erlauben und im DKL kodiert wurden:

```
Abkommen:
  abkommen.neut(NS2,NP0):neN:deN:aeN:nmN:gmN:dmN:amN
  abkommen.neut(NS2,NPSG):neN:deN:aeN
```

  In diesen Fällen sind die DKL-Einträge häufig so stark lexikalisiert, daß es sich um semantische Ambiguität handelt (*das Abkommen vom Weg - das Abkommen zur Vereinfachung der Zollformalitäten*).

- Homographie zwischen einfachen Formen und komplexen Formen (vgl. Abschnitt 2.5.4 auf Seite 78):

```
Tangente:
  tangente.fem(NS0,NP4):neF:geF:deF:aeF
  tang||ente.fem(NS0,NP4):neF:geF:deF:aeF##N,N
Bastion:
  bastion.fem(NS0,NP3):neF:geF:deF:aeF
  bast||ion.neut(NS2,NP3):neN:deN:aeN##N,N
Sozialisten:
  sozialist.mask(NS3,NP3):geM:deM:aeM:nmM:gmM:dmM:amM
  sozia||liste.fem(NS0,NP4):nmF:gmF:dmF:amF##N,N
  sozia||list.fem(NS0,NP3):nmF:gmF:dmF:amF##N,N
```

3.6.1.2 Segmentierungs- und Rekonstruktionsambiguitäten

Weitere Mehrfachanalysen ergeben sich durch verschiedene Segmentierungsmöglichkeiten bei komplexen Formen, die sich dann bei der Rekonstruktion von unter Koordination getilgten Elementen natürlich fortsetzen.

```
Aluminiumherstellung:
  aluminium||herstellung
  alu||mini||um||herstellung
  alu||mini||umher||stellung
  aluminium||her||stellung
  alu||mini||um||her||stellung

Alleinerziehende:
  allein||erziehende
  allein||erzieh||ende
  alle||in||erziehende
  alle||in||erzieh||ende
  all||einer||ziehende
  all||einer||zieh||ende
  all||ein||erziehende
  all||ein||erzieh||ende

fälschungs- und diebstahlsicher:
  fälschungs||sicher.ADJ2:up##N,ADJ
  fälschungs||stahl||sicher.ADJ2:up##N,N,ADJ
  und.KONJ
  diebstahl||sicher.ADJ2:up##N,ADJ
  dieb||stahl||sicher.ADJ2:up##N,N,ADJ
```

Als besonders problematisch bei der Segmentierung erweisen sich relativ kurze DKL-Einträge, die mit gängigen Wortendsequenzen oder auch echten Suffixen homograph sind:

```
ion (neut(NS2,NP3): elektrisch geladenes Teilchen)
  unfalls||tat||ion
  produkt||ion
  akt||ion

eren (mask(NS0,NP0): Hausflur)
  vorsichtig||eren

des (neut(NS0,NP0): Tonbezeichnung)
  verb||an||des

ger (mask(NS1,NP2): Wurfspieß)
  wieder||einst||ei||ger

usw.
```

Da die Kompositaanalyse auf einer Identifizierung der einfachen Bestandteile beruht, multiplizieren sich die Segmentierungsambiguitäten mit den lexikalischen Ambiguitäten, was zu einer Vielzahl von formal korrekten Analysen für die komplexen Formen führt.

3.6.1.3 Verschiedene Look-up-Möglichkeiten

Bei zahlreichen Formtypen sind verschiedene Look-up-Möglichkeiten gegeben, die, falls mehrere davon Erfolg haben, zu weiteren Ambiguitäten in der Analyse führen. Die WB-Formen beispielsweise können im EF-Lexikon nachgeschlagen werden, als Komposita zerlegt werden und bei den Eigennamen und Abkürzungen nachgeschlagen werden. Beispiele für Ambiguitäten durch Look-up als einfache Form und Analyse als komplexe Form wurden bereits in Abschnitt 3.6.1.1 angeführt.

```
in:
  in.PREP4
  in.EN:X
  In.AK:X
```

Ähnliches gilt auch für die MB-Formen, bei denen es sich um Akronyme und Abkürzungen handeln kann, die systematisch großgeschrieben werden, aber auch um normale Wortformen, die zur Hervorhebung in Großbuchstaben im Text auftauchen, und deshalb auch wie normale WB-Formen analysiert werden können.

Verschiedene Look-up-Möglichkeiten sind auch bei zahlreichen komplexen Formen im Bereich der Sonderformen gegeben. Eine Vielzahl dieser Formen (Zusammensetzungen mit “/”, “-”, “+”, “’”, usw.) kann sowohl als ganzes als lexikalisierte Form analysiert werden, indem die Form in speziellen Ausnahmelexika, in denen lexikalisierte Formen aufgeführt sind oder im CISLEX-AK bzw. CISLEX-EN nachgeschlagen wird, als auch komponentenweise durch Look-up der einzelnen Komponenten.

Neben diesen Fällen, bei denen es sich um zufällige Ambiguitäten handelt, tritt auch der Fall auf, daß die Mehrfachanalysen beabsichtigt sind. Formen mit interner Klammerung drücken meistens die Disjunktionen zweier Wortformen aus, ebenso wie das großgeschriebene Suffix *In* bzw. *Innen* bei Personenbezeichnungen:

```
LeserInnen:
  leser.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM
  leserin.fem(NS0,NP5):nmF:gmF:dmF:amF
```

In diesen Fällen darf natürlich keine Disambiguierung stattfinden.

3.6.1.4 Ambiguitäten bei Sonderformen

Die Ambiguitäten bei den X-Formen, den Formen also, die außer Buchstaben noch weitere Zeichen enthalten, können in zwei Arten aufgeteilt werden: einerseits Ambiguitäten, die sich aufgrund des Look-ups der enthaltenen B-Formen ergeben, und andererseits Ambiguitäten, die auf unterschiedliche Funktionen und Interpretationsmöglichkeiten der nicht-Buchstabenformen zurückzuführen sind. Außerdem gibt es, wie beim Apostroph unterschiedliche Funktionen des Zeichens innnerhalb der Form, die zu verschiedenen B-Form Look-ups führen können. Im Fall des Apostrophs ist dies einmal die Auslassung von Zeichen im Wortinnern und deren unterschiedliche Rekonstruktionsmöglichkeiten und die Möglichkeit, daß es sich um zwei zugrundeliegende Wörter handelt:

```
soll’n:
  (sollen+den).VUNR:1eGi:3eGi:eb          (2 Wörter)
  sollen.VUNR:OI:1mGi:3mGi:1mGc:3mGc      (e-Auslassung)
```

Die unterschiedlichen Funktionen der Zeichen machen sich vor allem im Bereich der ZSX-Formen bemerkbar. Der Slash kann zwischen Ziffern sowohl die normale Trennerfunktion (z.B. Schneelagebericht: *80/120*) haben, er kann als Bruchstrich auftreten und es kann sich um eine Jahreszahlfolge handeln:

```
1973/74:
  Bruchzahl(1973,74).NUM
  Jahreszahlfolge(1973,74).NUM
  Folge(Integerzahl(1973),Integerzahl(74)).NUM
```

Ähnliche Ambiguitäten können auch durch die unterschiedlichen Funktionen des Punkts (Dezimalpunkt, Gradpunkt, Ordinalzahlpunkt, Datumsangabe) auftreten. Auch der Bindestrich kann sowohl als Minuszeichen, als Bindestrich, zur Angabe einer Differenz und bei Tilgung unter Koordination auftreten.

Auf ähnliche Ambiguitäten wie bei der Kompositazerlegung stößt man bei den BSX-Formen, die als Zeichen einen Punkt am Ende enthalten. Es kann sich sowohl um eine “einfache” Abkürzung handeln als auch um eine komplexe Form, deren Hinterglied

eine Abkürzung ist. Im letzten Fall gibt es häufig verschiedene Segmentierungsmöglichkeiten.

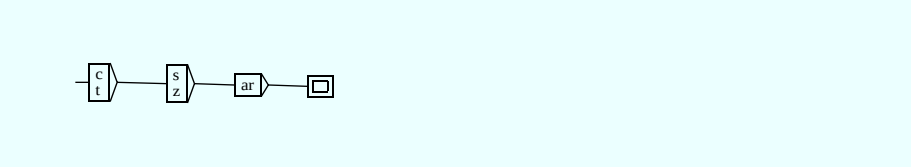
```
Fachbegr.:
  fach||beg||r..AK:X##N,N,AK
  fach||be||gr.AK:X##N,PREF,AK
  fach||Begr..AK:X##N,AK
```

3.6.2 Disambiguierungsmöglichkeiten

In diesem Abschnitt wird es darum gehen, für die oben beschriebenen Arten von Ambiguitäten zu entscheiden, ob eine Disambiguierung überhaupt sinnvoll und mit den zur Verfügung stehenden Mitteln möglich ist. Wie in den vorausgehenden Abschnitten werden wieder zuerst die lexikonbedingten Ambiguitäten betrachtet.

Homographie von Flexionsformen innerhalb eines Paradigmas kann von vornherein von der Disambiguierung ausgeschlossen werden. Theoretisch wäre es zwar möglich durch eine Kontextanalyse unter Berücksichtigung der Kongruenz die eine oder andere Merkmalskombination auszuschließen, dabei steht jedoch der Nutzen in keinem Verhältnis zum Aufwand. Für die praktischen Anwendungen ist eine Disambiguierung in diesem Bereich kaum von Interesse. Aus Gründen mangelnder Information im Lexikon sind des weiteren Auflösungen von semantischen Ambiguitäten bei gleicher Grundform und gleicher morphologischer Klasse nicht möglich. Ein Ausschluß bestimmter Alternativen ist lediglich im Rahmen der Disambiguierung von Homographen mit gleicher Wortart aber unterschiedlicher morphologischer Klasse möglich, wenn sich die Kongruenzmerkmale unterscheiden. Ansonsten ist eine semantische Disambiguierung erst möglich, wenn umfangreiche Teile des DKL auch semantisch kodiert sind. Auch die Auflösung von Mehrfachlemmatisierungen, die auf orthographische Varianten zur Grundform zurückzuführen sind, ist (wenn nicht durch andere Strategien) aufgrund fehlender Information im Lexikon nicht möglich. Da jedoch das Erkennen von orthographischen Varianten für viele Anwendungen des Lexikons eine Rolle spielt, muß hier das Lexikon dringend erweitert werden. Mögliche Lösungen für die Repräsentation orthographischer Varianten sind entweder Ergänzungslisten zum DKL, die zu jeder orthographischen Grundformvariante die kanonische Grundform enthalten oder aber in Anlehnung an das französische DELAS, eine Kodierung der orthographischen Grundformvarianten durch endliche Automaten, wie dies im folgenden Beispiel für die französischen Schreibvarianten von *czar*: *tzar, csar, tsar* (auf Deutsch *Zar*) dargestellt ist.

Abbildung 5. Automat zur Darstellung der Schreibvarianten von czar



Silberztein (1989), S. 28

Durch eine systematische Erfassung der Grundform- und morphologischen Varianten im Rahmen solcher Transduktoren könnten Mehrfachanalysen, die nur auf Grundformvarianten, ebenso wie die Fälle, die nur auf Varianz bei der morphologischen Klasse beruhen, auf eine kanonische Grundform reduziert werden. Ambiguitäten bezüglich der morphologischen Klasse (bei gleicher Wortart) lassen sich im allgemeinen Fall nur dann auflösen, wenn der Kontext so beschaffen ist, daß aufgrund von Kongruenztests eine Lemmatisierungsmöglichkeit tatsächlich ausgeschlossen werden kann (vgl. 3.6.4.2 Kongruenzketten).

Bei Wortartambiguitäten kann mittels der Groß- bzw. Kleinschreibung des Wortes schon eine gewisse Vorauswahl getroffen werden (vgl. 3.6.3.1 Der Groß/Klein-Filter). Eine weitere Einschränkung ist nur unter Heranziehung des Kontexts möglich (vgl. 3.6.4.1 Kontexte zur Wortartdisambiguierung).

Bei den Sonderformen gilt für die Ambiguitäten, die durch den Look-up von B-Form-Komponenten bedingt sind sowie für die Segmentierungsambiguitäten bei Punktformen dasselbe wie für die B-Formen. Bei den durch Funktionsambiguitäten der Sonderzeichen bedingten Mehrfachanalysen erfordert eine Disambiguierung jedoch eine detailliertere Analyse des Kontexts. In diesem Bereich wird jedoch auf eine weitere Disambiguierung verzichtet, da diese Sonderformen (in der Regel ZSX-Formen) als Bestandteile größerer Einheiten betrachtet werden müssen. Beispielsweise sollten alle Datumsangaben, unabhängig, ob sie nur eine A-Form enthalten (*11.11.1999*, *29.2*) oder ob sie aus mehreren Formen bestehen (*1. April*, *24. 12. 1980*, *erster Januar*, usw.) einheitlich im Text annotiert werden. Eine solche inhaltliche Lemmatisierung, die größere Einheiten als einzelne Formen behandelt, ist jedoch nicht Gegenstand dieser Arbeit.

3.6.3 Disambiguierung aufgrund der Orthographie

Einen guten Anhaltspunkt zur Einschränkung der Menge der möglichen Analysen bietet die Groß-/Kleinschreibung. Bei den WB-Formen ist die initiale Groß- bzw. Kleinschreibung ein wichtiges Kriterium zum Ausschluß bestimmter Wortarten. Bei den anderen B-Formen ist die gesamte Erscheinungsform relevant: je größer die Übereinstimmung der im Text gefundenen Form mit der Lemmaform, desto sicherer kann eine Analyse als richtig angenommen werden.

3.6.3.1 Der Groß/Klein-Filter

Die WB-Formen erhalten ein Merkmal, ob sie groß- oder kleingeschrieben im Text erscheinen. Für großgeschriebene Wörter am Satzanfang oder nach Satzzeichen (zum Beispiel ":") wird ein spezielles Merkmal angenommen, da in diesem Fall die Großschreibung durch den Kontext bedingt sein kann und nicht als Grundlage für eine Disambiguierung verwendet werden kann. Entsprechend werden auch Sonderformen, die an einer relevanten Stelle eine WB-Form enthalten, markiert. Bei der Vielfalt der Formen ist im Einzelfall jedoch sehr genau zu prüfen, wann das Merkmal für Groß- bzw. Kleinschreibung tatsächlich gesetzt wird. Bei Bindestrichwörtern oder Slash-Konstruktionen ist nicht die initiale Groß- oder Kleinschreibung relevant, sondern die des letzten Elements. Dagegen ist bei Konstruktionen vom Typ *Zahl + Zahlsuffix* ebenso wie im Fall eines Apostrophs am Wortanfang überhaupt keine Aussage möglich.

Alle möglichen Tags, die aufgrund des Look-ups im EF-Lexikon oder aufgrund einer Kompositaanalyse zustanden gekommen sind, werden mittels des Groß/Klein-Filters überprüft. Für die kleingeschriebenen Wörter sind die Analysen als Nomen bzw. als nominalisiertes Adjektiv oder nominalisierter Infinitiv auszuschließen. Es gibt lediglich einige Sonderfälle, die separat geprüft werden: einige der quantifizierenden Nomen (*bißchen, paar,* usw.) sowie die Tageszeiten können kleingeschrieben auftreten. Diese Fälle sind in einer Ausnahmeliste gesammelt und die entsprechenden Tags werden bei Kleinschreibung nicht eliminiert. Großgeschriebene Formen sind in der Regel Nomen oder nominalisierte Adjektive. Eine Ausnahme bilden die Anreden, d.h. die Pronomen in der zweiten Person. Diese Analysen müssen bei großgeschriebenen Wörtern natürlich zugelassen werden.

Der Groß/Klein-Filter findet weitere Anwendung bei EN-Tags. Wird eine WB-Form (ob es sich um eine isolierte WB-Form oder um die WB-Komponente einer Sonderform handelt, bei der die WB-Form für die Lemmakategorie entscheidend ist, spielt hierbei keine Rolle) als Eigenname analysiert, so ist diese Analyse für nicht weiterspezifizierte Eigennamen nur dann akzeptabel, wenn die Form das Merkmal für Großschreibung hat. Ausgeschlossen von dieser Regel sind nicht-nominale Ableitungen von Eigennamen.

3.6.3.2 Der Ähnlichkeitsfilter

Der Ähnlichkeitsfilter beruht auf dem Prinzip, daß bei Mehrfachanalysen diejenige ausgewählt werden sollte, deren Schreibweise am stärksten mit der im Lexikon verzeichneten Form übereinstimmt. Dieser Filter kommt vor allem bei Eigennamen und Abkürzungen zum Einsatz, da in diesem Fall erstens die Groß- oder Kleinschreibung bestimmter Buchstaben signifikant ist und zweitens durch die Regel der s-Anfügung zur Pural- oder Genitivmarkierung Formen erzeugt werden, die nicht im Lexikon stehen. Wie bereits in Abschnitt 3.3.1.5 auf Seite 105 beim Look-up im CISLEX-AK beschrieben wurde, werden Abkürzungen in einer normalisierten Form, d.h. alle Buchstaben klein und ohne Punkte, nachgeschlagen, da es in diesem Bereich große Abweichungen von der konventionalisierten Form gibt. Dort wurde als Lösungsmög-

lichkeit auch schon angedeutet, daß diese Varianten durch das AK-Lexikon abgedeckt werden sollten. Solange dies nicht der Fall ist, können bestimmte Analysen nur aufgrund eines Vergleichs der Originalschreibweise im Text mit der konventionalisierten Schreibweise im Abkürzungslexikon ausgeschlossen oder ausgewählt werden. Tritt beispielsweise die Form *IM* im Text auf, so ist die Lemmatisierung als Abkürzung für inoffizieller Mitarbeiter (*IM.AK:X*) der Lemmatisierung als Präposition mit Determinator (*in.PDET:deM:deN*) vorzuziehen. Im Falle einer Textform *im* wäre entsprechend die PDET-Lemmatisierung vorzuziehen.

Bei den Abkürzungen und Eigennamen sind aufgrund der Ähnlichkeitsbedingung auch solche Analysen vorzuziehen, die direkt aus dem Lexikon abgeleitet werden können, gegenüber solchen, die durch eine Regel abgeleitet werden.

3.6.4 Disambiguierung aufgrund des unmittelbaren Kontexts

Weitere Einschränkungen der Menge der möglichen Lemmatisierungen sind nur durch Betrachtung des Kontexts möglich.[1] Der Kontext ist in diesem Fall auf die benachbarten Formen beschränkt.

3.6.4.1 Kontexte zur Wortartdisambiguierung

Wortartambige Formen können häufig aufgrund des (Wortart-)Kontexts disambiguiert werden. Eine Analyse des Wortartkontexts kann zum Ausschluß einer Analyse aufgrund von ungrammatikalischen Abfolgen führen, sie kann jedoch in anderen Fällen, wenn die Kriterien nicht eindeutig sind, auch zu einer Präferenz einer bestimmten Analyse führen. Ein ganz typisches Ausschlußkriterium ist beispielsweise die Regel, daß Partikeln nicht das Vorfeld eines Verbzeitsatzes bilden können, von der folgender Ausschlußkontext abgeleitet wurde (SA steht für Satzanfang, SE für Satzende):

```
* SA PART Vfin
```

Neben solchen negativen Kontextbeschreibungen, die zum Ausschluß bestimmter Analysen führen, gibt es auch Ausschlußkriterien, die von mehreren Alternativen nur eine zulassen:

```
SA {DET,PRO} ADJ* N -> SA DET ADJ* N
```

Diese Auswahlkontextregeln können nur auf Alternativenmengen angewendet werden. Die Präferenzregeln haben dasselbe Format, unterscheiden sich jedoch im Status. Für die Regeln gilt die Konvention, daß *N* sowohl normale Nomen, deadjektivische Nomen als auch Eigennamen bezeichnet, es sei denn *EN* oder *NA* taucht explizit in der Regel als Alternative auf.

1. Statistische Methoden auf Wortebene, wie Auftretenswahrscheinlichkeiten bestimmter Wortarten oder bestimmter Wortformen werden hier nicht zur Disambiguierung herangezogen.

Zur Disambiguierung verwendete Ausschlußregeln:

```
* SA PART Vfin

* SA Vinf

* SA {PREP,DET} Vfin
```

Zur Disambiguierung verwendete “sichere” Auswahlregeln[1]:

```
DET {V|ADJ} N -> DET ADJ N

DET {ADV|ADJ} N -> DET ADJ N

SA {DET,PRO} ADJ* N -> SA DET ADJ* N

SA {ADJ,NA} N -> SA ADJ N

SA {ADJ,N} Vfin -> SA N Vfin

SA {ADJ,N,EN} Vfin -> SA EN Vfin

{PREP,VPART} SE -> VPART SE
```

Zur Disambiguierung verwendete Präferenzregeln:

```
{DET,V} N -> DET N

SA {EN,N} -> SA EN
```

Die Treffsicherheit der Präferenzregeln kann dadurch erhöht werden, daß wenn möglich zusätzlich Kongruenz überprüft wird (s.u.). So kann die Präferenz des Determinators gegenüber dem Verb-Lemma vor einem Nomen zusätzlich motiviert oder aber auch (wenn keine Kongruenz möglich ist) verworfen werden.

3.6.4.2 Kongruenzketten

Für Ambiguitäten innerhalb einer Wortart (in wenigen Fällen auch bei Wortartambiguitäten, s.u.) kann eine Disambiguierung aufgrund des morphologischen Kontexts erfolgen, wenn sich die ambige Form in einem Kontext befindet, der eine Kongruenzkette bildet. Kongruenzketten gibt es zwischen Subjekt und finitem Verb bezüglich Person und Numerus, zwischen Präposition und NP bezüglich Kasus[2] und in der NP bezüglich Kasus, Genus und Numerus. Die Kongruenz zwischen Subjekt und finitem Verb kann jedoch momentan zur Disambiguierung nicht herangezogen werden, da die eindeutige Identifikation des Subjekts in einem Satz mit den zur Verfügung stehenden

1. Als “sicher” gelten Regeln, wenn durch die Anwendung der Regel keine korrekten Analysen ausgeschlossen werden. Im Gegensatz dazu sind Präferenzregeln “unsicher”, da hierbei unter Umständen auch korrekte Analysen verworfen werden.

2. In diesem Fall handelt es sich strenggenommen nicht um Kongruenz sondern um Rektion.

Informationen nicht möglich ist. Auch die Kasusrektion in der Präpositionalphrase ist nur am Rande von Interesse, da sie in der Regel auch keine weiteren Anhaltspunkte zur Disambiguierung liefert als die Kongruenz innerhalb der Nominalphrase. Die Idee ist die, daß in Folgen der Form (DET) - ADJ* - N mindestens eine Lemmatisierung existieren muß, so daß die Merkmale für Kasus, Genus und Numerus bei allen beteiligten Elementen übereinstimmen. Das Beispiel *einige Blocks* soll dieses Prinzip veranschaulichen:

```
einige Blocks:

einige:
  einig.ADJ0
:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp
  einig.DET1:nmM:amM:aeF:deF

Blocks:
  block.mask(NS1,NP6):geM:nmM:gmM:dmM:amM
  block.mask(NS1,NP12):geM
```

Für diese Phrase gibt es theoretisch 4 Analysen. Lediglich zwei Analysen enthalten jedoch eine Kongruenzkette, nämlich:

```
  einig.DET1:nmM:amM:aeF:deF
  block.mask(NS1,NP6):geM:nmM:gmM:dmM:amM

einig.ADJ0
:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp[1]
block.mask(NS1,NP6):geM:nmM:gmM:dmM:amM
```

So konnte das Nomen *Blocks* disambiguiert werden. Eine Disambiguierung von *einig.DET* und *einig.ADJ* wäre, wenn überhaupt, nur im Rahmen einer vollständigen syntaktischen und semantischen Satzanalyse möglich. Wie das folgende Beispiel zeigt, kann oftmals auch dann die Ambiguität nicht aufgelöst werden:

Einige Blocks sind relativ selten.

In beschränktem Umfang kann dieses Verfahren auch zur Wortartdisambiguierung herangezogen werden, beispielsweise bei Ambiguitäten vom Typ DET/ADJ oder ADJ/N.

3.6.5 Heuristiken

Bei den im folgenden beschriebenen Disambiguierungsregeln handelt es sich durchweg um reine Heuristiken, die mangels klarer Kriterien für eine Disambiguierung hier eingesetzt werden, auch auf das Risiko hin, daß in seltenen Fällen korrekte Analysen

1. An dieser Stelle sei nochmals daran erinnert, daß *U* das Metamerkmal für keine Genuseinschränkung ist, und somit mit beliebigen Genera eine Kongruenzkette bilden kann.

ausgeschlossen werden. Diese Heuristiken sind gößtenteils motiviert durch linguistische Regularitäten und Korpusuntersuchungen.

3.6.5.1 Präferenz für lexikalisierte Formen

Wie bereits in Abschnitt 3.6.1.4 auf Seite 136 beschrieben, ist bei vielen Sonderformen die Möglichkeit der kompositionellen Analyse neben dem Look-up in einer Spezialliste eine Quelle für Ambiguitäten. In diesen Fällen wird, wenn eine Disambiguierung gewünscht wird, prinzipiell die Lemmatisierung als lexikalisierte Form präferiert.[1] Nach demselben Prinzip wird bei WB-Formen die Lemmatisierung auf eine einfache Form des DKL gegenüber einer Analyse als komplexe Form bevorzugt. Auch gegenüber einer Analyse als Abkürzung oder Eigenname wird die EF-Analyse präferiert. Anders dagegen bei den GB- und MB-Formen: hier wird als wahrscheinlichste Kategorie Abkürzung, gefolgt von Eigenname gegenüber den anderen Klassen ausgewählt.

3.6.5.2 Bewertung der Segmentierungen bei der Kompositaanalyse

Bei Segmentierungsambiguitäten werden auf die Hinterglieder zuerst die oben beschriebenen Filter angewendet. Kann damit noch keine vollständige Disambiguierung erreicht werden, so wird die Segmentierung selbst näher betrachtet. Die relevanten Parameter sind die Anzahl der Komponenten, die Kategorien der einzelnen Komponenten und die Kategorienabfolge. Dazu kommt noch eine lexikalische Untersuchung des Hinterglieds.

Nach einer Untersuchung von Schoenherr (1991) sind 77,3 % (bezogen auf die Lemmata) der Komposita zweigliedrig, 20,7 % dreigliedrig, 1,9% viergliedrig und nur 0,06 % sind fünfgliedrig. Nach diesen Ergebnissen sind die Segmentierungen mit der geringsten Anzahl von Segmenten gegenüber den Segmentierungen mit mehr Segmenten deutlich wahrscheinlicher. Der erste Schritt bei der Disambiguierung der Segmentierungen besteht demzufolge in der Auswahl der Segmentierungen mit der geringsten Anzahl an Komponenten. Mit Hilfe dieser Heuristik können bereits einige Zerlegungsambiguitäten beseitigt werden (die mit * markierten Analysen werden ausgeschlossen):

```
Volumenseinbuße:
  volumens||einbuße.fem(NS0,NP4):nmF:gmF:dmF:amF##N,N
  *volumen||sein||buße.fem(NS0,NP4):nmF:gmF:dmF:amF##N,N+V,N
  *volumens||ein||buße.fem(NS0,NP4):nmF:gmF:dmF:amF##N,V,N

Voreiligen:
  vor||eilige.NA:geMx:...##PREP,NA
  *vor||ei||liga.fem(NS0,NP18):nmF:gmF:dmF:amF##PREP,N,N
```

1. Tatsächlich wird die Wortformanalyse bereits nach dem Auffinden der Form in einer Spezialliste abgebrochen, wenn die Disambiguierungsoption gewählt wurde.

```
Zweitwichtigster:
  zweit||wichtige.NA:neMz:neMy:geFx:deFx:dmUx##ADJ,NA
  *zweit||wichtig||ster.mask(NS2,NP2):neM:deM:aeM##ADJ,ADJ,N

Sachverständigenkommission:
  sach||verständigen||kommission.fem(NS0,NP3)
:neF:geF:deF:aeF##N,V+N+NA,N
  *sach||verständigen||kommiss||ion.neut(NS2,NP3)
:neN:deN:aeN##N,V+N+NA,N,N
```

Wie die beiden letzten Beispiele zeigen, können durch diese Heuristik in vielen Fällen die unerwünschten Segmentierungen, die durch Homographie von kurzen EF-Formen mit gängigen Wortendsequenzen (z.B. *Ster, Ion, Des,* usw.) entstehen, beseitigt werden.

Aus derselben Studie (Schoenherr 1991) geht hervor, daß 84,2 % der in den untersuchten Komposita vorkommenden Vorderglieder (bezogen auf die Tokens) Nomen sind. Aus diesem Grund werden Segmentierungen mit möglichst vielen N-Komponenten bevorzugt. Bezüglich der kategorialen Abfolge können noch einige weitere Einschränkungen gemacht werden, wenngleich es bei der Komposition nur wenig Beschränkungen gibt. So kann ein Präfix nicht direkt von einem Suffix gefolgt sein, Suffixfolgen sind ausgeschlossen.[1] Auch Funktionswörter mit Ausnahme von Adverben sind gegebenenfalls auszuschließen. Ebenfalls auszuschließen sind V-V-Kombinationen. Zwei Verbstämme hintereinander sind nur bei Partizipien möglich. Die allerdings sind als Adjektive klassifiziert und fallen nicht unter diese Beschränkung: *gehbehindert* aber nicht *gehbehindern*. Durch diese Heuristiken läßt sich zwar die Menge der möglichen Segmentierungen schon relativ gut einschränken, in der Praxis bleiben jedoch noch eine Menge von Zerlegungsambiguitäten, die häufig daher rühren, daß ein *s* in der Wortmitte zum einen als Fugen-s und zum anderen als erster Buchstabe des folgenden Elements interpretiert werden kann:

```
Rechnungsprüfungsamt:
  rechnungs||prüfung||samt.mask(NS1,NP2):neM:deM:aeM##N,N,N
  rechnungs||prüfungs||amt.neut(NS1,NP14):neN:deN:aeN##N,N,N

Reichstag:
  reich||stag.neut(NS1,NP3):neN:deN:aeN##ADJ+N+V,N
  reichs||tag.mask(NS1,NP2):neM:deM:aeM##N,N
```

Dieser Fall kommt relativ häufig vor und betrifft Paare von FLEX-Formen der Form <sx, x> bzw. <x, sx>, wobei x und sx Formen derselben Wortklasse sind.[2] In den obigen Beispielen ist eine Lesart klar präferiert. Das ist jedoch nicht notwendig so:

1. Die im CISLEX verzeichneten Suffixe haben annähernd Wortcharakter (in der Literatur auch als Suffixoide oder Konfixe bezeichnet) und wurden nur aufgrund ihres selbständigen Charakters und der hohen Produktivität, die der Kompositabildung entspricht, ins Lexikon aufgenommen. Diese Suffixe sind nicht untereinander kombinierbar.

```
Wachstube:
  wach||stube.fem(NS0,NP4):neF:geF:deF:aeF##ADJ+N+V,N
  wachs||tube.fem(NS0,NP4):neF:geF:deF:aeF##N+V,N
```

In diesem Fall sollte keine der beiden Analysen präferiert werden. Aus diesem Grund wurden die in Frage kommenden Paare daraufhin untersucht, ob eines der Elemente in Komposita gegenüber dem anderen präferiert werden sollte. Paare, bei denen es für ein Element eine klare Präferenz gibt, sind in einer Liste gesammelt. Diese Liste wird bei der Disambiguierung überprüft und falls beide Formen als Kompositaglieder auftauchen, das Erstgenannte ausgewählt. Eine Ambiguität ähnlichen Typs, die ebenfalls sehr häufig auftritt, ist folgende:

```
schwindelerregende:
  schwindel||erregend.ADJ0:neFxp:...:aeFzp##N+V,ADJ
  schwindel||erreg||enden.VSW2:1eGi:1eGc:3eGc##N+V,V,V

Zugreisende:
  zug||reisende.NA:neFx:...:aeFz##N,NA
  zug||reis||ende.neut(NS0,NP4):neN:geN:deN:aeN##N,N+V,N
```

In diesen Fällen ist die Zerlegung, deren Hinterglied ein adjektivisches Partizip bzw. dessen Nominalisierung ist, gegenüber dem Verb *enden* bzw. dem Nomen *Ende* vorzuziehen.

Für die Zukunft wird das Kompositalexikon (DKL-KF, vgl. Abschnitt 2 auf Seite 26) eine wichtige Entscheidungsgrundlage zur Auswahl bestimmter Segmentierungen darstellen. In diesem Lexikon werden alle lexikalisierten und häufig gebrauchten Komposita aufgeführt. Bei der Analyse von Komposita, die nicht im Kompositalexikon enthalten sind, können für die Segmentierung sowohl das Lexikon der einfachen Formen als auch das der komplexen Formen verwendet werden. Durch die Präferenz von Segmentierungen mit möglichst wenig Elementen ist dann gewährleistet, daß lexikalisierte Komposita als Konstituenten komplexerer Komposita korrekt identifiziert werden.

3.6.5.3 Typische Kontexte für bestimmte Kategorien

Die in Abschnitt 3.6.4 auf Seite 140 vorgestellten Kontextregeln erlauben eine Disambiguierung nur in den dort aufgeführten Spezialfällen. In anderen Kontexten ist eine Disambiguierung momentan nicht möglich. Hier ist eine Erweiterung der Heuristiken mit Berücksichtigung des Kontexts dringend erforderlich. Im folgenden soll kurz skizziert werden, wie solche Heuristiken formuliert werden können.

2. Theoretisch können auch Zerlegungsambiguitäten bei Formen verschiedener Wortklassen entstehen, die Praxis hat jedoch gezeigt, daß die Auswahlmöglichkeiten durch die vorangehenden Heuristiken bereits soweit eingeschränkt sind, daß es es genügt, Paare gleicher Wortart zu behandeln.

Das Verfahren der Kontextregeln ist für Heuristiken wie bereits erwähnt deshalb ungünstig, da es nur auf ganz spezielle Kontexte anwendbar ist. In Kontexten, die durch diese Regeln nicht erfaßt werden, ist jedoch keine Einschränkung der Analysenmenge möglich. Bei Heuristiken sollte sinvollerweise umgekehrt vorgegangen werden, das heißt, es sollten für Kategorien, die häufig in Mehrfachanalysen auftreten, typische Kontexte definiert werden. Typische Kontexte für Eigennamen sind beispielsweise:

- nach Titeln
- vor oder nach anderen Eigennamen
- nach Initialen
- allein im Vorfeld

Für Adjektive typische Kontexte wären etwa:

- zwischen weiteren Adjektiven
- zwischen Determinator und Adjektiv
- nach Kopulaverben
- zwischen Determinator oder Adjektiv und Nomen
- nach Gradpartikeln wie z. B. *sehr*

Für die Definition solcher Kontexte können einerseits syntaktische Abfolge- und Kombinationsregeln ausgewertet werden, zum anderen können die Kontexte nicht ambiger Formen in bereits lemmatisierten Korpora analysiert werden. Für die Beschreibung solcher Kontexte bieten sich die bei den lokalen Grammatiken verwendeten endlichen Automaten an (siehe dazu die Ausführungen in Abschnitt 3.5.2 auf Seite 126ff).

3.7 Nicht-erkannte und fehlerhafte Formen

Trotz eines umfangreichen Lexikons und diverser Analysemöglichkeiten enthält jeder Text eine Reihe von nicht-erkannten Formen. Bei den Texten des Testkorpus betrug der Anteil der nicht-erkannten Formen durchschnittlich 2 % bezogen auf die Tokens. Als Gründe für die fehlende Lemmatisierung kommen sowohl fehlerhafte Formen oder fehlende Information im Lexikon in Frage. Insbesondere im Bereich der Eigennamen und Abkürzungen kann das Lexikon nicht vollständig sein (eine Aufschlüsselung der nicht erkannten Formen befindet sich in Tabelle 15, “Unbekannte Formen bezogen auf eine Ausgabe”, auf Seite 154. Bei den fehlerhaften Formen handelt es sich um Tippfehler in Wortformen[1] (falsche Buchstaben, vertauschte Buchstaben oder fehlende Buchstaben): *zugunsetn, Rindfleich, imerhin,* usw., fehlende Blanks, die schon bei der Tokenisierung zu falschen Ausgangsanalyseformen führen: *zuKreuzfahrten, 281Seiten, “,Ersatz”, 65und70Prozent,* usw. oder aber völlig fragwürdige Kombinationen wie *0o1ZDF.* Für manche der Fälle erlauben die Analyseregeln eine fehlertolerante Verarbeitung, wie etwa bei fehlenden Blanks zwischen Wortformen, wenn das zweite Wort großgeschrieben ist oder bei fehlenden Blanks in Zahlkonstruktionen. Andere Fälle, die sich nicht durch bestimmte Muster auszeichnen, können jedoch durch solche Strategien nicht abgefangen werden. In diesen Fällen ist es allerdings auch nicht die Aufgabe der Lemmatisierung, die entsprechende Fehlerkorrektur vorzunehmen.

3.7.1 Nicht-erkannte Formen

Ganz allgemein stellt sich die Frage, ob ein Lemmatisierungssystem bei nicht erkannten Wörtern versuchen sollte, eine hypothetische Lemmaform und Lemmaklasse anzunehmen oder nicht. Grundsätzlich müssen hypothetische Lemmata als solche markiert werden. Es gibt sicherlich keine für alle Anwendungen allgemeingültige Strategie. Im Einzelfall, wenn es um die möglichst genaue Analyse relativ kleiner Textmengen geht, kann auch ein interaktives Lemmatisieren, bei dem bei unbekannten Formen Vorschläge zur Lemmatisierung gemacht werden, unter denen der Anwender auswählen kann, sinnvoll sein. Dagegen ist ein solches Vorgehen bei großen Textmengen nicht praktikabel. Ein Lemmatisierungsalgorithmus sollte aus diesem Grund alle drei Optionen vorsehen. Das bedeutet, daß auf jeden Fall Heuristiken zur Klassifikation unbekannter Formen entwickelt werden müssen. Im folgenden werden zuerst die nicht-erkannten Formen auf ihre Analysemöglichkeiten hin untersucht. Danach werden einige Tests, die eine weitgehend fundierte Analyse nicht-erkannter Formen erlauben, vorgestellt, und schließlich werden einige Heuristiken für Fälle, denen auch bis dahin kein Lemma zugeordnet werden konnte, angegeben.

1. Ein Teil der fehlerhaften Formen kommt auch durch Konvertierungsprobleme zustande, da bei Übertragung Accents zum Teil verlorengehen, und so eine Orthographie entsteht, die nicht im Lexikon verzeichnet ist: *Renommee.*

3.7.1.1 Nichterkannte Formen im Text

Bei den aufgrund mangelnder Information im Lexikon nicht erkannten Formen handelt es sich in der Mehrzahl um Eigennamen. Obwohl es das Ziel ist, soviel Eigennamen wie möglich über das EN-Lexikon zu analysieren, kann in diesem Bereich nie Vollständigkeit erreicht werden. Es ist deshalb notwendig, Strategien zur Identifizierung im Lexikon unbekannter Eigennamen zu finden. Weitere Quellen für unbekannte Wörter sind:

- Der Fremdsprachenbereich: *correctness, don't, dreaming, electronic, indoculture, nuovo, outdoor,* usw. In diesen Fällen handelt es sich teilweise um fremdsprachliche Wörter, die oft schon im Übergang zum normalen deutschen Wortschatz sind, wie dies bei *indoor* oder *outdoor* der Fall ist. Solche Formen sollten auf jeden Fall im Lexikon verzeichnet werden. Es empfiehlt sich hier ein Übergangslexikon, aus dem die Formen nach hinreichender Auftretenshäufigkeit ins DKL übernommen werden.
  Weitere Vorkommen fremdsprachlicher Wörter sind im Rahmen fester Wendungen oder Termini, die im Deutschen bereits gebräuchlich sind, ohne daß es jedoch Sinn machen würde die einzelnen Formen als Fremdwörter aufzunehmen, wie *correctness* in *political correctness, facto* in *de facto,* usw., aber auch in mehrteiligen Eigennamen wie *British Airways* oder in Zitatkontexten, etwa bei Titeln von Musikstücken oder Filmen usw. Die Zitatkontexte sind aufgrund von textuellen Merkmalen und aufgrund der fehlenden Lemmainformation aufeinanderfolgender Formen von einer Verarbeitung auszuschließen, bzw. als Zitatkontext zu markieren. Die Mehrwortlexeme und komplexen Termini müssen in einer ohnehin auch fürs Deutsche notwendigen Komponente zur Analyse von Mehrwortlexemen bearbeitet werden. Dieses Thema ist jedoch nicht Gegenstand dieser Arbeit. Ähnliches gilt für die mehrteiligen Eigennamen.
- Veraltete Formen und mundartliche Formen: *produciren, bekömmt, danäxde, dreiazwanzgfuchzge, Hefeküchli, Häuslebauer, neig'schrieb'n, spuit's,* usw. Die veralteten Formen kommen in der Regel in Zitatkontexten vor. Bei den mundartlichen Formen ist auch das Vorkommen außerhalb spezieller Zitatkontexte häufig. Diese Formen können aufgrund der Lexikoninformation nicht lemmatisiert werden, und es kann auch nicht Ziel des Lexikons sein, derart abweichende Formen zu erkennen. Zu einer weiteren Textbearbeitung können jedoch (unter anderem auch aufgrund der fehlenden oder lückenhaften Lemmatisierung ganzer Textabschnitte) solche typischen Zitatkontexte abgegrenzt werden und von einer weiteren Bearbeitung ausgenommen werden. Nicht in Zitatkontexten auftretende Formen bleiben unerkannt. Für eine umfangreiche Bearbeitung von Texten, die einer Region zuzuordnen sind (wie das bei Zeitungstexten ja der Fall ist), ist es sicher sinnvoll ein

spezielles Lexikon der gebräuchlichsten mundartlichen Formen aufzubauen.[1] Im Testkorpus häufig auftretende Formen waren beispielsweise *radeln, Wies'n, Brez'n,* usw.

- Fachbegriffe: Obwohl das DKL eigentlich alle Wörter, die in "normalen" Texten vorkommen[2], enthalten sollte, tauchen auch in Zeitungstexten immer wieder spezielle Fachbegriffe auf, die nicht erkannt werden (z.B. *hämostaseologische*). Falls es sich in diesen Fällen um einfache Formen im Sinne der CISLEX-Definition handelt, bleiben diese Formen unanalysiert. Eine Lösungsmöglichkeit für die Zukunft wäre aber etwa eine Suffixanalyse anhand typischer Suffixe, die in Fachbegriffen auftreten, wie im obigen Fall etwa *-logisch*. Bei komplexen Formen, die als Hinterglied eine im CISLEX bekannte Form haben, wird diese mit unbekanntem Vorderglied ausgegeben.
- Interjektionen: Bei den Interjektionen gibt es neben den konventionalsierten Formen, die auch im Lexikon enthalten sind, eine Vielzahl spontaner Neubildungen, die häufig auch lautmalenden Charakter haben *(huhu, hmpf,...*). Diese Formen können zum Teil mit Hilfe von Heuristiken (siehe unten) unter Beachtung der Form als Interjektionen erkannt werden. Ansonsten bietet nur eine Betrachtung des Kontexts Anhaltspunkte. Interjektionen kommen typischerweise in 1-Wortsätzen, am Satzanfang und vor Ausrufezeichen vor.

3.7.1.2 Produktive Derivationen

Das DKL hat den Anspruch, den Bestand an einfachen Formen zusammen mit der Derivation (EF+) beziehungsweise nur Suffigierung (EF) abzudecken. Bei hochproduktiven Derivationsmustern, wie beispielsweise die nominale Ableitung auf *-ler* (*Produktler, Westler, ...*), kann es jedoch vorkommen, daß diese Formen nicht im Lexikon erfaßt sind. Für solche produktiven Fälle wurde eine einfache Derivationskomponente erstellt, die Formen mit entsprechenden Endungen auf Derivation überprüft. Es handelt sich hierbei nicht nur um eine Endungsanalyse, denn nur wenn die Derivationsbasis anhand des DKL identifiziert werden konnte, wird die entsprechende Analyse als Lemma gewählt.

3.7.1.3 Komplexe Formen mit unbekannten Teilen

Das Ziel bei der Analyse der Formen ist es, auch bei komplexen Formen jeden Bestandteil zu analysieren. Eine strikte Befolgung dieses Grundsatzes hätte allerdings

1. Ein speziell auf bestimmte Textsammlungen zugeschnittenes Zusatzlexikon empfiehlt sich auch im Bereich der Eigennamen, in dem dann die Namen lokaler Größen, bekannter Straßen und Plätze sowie geographische Bezeichnungen der Region aufgeführt werden, die den Rahmen des CISLEX-EN sprengen würden.

2. Für die Bearbeitung von Fachtexten sind natürlich entsprechende Fachlexika notwendig, da für Anwendungen, wie beispielsweise Information Retrieval oder automatische Indizierung gerade die Fachtermini von besonderer Relevanz sind.

eine wesentlich höhere Rate von nichterkannten Formen zur Folge. Es würde auch wenig Sinn machen beispielsweise Bindestrichwörter, bei denen eine Komponente unbekannt ist, vollständig als unbekannt zu analysieren. Können komplexe Formen (Sonderformen oder auch komplexe WB-Formen) nicht vollständig identifiziert werden, so wird bereits bei der Formanalyse der Versuch unternommen, soviel Bestandteile wie möglich zu analysieren. Nicht erkannte Bestandteile werden in der Strukturbeschreibung der Lemmakategorie mit der Kategorie *UK* angegeben. Dieses Vorgehen ist bei den X-Formen, bei denen durch entsprechende Zeichen (-, /, &,...) die Segmentierung bereits vorgegeben ist, unproblematisch. Für die komplexen WB-Formen ist dieses Vorgehen nicht in vollem Umfang praktikabel, da hier keine Segmentierung vorgegeben ist. Deshalb wird in diesem Fall bei der Analyse lediglich versucht das längstmögliche Endglied zu finden. Weitere Einschränkungen in der Anwendung dieses Verfahrens, das nur als letzte Analysemöglichkeit in Betracht gezogen wird, sind im Hinblick auf die bei der Segmentierung auftretenden Ambiguitäten (vgl. Abschnitt 3.6.5.2 auf Seite 143), bezüglich der Wortlänge und der Analyse des Hinterglieds zu machen. Da ein Wort, das bei der normalen Analyse nicht erkannt wird, in den meisten Fällen ein Eigenname ist, muß das Verfahren einer reinen Hintergliedanalyse so stark wie möglich restringiert werden. Die relevanten Parameter sind Wortlänge, der Kontext und die Art und Länge der erkannten Hinterglieder. Nur Wörter, die aufgrund ihrer Länge mit einiger Sicherheit Komposita sind (der Wert hängt davon ab, ob die Ergebnisse anschließend noch aufgrund des Kontextes überprüft werden oder nicht), werden auch segmentiert. Als Hinterglieder müssen die bereits in Abschnitt 3.6.5.2 auf Seite 143 erwähnten Wörter, die homograph mit gängigen Wortendungen sind und deshalb viele Mehrdeutigkeiten verursachen, ausgeschlossen werden.

3.7.1.4 Weitere Heuristiken

Falls eine Annotierung der nicht-erkannten (auch nicht in Teilen erkannten) Formen mit einem hypothetischen Lemma erwünscht ist, können bei den einzelnen Formen prototypische Funktionen definiert werden. Als Anhaltspunkte können die bereits bei der Disambiguierung gemachten Überlegungen dienen.

- WB-Formen: Die häufigsten nicht erkannten WB-Formen sind Eigennamen. Können Zitatkontexte ausgeschlossen werden und ist die Form im Text großgeschrieben, so ist die Wahrscheinlichkeit, daß es sich um einen Eigennamen handelt, groß. Zur weiteren Absicherung dieser Klassifikation kann noch der Kontext auf eine typische Eigennamenumgebung überprüft werden (vgl. dazu die Beschreibungen von Kategorien-typischen Kontexten in Abschnitt 3.6.5.3 auf Seite 145).
- CB-Formen: Einzelne Buchstaben werden generell nicht als unbekannt gewertet. So bleiben nur Konsonantenfolgen, die je nach Kontext und Form als Interjektionen oder als Abkürzungen klassifiziert werden können. Kontexte für Interjektionen lassen sich relativ einfach überprüfen, außerdem ist das Auftreten von mehr als zwei gleichen Buchstaben hintereinander ein weiterer Hinweis auf eine lautmalende Interjektion.

- GB-Formen: Für GB-Formen kommt typischerweise eine Klassifikation als Akronym bzw. Abkürzung in Frage. Aber auch Eigennamen sind häufig durch Großschreibung hervorgehoben. Hier kann wiederum nur anhand des Kontetxts eine Präferenz gefunden werden.
- MB-Formen: Für MB-Formen kommt eigentlich nur die Kategorie Abkürzung als Default in Frage. Vorher ist jedoch zu überprüfen, ob die gemischte Schreibweise nicht durch das Fehlen eines oder mehrerer Blanks zwischen WB-Formen zustande gekommen ist.
- ZSX-Formen: Für die ZSX-Formen bietet sich als Lösung die relativ nichtssagende Kategorie NUM an. Allerdings ist in diesem Bereich auch kaum mit unbekannten Formen zu rechnen.
- BSX-Formen: Bei den BSX-Formen tauchen unbekannte Formen als Komponenten komplexer Formen auf und bei gar nicht zu identifizierenden Formen, d.h. wenn die Form nicht in eines der Schemata paßt und bei Punktformen. Im ersten Fall genügt die Information über den Typ der Konstruktion für die meisten Zwecke vollauf, so daß auf eine Annotation der unbekannten Bestandteile verzichtet werden kann. Bei bislang noch nicht beobachteten Formtypen sollte meines Erachtens über den Versuch hinaus, eventuelle B-Form-Bestandteile zu identifizieren, keine weitere hypothetische Annotation stattfinden. Bei den Punktformen handelt es sich mit großer Wahrscheinlichkeit um Abkürzungen oder um komplexe Formen mit einer Abkürzung als Hinterglied. Lassen sich keine weiteren Bestandteile identifizieren, so sollte die Kategorie “Abkürzung” gewählt werden. Bei den Apostrophformen gibt es eine relativ einfach zu identifizierende Subklasse, die im Normalfall klar als Eigenname analysiert werden kann, nämlich die Formen, die mit *d’, dell’, l’, O’,* usw. beginnen, ebenso wie die Formen, die mit *’sch(e(mnrs))* enden.
- BZX-Formen: Bei den nicht-erkannten BZX-Formen ist der Anteil an fehlerhaften Formen im Vergleich zu den anderen Typen sehr hoch, obgleich die Wortformanalyse gerade bei diesen Formen sehr fehlertolerant ist. Bringt auch der Versuch die B-Anteile isoliert zu lemmatisieren keinen Erfolg, so sollte in diesem Fall kein weiterer Versuch einer Annotation mit einer Defaultkategorie unternommen werden.
- BSZX-Formen: Auch für die BSZX-Formen gilt, wie bei den BSX-Formen, daß eine unbekannte Form als Komponente einer komplexen Form nicht weiter analysiert wird. Ansonsten bietet sich auch hier die Klassifikation als Eigenname an, sofern dies durch den Kontext motiviert werden kann.

Insgesamt darf bei diesen Heuristiken jedoch nicht vergessen werden, daß auf diese Weise insbesondere bei Schreibfehlern eine große Anzahl von falschen Lemmatisierungen erzeugt wird, und daß im Einzelfall Vor- und Nachteile einer Lemmatisierung unbekannter Formen sorgfältig abgewägt werden müssen.

3.7.2 Fehlerhafte Formen

Ein weiteres Problem bei der Lemmatisierung von rohem Text stellen die fehlerhaften Formen des Textes dar. Diese Formen werden entweder gar nicht erkannt oder aber, was wesentlich problematischer ist, sie werden falsch analysiert. Obwohl es gerade im Bereich der Sonderformen sehr viele falsche Formen in Texten zu finden gibt, wirken sie sich hier kaum aus, da die Analyse dieser Formen in der Regel eine Prüfung auf Wohlgeformtheit mit einschließt, so daß fehlerhafte Formen in diesem Bereich in den meisten Fällen als unbekannt lemmatisiert werden oder sogar korrigiert werden. Bei den WB-Formen ist es jedoch so, daß versucht wird, alle denkbaren Analysemöglichkeiten zu überprüfen. Die Filter sind so konzipiert, daß sie nie alle Analysen ausschließen, sondern auch relativ unwahrscheinliche, aber theoretisch mögliche Analysen beibehalten, wenn es keine besseren Alternativen gibt. Diese Strategie des Soviel-wie-möglich-Analysierens erweist sich im Falle von Fehlern in WB-Formen als ausgesprochen nachteilig, wie die folgenden Fehlanalysen zeigen:

Gepräche: ge||prä||che.EN:X##PREF,N,EN

Aabischen: a||ab||ische.fem(NS0,NP4):nmF:gmF:dmF:amF##PREF,PREP,N

Die Kategorienabfolgen der beiden Beispiele an sich sind nicht generell auszuschließen. Im ersten Beispiel würde die Zusatzinformation, daß es sich um einen Vornamen handelt, sicher einen Grund liefern, die Abfolge als fragwürdig zu markieren. Doch auch hierfür besteht im zweiten Beispiel kein Anlaß. Beiden Beispielen gemein ist jedoch, daß es sich bei den N-Komponenten um seltene (zumindest mit dieser Wortart) einfache Formen handelt. Setzt man die vollständige Klassifikation der EF-Formen nach Grundwortschatz, erweitertem Grundwortschatz und ungebräuchlichen Formen voraus, so könnten auch auf diese Weise fragwürdige Lemmatisierungen identifiziert werden. Ein weiteres Indiz ist in beiden Fällen die geringe durchschnittliche Länge der Komponenten (2,7 bzw. 3) verglichen mit der durchschnittlichen Länge von 11,5 bezogen auf alle EF-Formen. Faktoren dieser Art können als Anhaltspunkte dienen, bestimmte Lemmatisierungen ganz auszuschließen, wenn dies gewünscht wird, bzw. sie in einem interaktiven Modus zu überprüfen.

3.8 Auswertung des Testkorpus

In diesem Kapitel sollen einige Ergebnisse, die das Lemmatisierungsprogramm für das Testkorpus liefert, vorgestellt und diskutiert werden. Das Ergebnis der Lemmatisierung eines größeren Textausschnitts ist in Anhang F auf Seite 211ff abgedruckt.

3.8.1 Bestand an Formen

Das Testkorpus enthält 133 310 Sätze und 2 208 414 Textformen. Eine einzelne Ausgabe der Süddeutschen Zeitung enthält durchschnittlich ungefähr 5500 Sätze mit ca. 100 000 A-Formen (A-Formen im Sinne von Abschnitt 3.2), denen im Durchschnitt ca. 92 000 Textformen zugrundeliegen. Die einzelnen Formtypen verteilen sich im Testkorpus bezogen auf die Menge der A-Formen wie folgt:

Tabelle 14. Verteilung der Formtypen

Formtyp	Anteil im Text
WB-Formen	82,11%
SX-Formen	13,04%
BSX-Formen	1,53%
ZX-Formen	1,42%
ZSX-Formen	0,72%
MB-Formen	0,45%
GB-Formen	0,44%
IX-Formen	0,14%
BZX-Formen	0,06%
CB-Formen	0,05%
BSZX-Formen	0,04%

Die B-Formen stellen im Text mit 83,05 % demnach den Hauptanteil gegenüber 16,95 % Sonderformen.

Der Anteil der bei der Lemmatisierung anhand des Lexikons nicht erkannten Wörter liegt bei 2,2% (insgesamt 54 696 A-Formen), wobei zu bemerken ist, daß es im Bereich der X-Formen schon vom Programm her kaum vorkommen kann, daß eine Form als vollständig unbekannt analysiert wird, es sei denn sie paßt auf keines der beschriebenen Muster. Eine genauere Aufschlüsselung der nichterkannten Formen (bezogen auf eine Ausgabe mit insgesamt 2 039 nicht erkannten Formen) befindet sich in Tabelle 15.

Tabelle 15. Unbekannte Formen bezogen auf eine Ausgabe

Art der Form	Anzahl Types	Prozent (bezogen auf Types)	Anzahl Tokens	Prozent (bezogen auf Tokens)
Unbekannte Formen insges.	1 120	100 %	2 039	100 %
Vornamen	56	5,0 %	91	4,5 %
Nachnamen	224	20,0 %	355	17,4 %
geograph. Namen	355	31,7 %	864	42,4 %
Firmennamen	4	0,4 %	5	0,2 %
sonst. Namen	119	10,6 %	192	9,4 %
Eigennamen insgesamt	758	67,7 %	1 507	73,9 %
fremdsprachige Wörter	178	15,9 %	252	12,4 %
Tippfehler	61	5,5 %	68	3,3 %
Abkürzungen	51	4,6 %	113	5,5 %
Dialekt-Wörter	44	3,9 %	63	3,1 %
einfache oder komplexe Formen	26	2,3 %	34	1,7 %
Sonderformen	2	0,2 %	2	0,1 %

Die Zahlen in Tabelle 15 zeigen deutlich, daß die Rate der nicht-erkannten Wörter durch eine weitere Verbesserung des Eigennamenlexikons entschieden gesenkt werden kann. Fast dreiviertel der nicht-erkannten Formen stammen aus dem Eigennamenbereich, wobei die geographischen Namen besonders stark vertreten sind. Erfreulich im Hinblick auf das CISLEX-DKL ist die Tatsache, daß die einfachen und komplexen Formen unter den nicht-erkannten Formen mit nur 1,7 % einen sehr geringen Teil ausmachen. Ein Problem stellen jedoch die vielen fremdsprachigen Formen dar, die nicht durch einfache Lexikonerweiterung in den Griff zu bekommen sind. Hier sollte in der Zukunft vor allem eine genaue Erkennung von Zitatkontexten angestrebt werden, um auf diese Weise einen Großteil dieser Formen entsprechend markieren zu können.

3.8.2 Art und Anzahl der Analysen

Um einen Überblick über die Anzahl der Ambiguitäten und ihre mögliche Einschränkung zu erhalten, wurde das Testkorpus zweimal lemmatisiert. Einmal mit Disambiguierung und einmal ohne Disambiguierung. Bei einer Lemmatisierung ohne anschließende Disambiguierung ergeben sich durchschnittlich 2,28 Analysen[1] je A-

Form im Text. Wird nach der Lemmatisierung noch disambiguiert, so bleiben durchschnittlich 1,2 Analysen je A-Form im Text übrig.

Tabelle 16. Anzahl der Analysen je A-Form bezogen auf das Testkorpus

Anzahl der Analysen	Anteil in % ohne Disambiguierung	Anteil in % mit Disambiguierung
1	38,05%	84,06%
2	24,86%	11,9%
3	21,55%	3,15%
4	8,87%	0,6%
5	3,82%	0,06%
> 5	2,85%	0,23%
6	1,33%	
7	0,47%	
8	0,34%	
9	0,22%	
> 10	0,46%	

Die Werte in Tabelle 16 zeigen, daß durch die in Abschnitt 3.6 auf Seite 131 und Abschnitt 4.6 auf Seite 189 beschriebenen Regeln und Heuristiken in 84% der Fälle eine eindeutige Analyse erreicht, gegenüber nur 38% ohne Disambiguierung. Zur weiteren Veranschaulichung, enthält Tabelle 16 einen Überblick über die Analysen der B-Formen innerhalb einer Ausgabe (insgesamt 103 937 A-Formen). Die Gesamtzahl der B-Formen betrug in dieser Ausgabe 86 324.

Tabelle 17. Anzahl der Analysen je Typ bezogen auf die B-Formen einer Ausgabe der SZ

Analysen als	Gesamtanzahl ohne Disambiguierung	Gesamtanzahl mit Disambiguierung
einfache Form	174 278	83 558
komplexe Form	55 108	11 106
Abkürzung	28 218	1 293
Eigenname	44 006	2 279

Der große Unterschied bei der Zahl der Analysen als Abkürzung beruht zum Teil darauf, daß der Look-up im Abkürzungslexikon verschiedene Alternativen liefert. Ausschlaggebender ist jedoch die Präferenz von Analysen als einfache und komplexe Form bei der Disambiguierung. Ohne Disambiguierung werden alle B-Formen, insbesondere, die in Texten sehr häufigen kurzen Funktionswörter im Abkürzungslexikon

1. Die UK-Analysen für Formen, die nicht identifziert werden konnten, sind bei der Anzahl der Analysen mitgerechnet.

nachgeschlagen. Ähnlich fällt auch bei den Eigennamen die Homographie zwischen Eigennamen und Nomen stark ins Gewicht.

Bei allen diesen Betrachtungen darf jedoch nicht vergessen werden, daß sich die Zahlen auf das absolute Vorkommen im Text beziehen. Untersucht man nur die WB-Formen einer Ausgabe, so ergibt sich folgende Verteilung von tatsächlich verschiedenen Formen (bezogen auf die Gesamtmenge von 85 346 verschiedenen WB-Formen):

Tabelle 18. Verteilung der WB-Formen bezogen auf eine Ausgabe

Art der Formen	Anzahl
verschiedene WB-Formen insgesamt	16 867
verschiedene EF-Formen insgesamt	8778
verschiedene EF-Grundformen insgesamt	6121
verschiedene komplexe Formen insgesamt	5334
verschiedene komplexe Grundformen insgesamt	4991

Unter “verschiedene EF- bzw. komplexe Grundform” sind die möglichen Lemmatisierungen der verschiedenen Formen zu verstehen.

4 Implementierungsaspekte

In diesem Kapitel soll auf einige Details der Implementierung eingegangen werden, sowohl was die interne Repräsentation des Lexikons betrifft, als auch den Lemmatisierer.

Der Lemmatisierer ist als Prototyp konzipiert, bei dem es in erster Linie darum geht, zu demonstrieren, wie die Daten, die im Testkorpus vorkommen, maschinell mit Hilfe des CISLEX-Lexikons vollständig zu analysieren sind. Effizienzgesichtspunkte standen weder bei der internen Repräsentation des Lexikons noch bei der Implementierung des Lemmatisierungsalgorithmus im Vordergrund. Der Lemmatisierer ist in *Perl* implementiert (Wall/Schwartz 1993). Der Grund für die Auswahl dieser Implementierungssprache waren die umfangreichen Möglichkeiten zur Textbearbeitung, unter anderem die Möglichkeit der Verwendung von regulären Ausdrücken in Suchmustern. Vom Aufbau her ist die Sprache Perl ähnlich zu C, jedoch, was die Datenstrukturen betrifft, wesentlich einfacher. Außerdem sind unter Perl eine Reihe der UNIX-Textverarbeitungsroutinen, wie beispielsweise *sed* oder *tr* verfügbar. Des weiteren verfügt Perl über einen einfachen Zugriff auf DBM-Datenbanken. Damit bietet sich als Arbeitsgrundlage die in Abschnitt 2.4.4 auf Seite 71 bereist beschriebene DBM-Version des FLEX-Lexikons an.

4.1 Lexikonzugriff

Für die Lemmatisierung werden, wie bereits in Kapitel 2 beschrieben, neben dem EF-FLEX-Lexikon noch weitere Lexika benötigt. Es handelt sich bei diesen Lexika einesteils um relativ umfangreiche Lexika, die von verschiedenen Funktionen bei der Lemmatisierung angesprochen werden, und zum anderen um kleine Speziallisten, die nur innerhalb einer speziellen Funktion benötigt werden. Diese Speziallisten sind im Rahmen der einzelnen Funktionen kodiert und werden in Anhang E auf Seite 209 mit Beispielen aufgeführt. Die Lexika, die von mehreren Funktionen abgefragt werden, liegen als DBM-Datenbank in einfacher Attribut-Wert-Struktur vor. Als Schlüssel dient die flektierte Form, als Wert sind die verschiedenen, zu dieser Form im jeweiligen Lexikon vorhandenen Einträge (durch "/" separiert) abgespeichert. Dieses Format wurde auch für die zur Lemmatisierung notwendigen Ausnahmelexika (vgl. Anhang E auf Seite 209ff) verwendet.

Als global zugängliche Lexika im DBM-Format werden beim Look-up folgende Lexika benötigt:

- EF-Flex-Lexikon: Die EF-DBM enthält als Schlüssel alle flektierten Formen zu Einträgen des DKL-EF+. Der Wert einer flektierten Form in der EF-DBM sind alle möglichen Einträge zu dieser Form im DKL-FLEX. Die EF-Formen werden in der Datenbank alle in normalisierter Form, d.h. kleingeschrieben abgespeichert. Alternative Einträge sind, wie bereits erwähnt, durch “/” abgetrennt. Beispiele zu den Einträgen in der EF-DBM befinden sich in Anhang A auf Seite 198.

- Abkürzungslexikon: Die AK-DBM enthält alle Abkürzungen des CISLEX-AK. Als Schlüssel dient die Abkürzung in kleingeschriebener Form. Der Wert besteht aus allen möglichen Abkürzungen, die der normalisierten Form entsprechen, in ihrer konventionalisierten Schreibweise, zusammen mit der Kategorie “AK”. (Für Beispieleinträge siehe Anhang C auf Seite 205).
- Eigennamenlexikon: Die Eigennamen des CISLEX-EN werden in der EN-DBM ohne weitere Zusatzinformation abgespeichert. Die Schlüssel sind die kleingeschriebenen Eigennamenformen, die Werte sind die Eigennamengrundformen zusammen mit der Kategorie “EN” oder gegebenenfalls einer den EF-Formen entsprechenden Kategorie mit den entsprechenden morphosyntaktischen Merkmalen.[1] (Für Beispieleinträge siehe Anhang C auf Seite 205)
- Affixlexikon: Das Affixlexikon enthält die Präfixe und Suffixe mit Ausnahme der auf numerische Konstruktionen eingeschränkten Affixe. Bei Präfixen ist in der AFFIX-DBM als Schlüssel das Präfix selbst und als Wert das Präfix mit der Kategorie “PREF” abgespeichert. Bei den Suffixen ist der Schlüssel die flektierte Form des Suffix. Als Wert dient die Grundform und die Kategorie zusammen mit den morphosyntaktischen Merkmalen. Die Kategorie eines Suffix richtet sich nach der Kategorie, die die entsprechende Derivation hat. Das Kategoriensymbol ist das Kategoriensymbol der Derivation erweitert durch “SUFF”. Die morphologische Kodierung und die morphosyntaktischen Merkmale entsprechen ebenfalls der Hauptkategorie. (Für Beispieleinträge siehe Anhang D auf Seite 206)
- Lexikon der numerischen Ausdrücke: Die NUMLEX-DBM enthält alle numerischen Ausdrücke des CISLEX. Das sind sowohl Kardinal- und Ordinalzahlwörter, als auch die entprechenden Affixe. Wie bereits bei den anderen DBMs dient auch hier als Schlüssel die flektierte Form. Die Darstellung der numerischen Affixe ist analog zum Affixlexikon. Bei den Zahlwörtern ist zusätzlich zur kategorialen und morphosyntaktischen Information noch die Information über den Exponenten und den Koeffizienten der entsprechenden Zehnerpotenzdarstellung abgespeichert (vergl. Abschnitt 3.3.1.3 auf Seite 102). (Für Beispieleinträge siehe Anhang D auf Seite 206).

1. Bislang wurden nur für Ländernamen und davon abgeleitete Adjektive und Personenbezeichnungen morphologische Kodes vergeben. Die anderen Eigennamen werden in der Grundform aufgeführt.

4.2 Das Lemmatisierungsprogramm

In den folgenden Kapiteln soll das Lemmatisierungsprogramm mit seinen wichtigsten Funktionen und deren Input/Output-Verhalten ausführlich beschrieben werden. Ein vollständiges Listing des Programms befindet sich in Anhang I auf Seite 223ff. Die Beschreibung erfolgt teilweise in einer Art Pseudo-Perl. Wenn möglich wurde aber versucht, die Beschreibung weitgehend unabhängig von der Implementierungssprache zu halten. Einige Konventionen wurden allerdings übernommen, die im folgenden kurz erläutert werden.

Perl kennt drei Typen von Variablen: skalare Variablen, die mit einem $-Zeichen beginnen, Listen- oder Array-Variablen, die mit einem @-Zeichen beginnen und Variablen für assoziative Vektoren, die mit einem %-Zeichen beginnen. Mit der Variable $liste[n] kann auft das n-te Element der Liste @liste zugegriffen werden, wobei n eine natürliche Zahl oder eine skalare Variable für eine natürliche Zahl ist. Assoziative Arrays sind Schlüssel-Wert-Paare, die intern als Hashtable realisiert werden. Die Variable $assoz{$key} enthält den Wert von $key im assoziativen Array %assoz. Mit Hilfe assoziativer Arrays ist auch der Zugriff auf die Lexikon-DBMs realisiert. Die Schlüssel-Wert-Struktur der DBMs wird unter Perl wie ein assoziativer Array behandelt. Unterprogrammaufrufe sind durch & vor dem Namen des Unterprogramms markiert. Kommentare werden in Perl mit dem Zeichen “#” am Beginn markiert.

Weiterhin werden bei der Beschreibung reguläre Ausdrücke verwendet, wie sie in Wall/Schwartz (1993) beschrieben werden. Die regulären Ausdrücke können in Perl sowohl im Rahmen von Bedingungen als auch bei Suche-/Ersetzeoperationen verwendet werden. In den regulären Ausdrücken werden folgende Variablen für spezielle Mengen von Buchstaben oder Zeichen verwendet:

$typ_buchstabe = "[ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzÀÁÂÃÄÅÇÈÉÊËÌÍÎÏÑÒÓÔÕÖÙÚÛÜàáâãäåçèéêëìÆíîïñŒòóôõöæùúûıüøœßÿ]";

$typ_majuskel = "[ABCDEFGHIJKLMNOPQRSTUVWXYZÀÁÂÃÄÅ ÇÈÉÊËÌÍÎÏÑÒÓÔÕÖÙÚÛÜÆŒ]";

$typ_minuskel = "[abcdefghijklmnopqrstuvwxyzàáâãäåçèéêëìíîïñòóôõöæùúûıüøœßÿ]";

$typ_alphanum = "[0-9]|$typ_buchstabe";

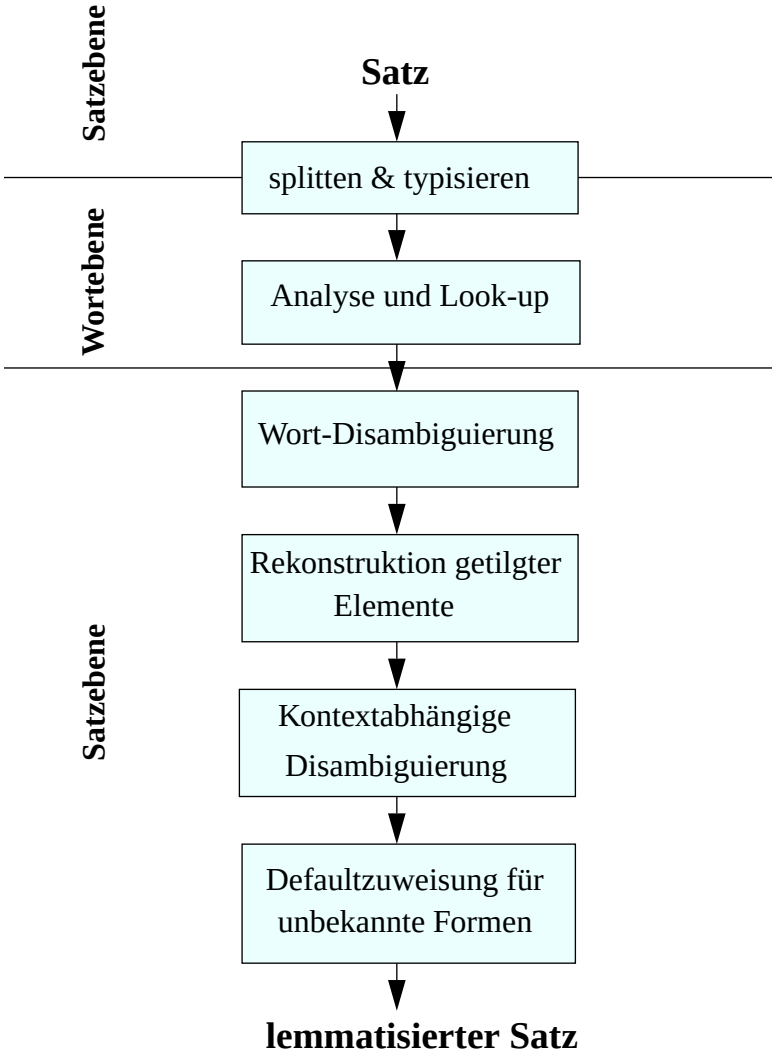
$typ_vokal = "[AEIOUYaeiouyÀ-ÏÒ-Üà-åè-ìíîïò-üœ]";

$typ_konsonant = "[BCDFGHJKLMNPQRSTVWXZbcdfghjklmnpqrstvwxzÇÑçß]";

4.3 Ablauf der Lemmatisierung

Das Lemmatisierungsprogramm arbeitet Satz-orientiert. Als einzige Vorverarbeitung wird mit dem Satzendeerkennungsprogramm (Schicht 1994b) der Text so umformatiert, daß jede Zeile genau einen Satz (oder eine Überschrift) enthält und die Satzendepunkte zur Unterscheidung von Abkürzungspunkten am Satzende mit Blank abgetrennt sind.

Abbildung 6. Lemmatisierungsschritte



Das Lemmatisierungsprogramm besteht im wesentlichen aus einer Hauptschleife, die alle für die Lemmatisierung notwendigen Funktionen ausführt. Außerhalb dieser

Hauptschleife werden lediglich die Flags für die Disambiguierung und die Defaultzuweisung gesetzt. Außerdem werden die Lexikon-DBMs für die Abfragen geöffnet.

Die Hauptschleife des Programms liest solange zeilenweise aus dem aktuellen Input, bis dort das EOF-Zeichen erreicht ist (in Perl durch die Bedingung “while(<>)” realisiert). Durch die Vorverarbeitung ist gewährleistet, daß auf diese Weise immer vollständige Sätze oder Überschriften eingelesen werden.

```
while(<>) {

  @a_formen = &splitte;

  while(@a_formen){

      &lemmatisiere;}

  &delayed_rekonstruiere_RKoord;

  &disambiguiere;

  &default_UK;

}
```

Als globale Listenvariablen werden in der Hauptschleife, bzw. in den einzelnen Subroutinen folgende für die Verarbeitung relevanten Listen erstellt:

- @a_formen: Das Ergebnis der Tokenisierung ist die Liste @a_formen, die alle A-Formen des Eingabesatzes in der Reihenfolge ihres Auftretens enthält.
- @erg_lemma: Das Ergebnis der Formanalyse der einzelnen Elemente wird in der Liste @erg_lemma unter demselben Index abgespeichert.
- @erg_delay_RKoord: Die vorläufigen Ergebnisse der Formanalyse von Formen mit rechtsseitigem Koordinationsbindestrich werden zusammen mit dem Index in @a_formen in der Liste @erg_delay_RKoord solange zwischengespeichert, bis eine komplexe Form oder ein Bindestrichwort analysiert werden konnte. Dann wird diese Liste zusammen mit dem Ergebnis der Analyse der komplexen Form oder des Bindestrichworts zur Rekonstruktion an **delayed_rekonstruiere_RKoord** übergeben. Nach erfolgreicher Rekonstruktion eines Elements wird dessen Eintrag aus der Liste gelöscht.
- @erg_typ: Zusätzlich zur Liste der Analyseergebnisse wird in @erg_typ am jeweiligen Index abgespeichert, welcher Art das Ergebnis in @erg_lemma ist. Jedes Element von @erg_typ ist eine Liste, die zu einem String verkettet wurde. Diese Liste hat 10 Stellen, von denen die letzten Stellen 9 nur binäre Werte annehmen. Die erste Stelle enthält den Formtyp. Die folgenden Stellen stehen für folgende Analysemöglichkeiten: EF-Form, komplexe Form, Bindestrichwort, Numeral, Ordinalzahl, Abkürzung, Eigenname, Zeichen und UK (unbekannt). Falls eine Analyse des jeweiligen Typs möglich ist, ist der Wert an der entsprechenden Stelle 1, andernfalls 0.

- @erg_ambig : Diese Liste enthält für jede A-Form einen Marker, ob bei der Formanalyse Ambiguitäten aufgetreten sind. “W” steht für Wortartambiguitäten, “L” für Ambiguitäten der Grundfom oder morphologischen Klasse bei gleicher Wortart, “WL” für Ambiguitäten beider Art und “0”, falls die Formanalyse ein eindeutiges Ergebnis geliefert hat. Diese Liste und @erg_kats wird zur Erkennung von Positionen, an denen kontextabhängige Disambiguierung stattfinden kann, benötigt.
- @erg_kats: Diese Liste enthält am jeweiligen Index die möglichen Wortarten, die die Formanalyse ergibt. Bei X-Formen ist die Kategorie oder Wortart des letzten Elements ausschlaggebend (bei Suffixen wird allerdings nicht die Kategorie “Suffix” angenommen, sondern die Wortart des Ergebnisses).

In Anhang H auf Seite 221 ist anhand eines Beispielworts in einem minimalem Kontext ein vollständiger Programmablauf mit allen notwendigen Funktionsaufrufen beschrieben.

4.4 Einlesen und Tokenisierung

Jede aus dem Standard-Input eingelesene Zeile wird als String der Routine **splitte** übergeben. Diese gibt einen Array mit den zu analysierenden A-Formen zurück.

splitte

Input: Satz als String

Output: Liste der A-Formen. Jede A-Form besteht aus der normalisierten-Form und dem WS-Vektor.

Splitte untersucht den Eingabestring auf wortübergreifende Klammerstrukturen und separiert erkannte Klammern durch Leerzeichen. Durch Blanks unterteilte Zahlen werden zusammengeführt. Dann wird der Eingabe-String an den Leerzeichen aufgesplittet in die Liste der Textformen und diese elementweise an **del_satzzeichen** übergeben, das als Ergebnis die Liste der zu analysierenden Formen liefert. Die zu analysierenden Formen werden in **ws_vektor** typisiert und der entsprechende Wortstrukturvektor erstellt. Die Form wird normalisiert, d.h. alle Buchstaben werden in Kleinbuchstaben umgewandelt.

```
splitte(24 000 neue Kunden eröffneten ein Sparbuch; die
Spareinlagen stiegen um 150 Millionen Mark .)=

  24000@ZX:24000:5:0:0:0:::0:SA,
  neue@WB:neue:4:0:0:0:::1:ZX,
  kunden@WB:Kunden:6:0:1:0:::0:WB,
  eröffneten@WB:eröffneten:10:0:0:0:::1:WB,
  ein@WB:ein:3:0:0:0:::1:WB,
  sparbuch@WB:Sparbuch:8:0:1:0:::0:WB,
  ;@SX:;:1:0:0:0:::0:WB,
  die@WB:die:3:0:0:0:::1:SX,
  spareinlagen@WB:Spareinlagen:12:0:1:0:::0:WB,
  stiegen@WB:stiegen:7:0:0:0:::1:WB,
  um@WB:um:2:0:0:0:::1:WB,
  150@ZX:150:3:0:0:0:::0:WB,
  millionen@WB:Millionen:9:0:1:0:::0:ZX,
  mark@WB:Mark:4:0:1:0:::0:WB,
  .@SX:.:1:0:0:0:::0:WB[1]
```

del_satzzeichen

Input: Textform als String

1. Die Beispiele zu den einzelnen Funktionen sind in der Form f(x) = y dargestellt, wobei f der Name der Routine ist, x der Input und y der Output.

Output: Liste der zu analysierenden Formen

Diese Prozedur isoliert alle nicht für die Form relevanten Zeichen und gibt die Liste der isolierten Zeichen und Formen zurück. Die Eingabeform wird auf folgende Zeichen untersucht: *:!;,?[(")]'*. Die Form wird unverändert zurückgegeben, wenn sie weder am Anfang noch am Ende eines dieser Zeichen enthält. Ansonsten werden folgende Fälle unterschieden:

1. Satzzeichen am Ende: Am Formende werden !;,?: als neue Formen abgetrennt und **del_satzzeichen** mit der verbleibenden Form wieder aufgerufen.
2. Numerierungsform: Wenn die Form dem Muster /^([\(\[])?([1-9]|$typ_alphanum+)([\)\]])$/ entspricht, wird sie ebenfalls unverändert zurückgegeben und dann als Typ IX für Numerierung weiterverarbeitet.
3. Klammerzeichen am Ende (und keines am Anfang): Wenn die Form dem Muster /^([\(\[\"])(.+)[^\)\]\"]$/ entspricht, wird getestet, ob die Endform das schließende Klammerzeichen enthält. Dann wird die gesamte Form zurückgegeben, andernfalls wird die Anfangsklammer als Form abgetrennt und der Rest wieder an **del_satzzeichen** übergeben.
4. Klammerzeichen am Anfang und am Ende: Wenn die Form auf das Muster /^([\(\[\"])(.+)([\)\]\"])$/ paßt und die beiden Klammern ein Paar bilden, werden die Klammern als Formen abgetrennt und die verbleibende Form wieder an **del_satzzeichen** übergeben. Handelt es sich um verschiedene Klammern wird geprüft, ob sich im Wortinnern eine duale Klammer zu einer der beiden Randklammern befindet. Wenn ja wird die Klammer ohne Pendant abgetrennt. Wenn nicht, werden beide Klammern abgetrennt und der Rest an **del_satzzeichen** übergeben.
5. Wenn die Form in Hochkommas steht, werden diese als neue Formen abgetrennt und die verbleibende Form wieder an **del_satzzeichen** übergeben.[1]
6. Klammerzeichen am Anfang (und keines am Ende): Dieser Fall verläuft analog zu Fall 4) mit der Klammer am Anfang.

Ouput ist die Liste der so separierten Formen.

```
del_satzzeichen(K(ritische)-Punkt) = K(ritische)-Punkt

del_satzzeichen((!),) = (, !, ), ,

del_satzzeichen((+28,7%),)= (, +28,7%, ), ,

del_satzzeichen((Kleinottweiler),) = (, Kleinottweiler, ), ,
```

1. Für Hochkommas ist eine Extrabehandlung notwendig, da unpaarig auftretende Hochkommas nicht abgetrennt werden dürfen, sondern als Apostroph zur A-Form gehören.

ws_vektor

Input: Analyseform als String

Output: Wortstrukturvektor

Der Wortstrukturvektor enthält folgende Informationen, die soweit bekannt innerhalb der Prozedur **ws_vektor** eingetragen werden (die Merkmale, die erst während des weiteren Programmablaufs bestimmt werden, bleiben vorerst unbelegt):

Typ, Originalorthographie, Länge der Form, Großschreibung am Satzanfang, Großschreibung in der Satzmitte, Großschreibung nach IX-Typen oder einem der Zeichen :-!?, Reihungsform[1] am Anfang groß, Reihungsform im Inneren groß, Reihungsform mit letztem Element groß, Kleinschreibung, Vorgängertyp, mögliche Kategorien des Vorgängers, mögliche Kategorien des Nachfolgers, Mehrfachanalyse, wortartambig, Grundform- oder morphologisch ambig, mögliche Kategorien.

Der Typ einer Form wird durch **formtyp** bestimmt, das die Form mittels regulärer Ausdrücke auf die in “Typen von A-Formen” auf Seite 96 beschriebenen Typen überprüft.

1. Als Reihungsform werden Bindestrich-, Slash-, &- und +-Reihungen bezeichnet, bei denen durch die entsprechenden Zeichen separat zu lemmatisierende Formen abgetrennt werden (vgl. auch die Ausführungen in Abschnitt 3.4.7 auf Seite 116ff). In diesen Fällen ist im Gegensatz zu Komposita nicht immer die Großschreibung der Anfangskomponente relevant.

4.5 A-Formenanalyse

Die Tokenisierung liefert eine Liste der A-Formen, die elementweise an **lemmatisiere** zur eigentlichen Lemmatisierung übergeben wird. Innerhalb der **lemmatisiere**-Funktionen werden durch die Funktionen vom Typ **look_up** die verschiedenen CISLEX-Komponenten angesprochen, bzw. die Regeln, die auf den Einträgen des CISLEX operieren, aufgerufen.

4.5.1 Die **lemmatisiere**-Funktionen

lemmatisiere

Input: Form als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Übergibt den Inputstring und den WS-Vektor in Abhängigkeit vom Typ (im WS-Vektor kodiert) an die spezielle Lemmatisierungsfunktion: **lemmatisiere_CB**, **lemmatisiere_WB, lemmatisiere_GB**, **lemmatisiere_MB**, **lemmatisiere_ZX**, **lemmatisiere_SX**, **lemmatisiere_ZSX**, **lemmatisiere_BZX**, **lemmatisiere_BSZX**, **lemmatisiere_BSX**.[1] Eine schematische Darstellung der Lemmatisierung der B- und X-Formen befindet sich am Ende dieses Kapitels auf Abbildung 7, “Lemmatisierung der B-Formen”, auf Seite 176 und Abbildung 8, “Lemmatisierung der X-Formen”, auf Seite 177. Beispiele für die Lemmatisierung von B-Formen sind in der Analyse eines Beispielstextes (Anhang F auf Seite 211ff) zu finden. Anhang G auf Seite 219 enthält einige Beispiele zur Lemmatisierung der Sonderformen.

lemmatisiere_CB

Input: CB-Form als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Überprüft zuerst, ob es sich um die Präposition *à* handelt, dann wird als Lemmatisierung *à.PREP0* zurückgegeben. Ansonsten wird **look_up_ak** aufgerufen, das im Abkürzungslexikon nachschlägt. **look_up_ak** liefert als Ergebnis die Liste der möglichen Abkürzungslemmata, falls nichts gefunden wird, die leere Liste. Falls die Form weder als Präposition noch als Abkürzung zu lemmatisieren war, wird als Lemma *zeichen(<form>).CC* als einziges Element der Liste zurückgegeben.

```
lemmatisiere_CB(é,CB:é:1:0:0:0::::1:SA) = [é]:zeichen(é).CC
```

1. Der Zusatz CB, WB, GB, MB, ZX, SX, ZSX, BZX, BSZX und BSX bezieht sich auf die in Abschnitt 3.2.3 auf Seite 96ff beschriebene Unterscheidung der A-Formen nach ihren Bestandteilen.

```
lemmatisiere_CB(à,CB:à:1:0:0:0:::1:SA) = [à]:à.PREP0

lemmatisiere_CB(v,CB:v:1:0:0:0:::1:SA) = [v]:v.AK:X
```

lemmatisiere_WB

Input: WB-Form als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Führt die verschiedenen Look-ups im CISLEX aus. Falls die Disambiguierungsoption gewählt wurde, wird das Ergebnis jedes Look-ups anschließend disambiguiert. Zuerst wird im EF-Lexikon (**look_up_ef**) nachgeschlagen und das Ergebnis, die Liste der möglichen EF-Lemmata, gegebenenfalls mit **filter_gross** bzw. **filter_klein** disambiguiert, je nachdem, ob die Textform groß- oder kleingeschrieben war. Danach wird versucht, die Inputform in **look_up_komp** als komplexe Form zu lemmatisieren. (Ist die Disambiguierungsoption gewählt, so werden, sobald ein Look-up erfolgreich war, keine weiteren Lemmatisierungsversuche mehr unternommen.) Gegebenenfalls wird das Ergebnis von **look_up_komp** mit **filter_gross** bzw. **filter_klein** und **filter_seg** disambiguiert. Anschließend wird versucht, mit **look_up_ak** und **look_up_en** im Abkürzungs- und Eigennamenlexikon Lemmatisierungen zu finden, die dann gegebenenfalls mit **filter_aehnlich** disambiguiert werden. Die Ergebnisse der verschiedenen Look-ups werden in einer Liste gesammelt. Konnte keine Lemmatisierung gefunden werden, so wird in **look_up_kompEN** die Form auf Kompositum mit Eigennamenbestandteilen überprüft. Als Ergebnis wird die Liste der möglichen Lemmatisierungen zurückgegeben. Falls keine Lemmatisierung gefunden wurde, enthält die Ergebnisliste nur ein Element: *WB(<form>).UK*.

```
lemmatisiere_WB(memme,WB:Memme:5:1:0:0:::0:SA) =
[memme]:memme.fem(NS0,NP4):neF:geF:deF:aeF

lemmatisiere_WB(schweinsbraten,WB:Schweinsbraten:14:1:0:0:::
:0:SA)=
[schweinsbraten]:schweins||braten.mask(NS2,NP0):neM:deM:aeM:
nmM:gmM:dmM:amM##N,N
[schweinsbraten]:schweins||braten.VST6:OI:1mGi:3mGi:1mGc:3mG
c##N,V
[schweinsbraten]:schweins||braten.neut(NS2,NPSG):neN:deN:aeN
##N,N

lemmatisiere_WB(bayer,WB:Bayer:5:1:0:0:::0:SA) =
[bayer]:bayer.EN:X
```

lemmatisiere_GB

Input: GB-Form als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Führt dieselben Look_ups und mit Ausnahme von **filter_gross** und **filter_klein** bei Disambiguierungsoption dieselben Filter aus wie **lemmatisiere_WB**. Nur die Reihenfolge der Look-ups ist verändert, um Abkürzungen und Eigennamen zu präferieren. Die Reihenfolge der Look-ups ist: **look_up_ak, look_up_en, look_up_ef, look_up_komp, look_up_kompEN.**

```
lemmatisiere_GB(aids,GB:AIDS:4:1:0:0:::0:SA) =
  [aids]:AIDS.AK:X
```

lemmatisiere_MB

Input: MB-Form als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Mit **look_up_ak** und **look_up_en** wird versucht die Eingabeform als Abkürzung bzw. Eigenname zu lemmatisieren. Falls die Originalorthographie der Eingabeform auf *In* oder *Innen* endet und der Anfangsstring keine nicht-initiale Großschreibung enthält, wird sowohl der Anfangsstring als auch Anfangsstring+*in* bzw. +*innen* nochmals mit **ws_vektor** typisiert, mit jeweils eigenem WS-Vektor versehen und mit beiden wird dann nochmals **lemmatisiere** aufgerufen. Das Ergebnis beider Aufrufe wird der Ergebnisliste angefügt.

Konnte keine Analyse gefunden werden und enthält die Originalorthographie keine aufeinanderfolgenden Großbuchstaben, so wird die Form in Teile, die mit Großbuchstaben beginnen und dann nur Kleinbuchstaben enthalten aufgespalten, die dann einzeln typisiert und lemmatisiert werden.

```
lemmatisiere_MB(gmbh,MB:GmbH:4:1:0:0:::0:SA) = GmbH.AK:X

lemmatisiere_MB(leserinnen,MB:LeserInnen:10:1:0:0:::0:SA)=
  [leser]:leser.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM
  [leserinnen]:leserin.fem(NS0,NP5):nmF:gmF:dm
```

lemmatisiere_ZX

Input: ZX-Form als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Für eine Ziffernfolge gibt es zwei Möglichkeiten: Sie ist der Form nach eine Zahl, d.h. sie beginnt nicht mit *0*, oder sie beginnt mit *0*. Im ersten Fall wird als Lemma *Integerzahl(<form>).NUM* und im zweiten Fall *ziffern(<form>).NUM* als einziges Element der Ergebnisliste zurückgegeben.

```
lemmatisiere_ZX(1994,ZX:1994:4:0:0:0:::0:SA) =
  [1994]:Integerzahl(1994).NUM

lemmatisiere_ZX(09,ZX:09:2:0:0:0:::0:SA) =
  [09]:ziffern(09).NUM
```

lemmatisiere_SX

Input: SX-Form als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Lemmatisierung von Zeichen. Das Zeichen oder die Zeichenkette wird, wenn sie keinen speziellen Namen hat, als *SX(<zeichen>).CC* lemmatisiert.

```
lemmatisiere_SX(?,SX:?:1:0:0:0:::0:SA) = [?]:SX(?).CC
```

lemmatisere_ZSX

Input: ZSX-Form als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Überprüft die verschiedenen Möglichkeiten für Zahlkonstruktionen:

```
if (/^([0-9]+)[,.]([0-9]+)$/) { &check_dezimalzahl;}
if (/\.$/) { &check_ordinalzahl;}
if (/\..+$/) { &check_date;}
if (/[\+-]/){
    &check_signzahl);
    &check_differenz;
}
if (/:/){
    &check_verhaeltniszahl;
    &check_gradzahl;
}
if (/\//){
    &check_bruchzahl;
    &check_jahreszahl;
    &check_ordsucc;
    &check_trenner;
    &check_date;
}
if (/%/){ &check_prozentzahl;}
if (/\'/){ &check_apojahr;}
```

Die Ergebnisse der einzelnen Tests werden in der Ergebnisliste gesammelt. Konnte keine Lemmatisierung gefunden werden, so wird noch überprüft, ob es sich um eine Zahl gefolgt von einem Zeichen, das im Lexikon der Einheiten

enthalten ist (z.B *$*), handelt. Trifft auch das nicht zu, wird das UK-Lemma als Ergebnis zurückgegeben.

```
lemmatisiere_ZSX(0,3,ZSX:0,3:3:0:0:0:::0:SA) =
  [0,3]:Dezimalzahl(0,3).NUM

lemmatisiere_ZSX(11.,ZSX:11.:3:0:0:0:::0:SA) =
  [11.]:Ord(Integerzahl(11)).ORD

lemmatisiere_ZSX(1/2,ZSX:1/2:3:0:0:0:::0:SA) =
  [1/2]:Bruchzahl(1,2).NUM
  [1/2]:Folge(Integerzahl(1),Integerzahl(2)).NUM

lemmatisiere_ZSX(93/94,ZSX:93/94:5:0:0:0:::0:SA) =
  [93/94]:Bruchzahl(93,94).NUM
  [93/94]:Jahreszahlfolge(93,94).NUM
  [93/94]:Folge(Integerzahl(93),Integerzahl(94)).NUM

lemmatisiere_ZSX(24.12.1994,ZSX:24.12.1994:10:0:0:0:::0:SA)
 = [24.12.1994]:Datum(24,12,1994).NUM

lemmatisiere_ZSX(5:0,ZSX:5:0:0:0:0:::0:SA) =
  [5:0]:Verhältnis(Integerzahl(5),Integerzahl(0)).NUM

lemmatisiere_ZSX(+12,8,ZSX:+12,8:5:0:0:0:::0:SA) =
  [+12,8]:signDezimalzahl(12,8).NUM

lemmatisiere_ZSX(+21,9%,ZSX:+21,9%:6:0:0:0:::0:SA) =
  [+21,9%]:Prozentzahl(signDezimalzahl(21,9)).NUM
```

lemmatisiere_BZX

Input: BZX-Form als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Lemmatisierung der Mischformen aus Ziffern und Buchstaben. Zuerst wird mit **look_up_en, look_up_ak** und **look_up_akEN** die Form im Eigennamen-lexikon oder im Abkürzungslexikon nachgeschlagen, bzw. wird versucht, die Form als Kompositum aus Eigenname und Abkürzung zu analysieren. Dann werden 3 Fälle unterschieden:

```
if (/^([0-9]+)($typ_buchstabe+)$/) # Suffix

elsif (/^($typ_buchstabe+)([0-9]+)$/) # Präfix

elsif (/^($typ_buchstabe+)([0-9]+) ($typ_buchstabe+)$/) #
Zirkumfix
```

In diesen Fällen wird jeweils versucht, den Buchstabenanteil als Affix im Lexikon der numerischen Ausdrücke nachzuschlagen (vgl. dazu auch die Beispiele

in Anhang D auf Seite 206). Die Form wird dann als Kompositum aus dem Affix und einer Konstituente vom Typ *NUM* lemmatisiert. Konnte im ersten Fall auf diese Weise keine Lemmatisierung gefunden werden, wird zusätzlich der Buchstaben-Anteil im numerischen Lexikon als Einheit nachgeschlagen. Zurückgegeben wird die Liste der möglichen Lemmatisierungen, wenn keine möglich war, das UK-Lemma.

```
lemmatisiere_BZX(60er,BZX:60er:4:0:0:0:::0:SA) =
[60er]:60||er.mask(ZNS2,ZNP1):neM:deM:aeM:nmM:gmM:amM##NUM,N
```

```
lemmatisiere_BZX(100jährig,BZX:100jährig:9:0:0:0:::0:SA) =
  [100jährig]:100||jährig.ZA0:up##NUM,A
```

lemmatisiere_BSZX

Input: BSZX-Form als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Lemmatisierung von Mischformen aus Zahlen, Ziffern und Buchstaben. Zuerst erfolgt mittels **look_up_en** und **look_up_ak** das Nachschlagen im Eigennamen- und Abkürzungslexikon. Danach wird versucht, die Form kompositionell zu analysieren. Folgende Fälle werden unterschieden:

```
  if (/^([0-9]+)([^-\.0-9]+)$/) # Suffix vom Typ %ig
```

Look_up des Suffix im numerischen Lexikon

```
  if (/^([0-9]+)-([^\.0-9]+)$/) # Suffix nach Bindestrich
```

Look_up des Suffix im numerischen Lexikon

```
  if (/^([0-9]+\.)($typ_buchstabe+)$/)
```

Look_up eines Ordinalzahlsuffix im numerischen Lexikon und Look_up des Buchstabenanteils im EN- und AK-Lexikon.

```
  if (/-/) { &lemmatisiere_BSX_BW; }

  if (/\// && !/[-\']/) { &lemmatisiere_BSX_Slash;}

  if (/.[+&]./ && !/[-\'\/]/){ &lemmatisiere_BSX_Plus;}

  if (/^($typ_buchstabe+)(\'[0-9]+)$/)
```

Look_up des Buchstabenanteils im EN-Lexikon.

Zurückgegeben wird die Liste der möglichen Lemmatisierungen, falls keine Lemmatisierung möglich war, das UK-Lemma.

```
lemmatisiere_BSZX(100%ig,BSZX:100%ig:6:0:0:0:::0:SA) =
  [100%ig]:100||%ig.ZA0:up
```

```
lemmatisiere_BSZX(5.kläßler,BSZX:5.kläßler:9:0:0:0:::0:SA)=
[5.kläßler]:5.||kläßler.mask(ONS2,ONP1):neM:deM:aeM:nmM:gmM:
amM##ORD,N

lemmatisiere_BSZX(audi-100-modelle,BSZX:Audi-100-
Modelle:16:1:0:0:1:0:1:0:SA) =
  [audi-100-modelle]:audi-100-modell.
  neut(NS2,NP2):nmN:gmN:amN###EN-NUM-N

lemmatisiere_BZX(u4,BZX:u4:2:::::::1) =
  folge(u4,u5).AK:X###AK/AK
```

lemmatisiere_BSX

Input: BSX-Form als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Zuerst wird durch **look_up_en** überprüft, ob es sich um einen Eigennamen handelt. Dann werden in Abhängigkeit von den enthaltenen Zeichen spezielle **lemmatisiere**-Funktionen aufgerufen:

```
  if (/\./){
      &look_up_ak;
      &look_up_kompAk;
  }
  if (/-/){ &lemmatisiere_BSX_BW; }
  if (/\'/ && !/-/){ &lemmatisiere_BSX_AP; }
  if (/\// && !/[-\']/){ &lemmatisiere_BSX_Slash;}
  if ( /.[+&]./ && !/[-\'\/]/){ &lemmatisiere_BSX_Plus;}
```

Falls es sich um wortinterne Klammerung handelt, wird die Form ohne den geklammerten Anteil und die ganze Form ohne Klammerzeichen nochmals typisiert, mit WS-Vektor versehen und an **lemmatisiere** übergeben. Konnte kein Lemma gefunden werden, wird als Ergebnis das UK-Lemma zurückgegeben.

```
lemmatisiere_BSX(u.,BSX:u.:2:0:0:0:::1:SA) = [u.]:u..AK:X

lemmatisiere_BSX(verlagsverz.,BSX:Verlagsverz.:12:1:0:0:::0
:SA)=
  [verlagsverz.]:ver||lags||ver||z..AK:X##PREF,N,PREF,AK
  [verlagsverz.]:ver||lags||Verz..AK:X##PREF,N,AK

lemmatisiere_BSX(cdu/csu,BSX:CDU/CSU:7:1:0:0:::0:SA) =
  [cdu/csu]:folge(cdu,csu).AK:X###AK/AK

lemmatisiere_BSX(verkaufs-gmbh&co,BSX:Verkaufs-
GmbH&Co:16:1:0:0:1:0:1:0:SA)=
  [verkaufs-gmbh&co]:verkaufs-pfolge(gmbh,co).AK:X###N-AK&AK
```

lemmatisiere_BSX_AP

Input: Apostroph-Form als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Lemmatisierung der Apostrophformen. Es wird nach den verschiedenen Stellungen des Apostroph unterschieden:

```
if (/^\'(.*)$/) { &look_up_LApostroph;}

elsif (/^(.*)\'$/) { &look_up_RApostroph;}

else { &look_up_MApostroph; }
```

look_up_LApostroph und **look_up_RApostroph** erhalten zur Lemmatisierung den String ohne Apostroph, **look_up_MApostroph** bekommt den gesamten String mit Apostroph zur Lemmatisierung übergeben. Die look_up-Routinen geben die Liste der möglichen Lemmatisierungen zurück.

```
lemmatisiere_BSX(barclay's,BSX:Barclay's:9:1:0:0::::0:SA) =
  [barclay's]:barclay.EN:g
```

lemmatisiere_BSX_Slash, lemmatisiere_BSX_Plus

Input: /-, +-, oder &-Form als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Die Slash-Formen und Formen, die mit + oder & verbunden sind, werden in ihre Komponenten zerlegt. Die Komponenten werden mit **ws_vektor** typisiert und dann einzeln an **lemmatisiere** übergeben. Die Ergebnisse der Lemmatisierung der Komponenten (möglicherweise auch UK-Lemmata) werden als *slashfolge* bzw. *pfolge* in der Liste der möglichen Lemmatisierungen gesammelt, die dann zurückgegeben wird.

```
lemmatisiere_BSX(c&a,BSX:C&A:3:1:0:0::::0:SA) =
  [c&a]:pfolge(c,a).AK:X###AK&AK
```

lemmatisiere_BSX_BW

Input: Bindestrich-Form als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Die Bindestrichformen werden unterschieden nach Bindestrichwörtern und Koordinationsformen:

```
if (/^([^-].*)-$/) { &lemmatisiere_RKoord; }
```

```
  elsif(/^-(.*[^-])$/) { &lemmatisiere_LKoord; }

  elsif (/^[^-].*[^-]$/) { &lemmatisiere_BWort; }
```

Als Spezialfall werden folgende Formen behandelt:

```
elsif(/^\((.+)-\)(.+)$/)
```

In diesem Fall wird der geklammerte Buchstabenbestandteil als Präfix und der zweite Buchstabenbestandteil als einfache Form nachgeschlagen und wie ein Bindestrichwort lemmatisiert. Zurückgegeben wird die Liste der möglichen Lemmatisierungen.

lemmatisiere_BSX_BWort

Input: Bindestrichwort als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Die einzelnen Komponenten des Bindestrichworts werden mit **ws_vektor** neu typisiert und dann lemmatisiert. Die letzte Komponente wird an **lemmatisiere_bwHead** übergeben, die Vorderglieder an **lemmatisiere_bwleft.** Die Kategorien der einzelnen Komponenten werden bestimmt, und die Struktur wird als Zusatzinformation der Lemmakategorie, die durch **lemmatisiere_bwHead** bestimmt wurde, annotiert.

```
lemmatisiere_BSX(wein-soße,
  BSX:Wein-Soße:9:1:0:0:1:0:1:0:SA) =
  [wein-soße]:wein-soße.fem(NS0,NP4):neF:geF:deF:aeF###N+V-N
```

lemmatisiere_bwHead

Input: Endkomponente eines Bindestrichworts als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Die Form wird zusammen mit dem WS-Vektor der Lemmatisierungsroutine des entsprechenden Typs übergeben. Bei Wortklassen, die spezielle Fugenformen bilden, müssen diese ausgeschlossen werden. Ebenso müssen Präfixe und Kategorien, die kein Hinterglied einer komplexen Form bilden können, ausgeschlossen werden.

```
lemmatisiere_bwHead(soße,WB:Wein-Soße:4:1:0:0:0:0:1:0:SA) =
   [soße]:soße.fem(NS0,NP4):neF:geF:deF:aeF
```

lemmatisiere_bwleft

Input: linke Komponente eines Bindestrichworts als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Die Form wird zusammen mit dem WS-Vektor der Lemmatisierungsroutine des entsprechenden Typs übergeben. Bei flektierenden Wortklassen dürfen nur Fugenformen zugelassen werden.

```
lemmatisiere_bwLeft(wein,WB:Wein-Soße:4:1:0:0:1:0:0:0:SA) =
  wein.mask(NS1,NP2):FF1 weinen.VSW1:FF
```

lemmatisiere_BSX_LKoord

Input: Koordinationsform als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Lemmatisierung von Formen mit linksseitigem Koordinationsbindestrich. Die Form wird ohne Bindestrich neu typisiert und zusammen mit dem neuen WS-Vektor an **lemmatisiere_bwHead** übergeben. Dann wird vom aktuellen Lemmatisierungsindex rückwärts nach einer komplexen Form gesucht. Deren Lemmatisierung und das Ergebnis von lemmatisiere_bwHead wird dann der Routine **rekonstruiere_LKoord** übergeben, die die Liste der Lemmatisierungen aller Rekonstruktionsmöglichkeiten zurückgibt. Diese Liste wird, falls sie nicht leer ist, als Ergebnis weitergegeben. Ist die Liste leer, wird die Lemmatisierung der Form ohne Bindestrich (Ergebnis von **lemmatisiere_bwHead**) zurückgegeben.

```
lemmatisiere_BSX_LKoord(-vereinigungen,BSX:
-vereinigungen:14:0:0:0::::0:SA) =
  [-vereinigungen]:vereinigung.fem(NS0,NP3):nmF:gmF:dmF:amF
```

lemmatisiere_BSX_RKoord

Input: Koordinationsform als String, WS-Vektor als Liste

Output: Liste der möglichen Lemmatisierungen

Lemmatisierung von Formen mit rechtsseitigem Bindestrich. Die Form wird ohne Bindestrich typisiert und zusammen mit dem neu erstellten WS-Vektor an **lemmatisiere_bwleft** übergeben. Das Ergebnis wird als Liste der möglichen Lemmatisierungen zurückgegeben. Zusätzlich wird das Ergebnis an den Anfang der Liste @erg_delay_RKoord geschrieben, die abgearbeitet wird, sobald im Verlauf der weiteren Lemmatisierung des Satzes eine komplexe Form auftaucht.

```
lemmatisiere_BSX(ost-west-,BSX:Ost-West-:9:1:0:0::::0:SA) =
  [ost-west-]:ost-west-.ADV###N+ADV-ADV
```

Abbildung 7. Lemmatisierung der B-Formen

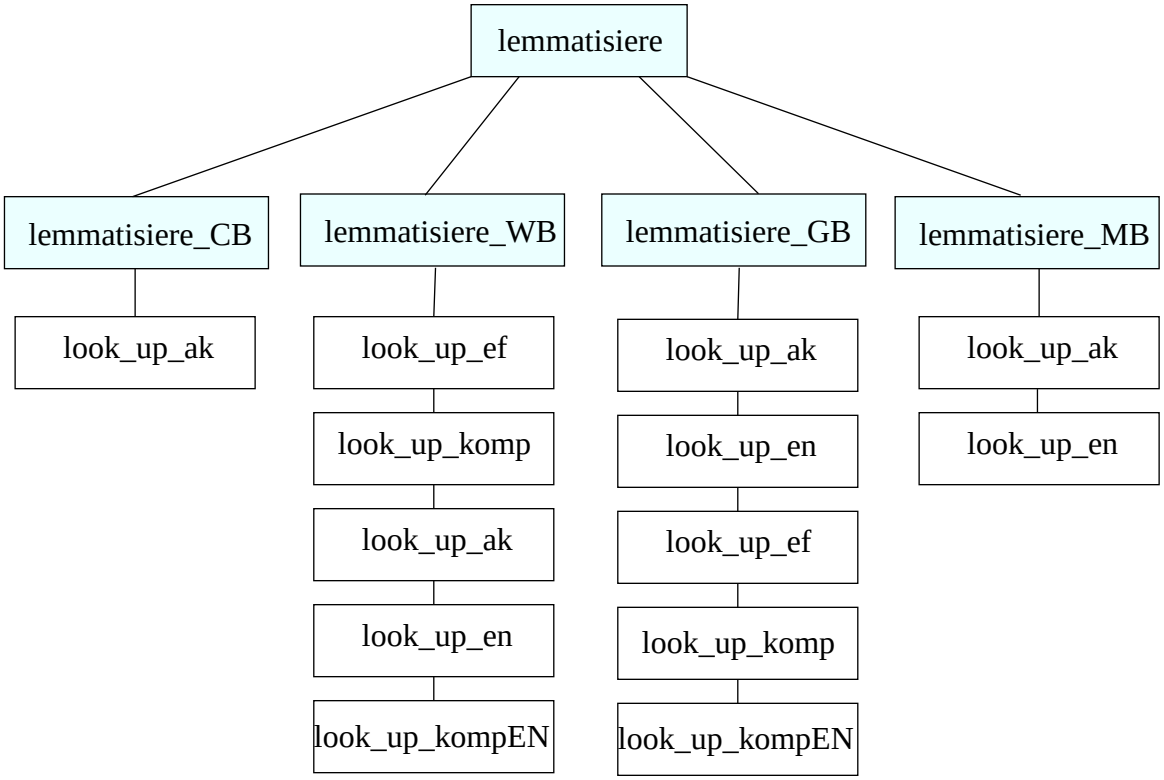
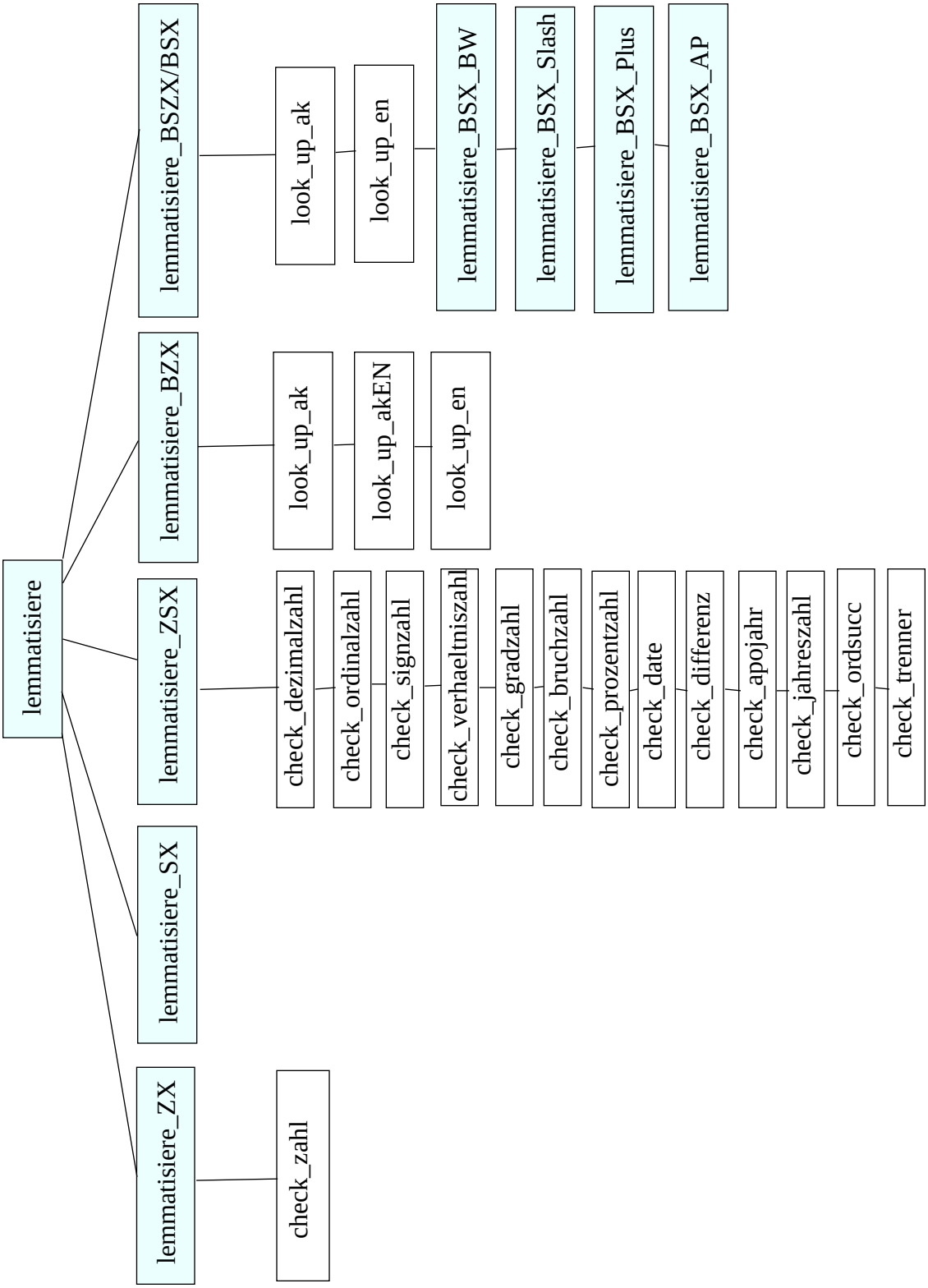


Abbildung 8. Lemmatisierung der X-Formen



4.5.2 Die **look_up**-Funktionen

look_up_ef

Input: Wort als String

Output: Liste der möglichen lexikalischen Analysen

Führt den Look-up im CISLEX-EF durch. Jeder Eintrag, der zum Eingabewort im EF-Lexikon gefunden wurde, wird an **check_ef** übergeben, wo überprüft wird, ob es sich um eine freie einfache Form (im Gegensatz zu Fugenformen, die gebunden sind) handelt. Zusätzlich werden die lexikalischen Regeln zur Überprüfung auf zu-Infinitiv von Verben mit abtrennbarer Partikel (**check_zu-Inf**), auf nominalisierten Verbinfinitiv (**check_nomInf**) und dessen genitivische Form (**check_GenInf**), auf nominalisierte Adjektive (**check_nomAdj**) und auf hochproduktive Derivationen (**check_deriv**) aufgerufen. Als Ergebnis wird die Liste aller möglichen lexikalischen Analysen (das sind sowohl die gefundenen Lexikoneinträge, als auch die durch Regeln abgeleiteten Analysen) zurückgegeben.

```
look_up_ef(spätzle) =
  spätzle.neut(NS2,NP0):neN:deN:aeN:nmN:gmN:dmN:amN

look_up_ef(langweiligste) =
langweilig.ADJ0:
  neFxs:nmUxs:aeFxs:amUxs:neUys:aeFys:aeNys:neFzs:aeFzs
langweiligste.NA:
  neFx:nmUx:aeFx:amUx:neUy:aeFy:aeNy:neFz:aeFz
```

look_up_ak

Input: Form als String

Output: Liste der möglichen lexikalischen Analysen

Führt den Look_up im CISLEX-AK durch. Eventuelle Punkte in der Eingabeform werden vor dem Look-up gelöscht. Falls das letzte Zeichen der Eingabeform *s* ist, wird die Form an **check_GenAk** übergeben, wo überprüft wird, ob es sich um eine Abkürzung mit angefügtem Genitiv- oder Plural-s handelt. Falls die Filteroption gewählt ist, werden die lexikalischen Analysen noch durch **filter_aehnlich** auf ihre Übereinstimmung mit der Originalorthographie überprüft. Die Analysen, die die wenigsten Abweichungen aufweisen, werden dann als Ergebnis zurückgegeben. Ansonsten wird die gesamte Menge der Analysen zurückgegeben.

```
look_up_ak(im) = IM.AK:X Im..AK:X im.AK:X im..AK:X (ohne
Filter)

look_up_ak(im) = IM.AK:X (mit Filter, Eingabe war “IM”)
```

look_up_en

Input: Form als String

Output: Liste der möglichen lexikalischen Analysen

Das Eingabewort wird im CISLEX-EN nachgeschlagen. Zusätzlich wird überprüft, ob es sich um einen Eigennamen mit Genitiv- bzw. Plural-s handelt (**check_GenEn**), oder ob es sich um eine Derivation von Eigennamen handelt (**check_DerivEn**). Ist die Filteroption gewählt, werden die gefundenen Analysen mit **filter_aehnlich** auf ihre Übereinstimmung mit der Originalorthographie hin überprüft. Als Ergebnis werden die Analysen mit den geringsten Abweichungen zurückgegeben. Ohne Filteroption werden alle Analysen als Ergebnis zurückgegeben.

```
look_up_en(maier) = maier.EN:X

look_up_en(maiersch) = maiersch.END:up
```

look_up_komp

Input: Form als String

Output: Liste der möglichen Segmentierungen mit lexikalischer Analyse des Hinterglieds

Versucht den Eingabestring als Kompositum zu identifizieren. Dazu werden alle möglichen Endstrings des Eingabestrings an **look_up_khead** übergeben, wo der Endstring im EF-Lexikon nachgeschlagen wird und auf die Fähigkeit, den Kopf eines Kompositums zu bilden, überprüft wird. Als Ergebnis liefert **look_up_khead** die Liste der möglichen lexikalischen Endgliedanalysen. Ist mindestens eine Analyse des Endstrings möglich, wird in **look_up_komp_left** versucht, den verbliebenen Anfangsstring als Folge von Vordergliedern zu analysieren. An dieser Stelle wird auch überprüft, ob an der Trennstelle zwischen Endstring und Anfangsstring Konsonantentilgung bei dreifach Konsonanz auftrat. Das bedeutet, wenn der Endstring mit der Folge Konsonant-Vokal beginnt, und der Anfangsstring mit demselben Konsonanten endet, wird der End-Konsonant des Anfangsstrings verdoppelt und auch dieser String zur weiteren Segemntierung an **look_up_komp_left** übergeben. Das Ergebnis von **look_up_komp_left** ist die Liste der möglichen Segmentierungen des Anfangsstrings. Wenn der Anfangsstring *zu* enthält, wird er zusätzlich in **check_kompZuInf** auf komplexen zu-Infinitiv hin überprüft. Ist auch eine Analyse als zu-Infinitiv möglich, wird diese Segmentierung (dann ohne *zu*) der Liste der Segmentierungen angefügt. Abschließend wird mit **look_up_numwort** überprüft, ob es sich um ein Zahlwort handelt. Zahlwörter werden zusätzlich zu ihrer kategorialen Analyse auf ihre numerische Darstellung reduziert. **look_up_komp_left** gibt zusätzlich zur Liste der Segmentierungen auch noch

die kategoriale Struktur Anfangsteils zurück. Diese kategoriale Struktur zusammen mit der Kategorie des Endglieds wird der Gesamtanalyse angefügt. Das Ergebnis von **look_up_komp** ist die Liste aller Segmentierungen mit lexikalischer Analyse des Hinterglieds und kategorialer Struktur.

```
look_up_komp(zucchinicremesuppe) =
  zucchini||creme||suppe.fem(NS0,NP4):neF:geF:deF:aeF
   ##N,N,N

look_up_komp(vierundvierzig) =
  vier||und||vierzig.ADJ##ADJ,KONJ,ADJ
  44.NUM

look_up_komp(stoffetzen) =
  stoff||fetzen.mask(NS2,NP0):neM:deM:aeM:nmM:gmM:dmM:amM
  ##N,N
  stoff||fetzen.VSW3:OI:1mGi:3mGi:1mGc:3mGc##N,V
  stoff||fetzen.neut(NS2,NPSG):neN:deN:aeN##N,N
```

look_up_khead

Input: Endstring einer komplexen Form

Output: Liste der möglichen lexikalischen Analysen

Die Routine überprüft, ob der Eingabestring mittels **look_up_ef** oder **look_up_suff** im EF-Lexikon oder als Suffix im Affixlexikon gefunden werden kann. Mit **check_khead** werden aus den gefundenen Analysen die Fugenformen und die nicht-Head-Kategorien (alles außer Adjektiv, Nomen, Verb, Adverb und Suffix) ausgefiltert. Die Restmenge wird dann zurückgegeben.

```
look_up_khead(suppe) = suppe.fem(NS0,NP4):neF:geF:deF:aeF
```

look_up_komp_left

Input: Anfangsstring einer komplexen Form, Analyse(n) des Endstrings

Output: Liste der möglichen Segmentierungen mit lexikalischer Analyse der Komponenten

Es wird versucht, den Eingabestring als Kompositumsvorderglied oder als Folge von Vordergliedern zu identifizieren. Der Eingabestring wird direkt im EF- und Affixlexikon nachgeschlagen. Dann werden alle möglichen Segmentierungen gebildet (unter Berücksichtigung möglicher Tilgung bei Dreifachkonsonanz) und komponentenweise nachgeschlagen. Jede im Lexikon gefundene Komponente wird durch **check_left** auf die Möglichkeit als Vorderglied aufzutreten überprüft. Mögliche Vorderglieder sind Fugenformen der Nomen, Verben und Adjektive[1], sowie die nichtflektierenden Funktionswortarten Präposition, Adverb, Partikel, Verbpartikel, Präfixe und die nicht-klassifi-

zierbaren XINC. Das Ergebnis ist die Liste aller möglichen Segmentierungen als Vorderglied des Eingabestrings.

```
look_up_komp_left(zucchinicreme,|suppe!suppe.
  fem(NS0,NP4):neF:geF:deF:aeF) =
zucchini;N||creme;N+ADJ||suppe!suppe.
  fem(NS0,NP4):neF:geF:deF:aeF
```

look_up_kompEN

Input: komplexe Form als String

Output: Liste der möglichen Segmentierungen mit lexikalischer Analyse des Hinterglieds

Diese Funktion entspricht der Funktion **look_up_komp**, mit der Ausnahme, daß Eigennamen als Bestandteile des Kompositums zugelassen sind.

```
look_up_kompEN(bonnmörder) =
bonn||mörder.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM##EN,N
```

look_up_kompAK

Input: komplexe Form als String

Output: Liste der möglichen Segmentierungen mit lexikalischer Analyse des Hinterglieds

Diese Funktion entspricht der Funktion **look_up_komp**, mit der Ausnahme, daß als Hinterglieder nur Abkürzungen zugelassen sind. Anstatt **look_up_khead** wird die Funktion **look_up_kAKhead** aufgerufen, die nur im AK-Lexikon nachschlägt.

```
look_up_kompAK(verlagsverz) = verlags||Verz..AK:X##N,AK
```

look_up_pref, look_up_suff

Input: Form als String

Output: Liste der möglichen lexikalischen Analysen

Beide Routinen schlagen den Eingabestring im Affix-Lexikon (vgl. Anhang D auf Seite 206) nach. **look_up_pref** überprüft auf die Kategorie Präfix und **look_up_suff** gibt nur Einträge der Kategorie Suffix zurück.

```
look_up_pref(agrar) = agrar.PREF (z.B. Agrarpolitik)
```

1. Fugenformen von Suffixen dieser Kategorien können nur in nicht-Anfangspositionen auftreten.

```
look_up_suff(farben) = farben.ASUFF2:up (z.B. cremefarben)
```

look_up_LApostroph

Input: Restform nach Apostroph am Wortanfang als String

Output: Liste der möglichen lexikalischen Analysen

Es wird zuerst überprüft, ob der Eingabstring in der Liste der lexikalisierten Reduktionsformen nach Apostroph am Wortanfang (vgl. Anhang E auf Seite 209) verzeichnet ist. Ist dies nicht der Fall, wird versucht ein Anfangssegment abzuspalten, das in dieser Liste enthalten ist. Konnte ein solches Anfangssegment abgespalten werden, wird der Rest mit **look_up_ef** im EF-Lexikon nachgeschlagen.

```
look_up_LApostroph(nausgehen) =
  hinausgehen.neut(NS2,NPSG):neN:deN:aeN
  hinausgehen.VST1:OI:1mGi:3mGi:1mGc:3mGc
(Apostrophform war ’nausgehen)
```

look_up_RApostroph

Input: Restform vor Apostroph am Wortende als String

Output: Liste der möglichen lexikalischen Analysen

Falls der Eingabestring nicht auf *e* endet, wird in **check_RApo_e** versucht, die Eingabeform mit angehängtem *e* mittels **look_up_ef** und **look_up_komp** als einfache oder komplexe Form zu identifizieren. Die Eingabeform wird außerdem mit **look_up_ef** und **look_up_ef** im Eigennamen bzw. Abkürzungslexikon nachgeschlagen. Handelt es sich um einen nichtflektierten Eigennamen bzw. Abkürzung, wird als morphosyntaktisches Merkmal “g” für Genitiv gesetzt und die Analyse der Ergebnisliste angfügt. Die möglichen Analysen werden als Liste zurückgegeben.

```
look_up_RApostroph(charles) = charles.EN:g
(Apostrophform war Charles’)

look_up_RApostroph(hätt) = haben.VUNR:1eVc:3eVc
(Apostrophform war hätt’)
```

look_up_MApostroph

Input: Apostrophform als String

Output: Liste der möglichen lexikalischen Analysen

Die Formen mit Apostroph in der Wortmitte werden mit Apostroph übergeben. Die Form wird zuerst anhand der Liste der lexikalisierten Apostrophformen

(vgl. Anhang E auf Seite 209) überprüft. Der weitere Look-up richtet sich nach der Position des Apostrophs im Wort (nach dem ersten Buchstaben, vor dem letzten Buchstaben, in der Mitte).

```
  if (/^(.)\'(.[^\']+)$/) { # nach dem ersten Zeichen
      if (/^[dlo]/) { &look_up_en;}
      if (/^z$/){ &check_MApo_u; }
      &check_MApo_e;
      }
  elsif (/^([^\']+)\'(.)$/) { # vor dem letzten Zeichen
      &check_MApo_2w;
      &check_MApo_EN;
      &check_MApo_e;
  }
  elsif (/^([^\']+)\'([^\']+)$/) { # in der Mitte
      if (/^(dell|all/) {
         &look_up_en($rest);
      }
      &check_MApo_EN;
      &check_MApo_e;
  }
```

Die Funktionen **check_MApo_u** und **check_MApo_e** überprüfen, ob die Apostrophform als einfache oder komplexe Form analysiert werden kann, wenn anstelle des Apostrophs ein *u* bzw. ein *e* eingefügt wird. **check_MApo_2w** überprüft, ob es sich um zwei Wörter handelt und **check_MApo_EN** überpüft, ob es sich um einen Eigennamen oder einen Eigennamen mit Eigennamensuffix handelt. Zurückgegeben wird die Liste der möglichen Lemmatisierungen.

```
look_up_MApostroph(o'connor) = connor.EN:X

look_up_MApostroph(z'sammen) = zusammen.ADV

look_up_MApostroph(cd's) = CD.AK:m

look_up_MApostroph(wies'n) =
  wiesen.fem(NS0,NPSG):neF:geF:deF:aeF
```

```
look_up_MApostroph(blüh'n) =
  blühen.VSW1:OI:1mGi:3mGi:1mGc:3mGc
  blühen.neut(NS2,NPSG):neN:deN:aeN

look_up_MApostroph(auf's) = (auf+das).PREP4

look_up_MApostroph(reing'schaut) = reinschauen.VSWT1#4:OZ

look_up_MApostroph(beck'schem) = becksch.END:deMxp:deNxp
```

4.5.3 Lexikalische Regeln

check_zuInf

Input: Wortform als String

Output: Liste der möglichen lexikalischen Ableitungen

Testet ob es sich beim Eingabestring um einen Infinitiv oder ein Gerundiv mit *zu* handelt.[1] Falls der Eingabestring auf das Muster /^.+zu.+$/ paßt, wird die Form ohne *zu* im EF-Lexikon nachgeschlagen. Wenn es sich um eine Infinitiv- oder Partizip-Präsens-Form eines Verbs handelt, wird dieses Ergebnis zurückgegeben.

```
check_zuInf(aufzustehen) = aufstehen.VUNRT#3:OI

check_zuInf(auszutauschend) = austauschen.VSWT1#3:OE
```

check_kompZuInf

Input: Vorderglied einer komplexen Form, Hinterglied einer komplexen Form (jeweils als Strings), Liste der möglichen Lemmatisierungen des Hinterglieds

Output: Liste der möglichen Segmentierungen

Funktioniert ähnlich wie **check_zuInf**. Allerdings ist in diesem Fall das komplexe Verb nicht im Lexikon gefunden worden. Wenn die Lemmatisierungen des Hinterglieds ein Verb-Infinitiv- oder Partizip-Präsens-Lemma enthalten, und das Vorderglied mit *zu* endet, wird mit **look_up_komp_left** nach einer Segmentierung des Vorderglieds ohne *zu* gesucht. Zurückgegeben werden die möglichen Segmentierungen der Vorderglieder zusammen mit den entsprechenden Verblemmatas des Hinterglieds.

1. Die zu-Infinitive und zu-Gerundiva der Verben mit abtrennbarer Partikel des EF+-Lexikons sind nicht im Flex-Lexikon (also auch nicht in der EF-DBM), und müssen daher über eine Regel abgeleitet werden.

```
check_kompZuInf(still,halten,
     halt.mask(NS1,NP2):dmM
     halten.VST6:OI:1mGi:3mGi:1mGc:3mGc
     halten.neut(NS2,NPSG):neN:deN:aeN) =
  still||halten.VST6:OI##ADJ,V

check_kompZuInf(still,haltend,
     halten.VST6:OE haltend.ADJ0:up) =
  still||halten.VST6:OE##ADJ,V
```

check_nomInf

Input: Wortform als String

Output: Liste der möglichen lexikalischen Ableitungen

Wenn die Eingabeform als Verbinfinitiv im EF-Lexikon gefunden wird, so wird eine zusätzliche Analyse für diese Form mit der Lemmakategorie *neut(NS2, NPSG):neN:deN:aeN* erzeugt und als Ergebnis zurückgegeben.

```
check_nomInf(laufen) = laufen.neut(NS2,NPSG):neN:deN:aeN
```

check_genInf

Input: Wortform als String

Output: Liste der möglichen lexikalischen Ableitungen

Falls die Eingabeform mit *s* endet, wird überprüft, ob es sich um die Genitivform eines nominalisierten Infinitivs handelt, d.h. ob die Form ohne *s* im EF-Lexikon als Verbinfinitiv verzeichnet ist. In diesem Fall wird eine zusätzliche Analyse mit der Lemmakategorie *neut(NS2,NPSG):geN* erzeugt und als Ergebnis zurückgegeben.

```
check_genInf(laufens) = laufen.neut(NS2,NPSG):geN
```

check_nomAdj

Input: Wortform als String

Output: Liste der möglichen lexikalischen Ableitungen

Falls die Eingabeform mit /(^.+e)[mnrs]?$/ matcht und als attributive Adjektivform im EF-Lexikon steht, wird ein Lemma als nominalisiertes Adjektiv gebildet. Die Lemmaform ist der Stamm der Adjektivform (Positiv-, Komparativ- oder Superlativstamm) mit der Endung *e*. Als Lemmakategorie fungiert *NA* und die Merkmale entsprechen bis auf die Gradmerkmale, die nun fehlen, den Adjektivmerkmalen.

```
check_nomAdj(behinderter) =
  behinderte.NA:neMz:neMy:geFx:deFx:dmNx:dmMx:dmFx

check_nomAdj(höchste) =
  höchste.NA:neFx:nmUx:aeFx:amUx:neUy:aeFy:aeNy:neFz:aeFz
```

check_deriv

Input: Wortform als String

Output: Liste der möglichen lexikalischen Ableitungen

Prüft die Eingabewortform auf die produktiven nominalen Derivationssuffixe *chen, lein* und *ler*[1]. Kann die Form in eine Nomengrundform (bzw. bei Umlaut und den Suffixen *chen* und *lein* in den Pluralstamm eines Nomens) und eine Flexionsform der oben genannten Suffixe zerlegt werden, so wird ein Lemma mit der entsprechenden Nomenkategorie des Suffix und den entsprechenden morphosyntaktische Merkmalen gebildet und zurückgegeben.

```
check_deriv(produktler) =
  produktler.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM
```

check_MApo_2w

Input: Paar aus Form vor und nach dem Apostroph jeweils als String

Output: Liste der möglichen Analysen

Prüft, ob der erste Eingabestring als Verb, Präposition oder Konjunktion analysiert werden kann (**look_up_ef** und **look_up_komp**). Wenn dies nicht möglich ist, wird zusätzlich überprüft, ob sich der erste String nach Anfügen von *e* am Ende als Verb analysieren läßt. Der zweite String kann dann entwerder “s” oder “n” sein. “s” nach Präposition wird zu *das*, ansonsten zu *sie* oder *es*. “n” in allen Fällen zu *den*. Als Ergebnis werden die möglichen Paare aus Lemma der Anfangsform und Endform zurückgegeben.

```
check_MApo_2w(weil,s) = (weil+/es/sie).KONJ
```

4.5.4 Rekonstruktionsregeln

rekonstruiere_LKoord

Input: Liste der möglichen Lemmatisierungen der vorangehenden komplexen Form und Liste der Lemmatisierungen der Koordinations-Bindestrichform

1. Die gängigen Derivationen dieser Art sind bereits im EF-Lexikon erfaßt. Diese Regel wird nur dann aufgerufen, wenn es sich um eine Neubildung handelt.

Output: Liste der möglichen Lemmatisierungen der rekonstruierten Form

Aus der Liste der möglichen Lemmatisierungen einer vorangehenden komplexen Form werden diejenigen Analysen, die eine komplexe Form beschreiben, ausgewählt. Handelt es sich um ein Bindestrichwort, wird jede Endfolge von Komponenten in **vergl_typ** geprüft, ob sie mit der gekürzten Bindestrichform verträglich ist. Wenn dies der Fall ist, wird die entsprechende Anfangsfolge als Anfangsfolge der Tilgungsform gewählt. Handelt es sich bei der vorangehenden komplexen Form um ein Kompositum, wird mit den Segmentfolgen analog verfahren. Als Ergebnis werden die möglichen Rekonstruktionen mit den Lemmakategorien der Bindestrichform zurückgegeben.

```
Studienfächer und -inhalte:
WB Studienfächer:
  studien||fach.neut(NS1,NP14):nmN:gmN:amN##N,N
  studien||fächer.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM##N,N
WB und:
  und.KONJ
BSX -inhalte:
  studien||inhalt.mask(NS1,NP2):deM:nmM:gmM:amM##N,N
```

rekonstruiere_RKoord

Input: Koordinations-Bindestrichform als String, Liste der Lemmatisierungen der komplexen Form, Liste der Lemmatisierungen der vorangehenden Bindestrichform

Output: Liste der möglichen Lemmatisierungen der rekonstruierten Form

Diese Funktion wird von **delayed_rekonstruiere_RKoord** aus aufgerufen, das seinerseits in der Hauptschleife aufgerufen wird, sobald eine komplexe Form analysiert werden konnte, nachdem ein Eintrag in die @erg_delay_RKoord-Liste erfolgte. Die Funktion ist ansonsten weitgehend analog zum linksseitigen Fall, der in **rekonstruiere_LKoord** behandelt wird. Als Ergebnis wird die Liste der möglichen Rekonstruktionen mit den entsprechenden Analysen zurückgegeben. Konnte eine Rekonstruktion erfolgen, so wird das entsprechende Element in der @erg_delay_RKoord-Liste gelöscht. Die anderen @erg-Listen werden am Index der Bindestrichform gemäß des Ergebnisses der Rekonstruktion aktualisiert.

```
Ost-West- und Nord-Süd-Verbindungslinien:
BSX Ost-West-:
  ost-west-verbindungs||linie.
  fem(NS0,NP4):nmF:gmF:dmF:amF###N+ADV-N+ADV-N,N
WB und:
  und.KONJ
BSX Nord-Süd-Verbindungslinien:
  nord-süd-verbindungs||linie.
  fem(NS0,NP4):nmF:gmF:dmF:amF##N,N###N+ADV-N+ADV-N,N
```

4.6 Disambiguierung

4.6.1 Wortbezogene Filter

Die wortbezogenen Filter werden - immer unter der Voraussetzung, daß die Disambiguierungsoption gewählt wurde - zum Teil direkt in den Look-up-Routinen, zum Teil unmittelbar nach den Look-up-Routinen aufgerufen.

filter_gross

Input: Liste der möglichen Analysen

Output: Liste der gefilterten Analysen

Mit Ausnahme von Anreden werden nur nominale Analysen zugelassen. Falls keine Analyse übrigbleibt, wird die Eingabeliste wieder zurückgegeben. Anreden sind Pronomen in der zweiten Person, außerdem die Possessivpronomen *dein, ihr, euer, eur* und deren Flexionsformen.

```
WB Behinderten:
  *behindern.VSW4s:1mVi:3mVi:1mVc:3mVc
  *behindert.ADJ0:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp
  :geUzp:deUyp:geUyp:aeMzp:aeMyp:amUzp:amUyp
  behinderte.NA:geMx:geNx:gmUx:aeMx:nmUz:nmUy:deUz:geUz
  :deUy:geUy:aeMz:aeMy:amUz:amUy[1]
```

filter_klein

Input: Liste der möglichen Analysen

Output: Liste der gefilterten Analysen

Die nominalen Analysen und diejenigen Pronomen, die nur als Anrede gebraucht werden können, werden ausgeschlossen. Eine Ausnahme bilden die explizit aufgeführten Nomen, bei denen Kleinschreibung möglich ist (z.B. Tageszeiten oder häufig auftretende Verbpartikel). Falls keine Analyse übrigbleibt, wird die Eingabeliste unverändert zurückgegeben.

```
WB behinderten:
  behindern.VSW4s:1mVi:3mVi:1mVc:3mVc
  behindert.ADJ0:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp
  :geUzp:deUyp:geUyp:aeMzp:aeMyp:amUzp:amUyp
  *behinderte.NA:geMx:geNx:gmUx:aeMx:nmUz:nmUy:deUz:geUz
  :deUy:geUy:aeMz:aeMy:amUz:amUy
```

1. Die mit * markierten Analysen werden jeweils ausgeschlossen.

filter_ef

Input: Liste der möglichen Analysen als einfache Form, Boolescher Wert als Merkmal für Groß- (= 1) bzw. Kleinschreibung (= 0) im Text

Output: Liste der gefilterten Analysen

Bei Kleinschreibung wird bei konkurrierenden Analysen als prädikatives Adjektiv und als infinite Verbform die Analyse als Verbform präferiert und zurückgegeben. Bei Großschreibung werden Analysen als Nomen gegenüber den Analysen als nominalisiertes Adjektiv präferiert und zurückgegeben.

```
WB Alter:
  alter.neut(NS2,NP1):neN:deN:aeN:nmN:gmN:amN
  *alte.NA:neMz:neMy:geFx:deFx:dmUx
```

filter_seg

Input: Liste der möglichen Segmentierungen mit Kategorien, Boolescher Wert als Merkmal für Groß- (= 1) bzw. Kleinschreibung (= 0) im Text

Output: Liste der gefilterten Analysen

Zusätzlich zu den Goß-/Kleinfiltern werden die Ergebnisse der Segmentierung in **look_up_komp** mit diesem Filter bearbeitet. Zuerst wird **filter_seg_anzahl** aufgerufen, der nur die Segmentierungen mit der geringsten Anzahl an Segmenten zuläßt. Dessen Ergebnis wird (bei Großschreibung) an **filter_seg_NN** übergeben, der die Segmentierungen mit der größten Anzahl an nominalen Komponenten auswählt. Dieses Ergebnis wird schließlich mit **filter_seg_head** auf spezielle Alternativenpaare mit Präferenz eines Endglieds überprüft. Ähnliche Funktion hat auch die Funktion **filter_seg_ENDE**, die für den Spezialfall zuständig ist, daß eine Form sowohl mit einem Hinterglied als nominalisiertes Adjektiv einer Partizip-Präsens-Form (Partizip endet auf *end*) als auch mit einem Hinterglied, das eine Flexionsform von *Ende* ist, analysiert wurde. In diesem Fall wird die Analyse als nominalisiertes Adjektiv ausgewählt. Als Ergebnis wird die Liste der Analysen, die nach Anwendung der verschiedenen Filter übriggeblieben sind, zurückgegeben.[1]

1. Die einzelnen Filter werden nur aufgerufen, wenn tatsächlich noch Alternativen vorhanden sind. Außerdem liefert jeder Filter als Ergebnis die Eingabeliste zurück, falls durch den Filter alle Analysen ausgeschlossen würden. So ist gewährleistet, daß der Aufruf von **filter_seg** nie alle Analysen ausschließt.

```
WB Beisitzende:[1]
  bei||sitzende.NA:neFx:nmUx:aeFx:amUx:neUy:aeFy:aeNy
:neFz:aeFz##N+PREP,NA
  *beisitz||ende.neut(NS0,NP4):neN:geN:deN:aeN##N,N
  *bei||sitz||ende.neut(NS0,NP4):neN:geN:deN:aeN
##N+PREP,N+V,N

WB Alleinerziehende:
  allein||erziehende.NA:neFx:nmUx:aeFx:amUx:neUy:aeFy:aeNy
:neFz:aeFz##ADJ+PART+ADV,NA
  *al||lein||erziehende.NA:neFx:nmUx:aeFx:amUx:neUy:aeFy
:aeNy:neFz:aeFz##PREF,N+V,NA
  *all||ein||erziehende.NA:neFx:nmUx:aeFx:amUx:neUy:aeFy
:aeNy:neFz:aeFz##N,V,NA
  *alle||in||erziehende.NA:neFx:nmUx:aeFx:amUx:neUy:aeFy
:aeNy:neFz:aeFz##ADV,PREP,NA
  *all||einer||ziehende.NA:neFx:nmUx:aeFx:amUx:neUy:aeFy
:aeNy:neFz:aeFz##N,N,NA
  *allein||erzieh||ende.neut(NS0,NP4):neN:geN:deN:aeN
##ADJ+PART+ADV,V,N
  *al||lein||erzieh||ende.neut(NS0,NP4):neN:geN:deN:aeN
##PREF,N+V,V,N
  *all||ein||erzieh||ende.neut(NS0,NP4):neN:geN:deN:aeN
##N,V,V,N
  *alle||in||erzieh||ende.neut(NS0,NP4):neN:geN:deN:aeN
##ADV,PREP,V,N
  *all||einer||zieh||ende.neut(NS0,NP4):neN:geN:deN:aeN
##N,N,V,N
```

filter_seg_head

Input: Liste der möglichen Analysen

Output: Liste der gefilterten Analysen

Die Liste, die Paare von alternativen Hintergliedern mit eindeutiger Präferenz einer Analyse (eine Liste dieser Paare befindet sich in Anhang E auf Seite 209) beinhaltet, wird mit den Hintergliedanalysen der Eingabeliste verglichen und falls diese für jedes Element eines solchen Paares Analysen enthält, werden die Analysen des nicht-präferierten Elements ausgeschlossen.

```
WB Vereinsamt:
  *verein||samt.mask(NS1,NP2):neM:deM:aeM##N+V,N
  vereins||amt.neut(NS1,NP14):neN:deN:aeN##N,N
```

1. Die Analysen sind bereits durch die Groß-/Kleinfilter eingeschränkt.

```
WB Glückstag:
  glück||stag.neut(NS1,NP3):neN:deN:aeN##N+V,N
  *glücks||tag.mask(NS1,NP2):neM:deM:aeM##N,N

WB Kunststoffschi:
  kunst||stoff||schi.mask(NS2,NP8):neM:deM:aeM##N,N,N
  *kunst||stoffs||chi.neut(NS13,NP6):neN:geN:deN:aeN##N,N,N

WB Lieblingsoma:
  *liebling||soma.neut(NS0,NP10):neN:geN:deN:aeN##N,N
  lieblings||oma.fem(NS0,NP6):neF:geF:deF:aeF##N,N
```

filter_aehnlich

Input: Textform der analysierten Abkürzung bzw. Eigenname, Liste der möglichen Analysen

Output: Liste der gefilterten Analysen

Mit der Funktion **string_diff** wird die Anzahl der Abweichungen zwischen der Textform und der Lemmaform ermittelt. Die Analysen mit der geringsten Zahl an Abweichungen werden als Ergebnis zurückgegeben.

```
GB IM:
  IM.AK:X
  *Im..AK:X
  *im.AK:X
  *im..AK:X
```

4.6.2 Kontextbezogene Disambiguierung

Im Gegensatz zu den wortbezogenen Filtern findet die kontextbezogene Disambiguierung erst nach der Analyse des gesamten Satzes statt.

disambiguiere

Der Aufruf von **disambiguiere** (ohne Argumente) bewirkt, daß bei gewählter Disambiguierungsoption die beiden Funktionen **disambiguiere_WA** (Disambiguierung von Wortartambiguitäten) und **disambiguiere_L** (Disambiguierung mittels Kongruenzketten) aufgerufen werden. Die Funktionen benutzen die globalen Variablen in denen die Ergebnisse der Analysen der einzelnen Formen des aktuellen Satzes abgespeichert sind (@erg-Listen) und überschreiben diese gegebenenfalls mit den Ergebnissen der Disambiguierung.

disambiguiere_WA

Die @erg_ambig-Liste wird auf Indizes überprüft, an denen Wortartambiguitäten auftreten. Mit jedem Index, an dem solche Ambiguitäten auftreten, wird **disambiguiereWA_kontext** aufgerufen, das als Ergebnis die - soweit möglich

disambiguierten - Lemmata des entsprechenden Index zurückgeben. Mit diesem Ergebnis werden die @erg-Listen aktualisiert.

disambiguiereWA-kontext

Input: Index im Satz an dem Wortartambiguitäten auftreten

Output: Liste der disambiguierten Analysen

Je nachdem, ob der Index 0 ist, der vorletzte Index im Satz ist[1] oder in der Satzmitte liegt, werden verschiedene Funktionen benutzt. Am Satzanfang wird **pruefe_satzanfang_ausschluss** und **pruefe_satzanfang_alternativen** aufgerufen, in der Satzmitte **pruefe_satzmitte_alternativen** und am Satzende **pruefe_satzende_alternativen**.[2] Diese Funktionen geben jeweils die Liste der soweit möglich disambiguierten Analysen zurück.

```
Beide kommen ...:
WB Beide:
  *beid.DET2:amN:nmN
  beid.Pron:amN:nmN
WB kommen:
  kommen.VST1:OI:1mGi:3mGi:1mGc:3mGc

Zu handeln...:
WB Zu:
  *zu.PART
  zu.KONJ
  *zu.PREP3
WB handeln:
  handeln.VSW6:OI:1mGi:3mGi:1mGc:3mGc

Schnelle schreibt...:
WB Schnelle:
  *schnell.ADJ0:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp
:neFzp:aeFzp
  *schnelle.fem(NS0,NP4):neF:geF:deF:aeF
  *schnellen.VSW1:1eGi:1eGc:3eGc
  *schnelle.NA:neFx:nmUx:aeFx:amUx:neUy:aeFy:aeNy:neFz:aeFz
  schnelle.EN:X
WB schreibt:
  schreiben.VST1:2mGi:3eGi
```

1. Die Liste der A-Formen eines Satzes enthält als letztes Element das Satzendezeichen (*.!?*). Deshalb bezeichnet der vorletzte Index das letzte Wort im Satz.

2. Ausschlußkontexte wurden bislang nur für den Satzanfang definiert.

disambiguiere_L

Ähnlich wie **disambiguiere_WA** wird nur überprüft, an welchen Indizes Ambiguitäten der Grundform oder morphologischen Klasse auftreten. (Bislang ist dieses Verfahren nur für Ambiguitäten bei Nomen implementiert.) Mit den entprechenden Indizes wird dann die Funktion **disambiguiere_LN** aufgerufen, die als Ergebnis die Liste der soweit möglich disambiguierten Analysen zurückgibt. Die @erg-Listen werden gemäß diesem Ergebnis aktualisiert.

disambiguiere_LN

Input: Index im Satz, an dem disambiguiert werden soll (im Moment nur bei ambigen Nomen)

Output: Liste der disambiguierten Analysen

Es wird überprüft, ob sich eine bis auf das Nomen eindeutige Kongruenzkette bilden läßt. Zu diesem Zweck werden zwei Fälle überprüft:

1. Die A-Form an der Position Index-1 ist ein Determinator oder die A-Form ist ein attributives Adjektiv und die A-Form bei Index-2 ist weder Adjektiv noch Determinator.
2. Die A-Form an der Position Index-1 ist ein attributives Adjektiv und die Positionen davor sind auch Adjektive oder Adjektive und Determinator.

Im 1. Fall werden die morphosyntaktischen Merkmale des Vorgängers (an der Stelle Index-1) und die möglichen Analysen des Nomens an **pruefe_kongruenz** übergeben, wo diejenigen Nomenanalysen ausgeschlossen werden, bei denen keine Kongruenz möglich ist. Im zweiten Fall wird vor dem Aufruf von **pruefe_kongruenz** die Kongruenz-Schnittmenge der Merkmale der Vorgängerformen[1] gebildet (das erledigt die Funktion **schneide_merkmale**). Als Ergebnis wird die Liste der disambiguierten Nomenanalysen zurückgegeben.

```
mit gebeugtem Gang
WB mit:
  mit.PREP3
WB gebeugtem:
  gebeugt.ADJ0:deMxp:deNxp
WB Gang:
  gang.mask(NS1,NP12):neM:deM:aeM
  *gang.fem(NS0,NP6):neF:geF:deF:aeF
```

pruefe_satzanfang_ausschluss

Input: Liste der Analysen der ersten A-Form im Satz

1. Dies sind alle vorangehenden attributiven Adjektive und, falls vor diesen ein Determinator steht, auch der Determinator.

Output: Liste der am Satzanfang möglichen Analysen

Die Funktion gibt nur die Analysen zurück, deren Kategorie nicht in der Liste der am Satzanfang ausgeschlossenen Kategorien auftaucht. Ausschlußkategorien sind momentan (sofern andere Analysen vorhanden sind): finites Verb, Partikel vor Verb, Determinator vor Verb und Präposition vor Verb.

pruefe_satzanfang_alternativen

Input: Liste der Analysen der ersten A-Form im Satz

Output: Liste der am Satzanfang möglichen Analysen

Die Alternativenmengen zusammen mit der präferierten Kategorie und gegebenfalls der nachfolgenden Kategorie sind in einem assoziativen Array kodiert. Falls eine der Alternativenmengen des assoziativen Arrays in der Menge der möglichen Analysen enthalten ist und die folgende A-Form die entsprechende Kategorie hat, werden die Analysen, deren Kategorie in der Alternativenmenge nicht präferiert werden, ausgeschlossen. Der Eintrag ("ADJ+EN+N>EN",V) beispielsweise bedeutet, wenn am Satzanfang die Kategorien ADJ, EN oder N möglich sind, so ist vor einem Verb die Kategorie EN zu präferieren.

Die Funktionen **pruefe_satzmitte_alternativen** und **pruefe_satzende_alternativen** funktionieren analog. Bei den Satzende-Alternativen wird im assoziativen Array, wenn gefordert, die Vorgängerkategorie angegeben. Bei den Satzmitte-Alternativen wird, wenn notwendig, sowohl die Vorgänger- als auch die Nachfolgerkategorie angegeben und überprüft.

pruefe_kongruenz

Input: als String konkatenierte Liste der Kongruenzmerkmale des Vorgängers (bzw. Kongruenz-Schnittmenge der Merkmale der Vorgängerkategorien), als String konkatenierte Liste der Nomenanalysen

Output: Liste der Nomenanalysen, bei denen Kongruenz möglich ist.

Jede Nomenanalyse wird in **pruefe_possKongruenz** überprüft, ob die Liste der Merkmale der Vorgänger mindestens ein Merkmalsbündel enthält, das mit den Merkmalen des Nomens kongruiert. Ist dies nicht der Fall, wird die entsprechende Nomenanalyse ausgeschlossen.

5 Schlußbemerkung

In dieser Arbeit wurde die Rolle elektronischer Wörterbücher bei der automatischen Lemmatisierung genauer untersucht. Die Ergebnisse zeigten, daß durch den Einsatz eines elektronischen Wörterbuchs viele Probleme, die bei der Lemmatisierung ohne entsprechende lexikalische Datenbasis entstehen, gelöst werden können. Die pessimistische Einschätzung von Willée (1993) (Willée veranschlagt ein Mannjahr an Nachkorrektur für ein Korpus mit 1 Mio. Wörtern bei einer regelbasierten Lemmatisierung) konnte damit widerlegt werden. Weiterhin wurde gezeigt, wie Sonderformen, die in anderen Systemen stark vernachlässigt wurden, im Rahmen der automatischen Lemmatisierung zu behandeln sind.

In Bezug auf das Lexikon wurden die Kriterien, denen elektronische Wörterbücher genügen müssen, herausgearbeitet. Am Beispiel der morphologischen Kodierung des DKL-EF und der Generierung des entsprechenden Vollformenlexikons DKL-FLEX wurde gezeigt, wie diese Anforderungen im Bereich der Flexionsmorphologie zu realisieren sind. Für die Lemmaauswahl und die Klassifikation wurden objektivierbare und für die morphologische Kodierung operationalisierbare Kriterien angegeben. Insgesamt stellt die morphologische Klassifikation mit ihrer deklarativen Formalisierung als Prologprogramm eine umfassende formale Beschreibung der deutschen Flexionsmorphologie dar, die es erlaubt jedes einfache Wort des Deutschen in geeigneter Weise zu erfassen. Mit über 700 morphologischen bzw. morphosyntaktischen Kategorien geht das Klassifikationsschema weit über die in Standardwerken beschriebenen Phänomene hinaus.

Aufbauend auf diesem Lexikon und verschiedenen weiteren Komponenten des CIS-LEX-Systems wurde ein Lemmatisierungsalgorithmus entwickelt, der ausgehend von verschiedenen Formtypen eine Identifikation von 98% der in Texten vorkommenden Formen ermöglicht. Es werden sowohl Wortformen als auch Sonder- und Mischformen in großer Breite analysiert und zum Teil mit Hilfe spezieller Lexika identifiziert. Die behandelten Formtypen umfassen sowohl Formen, die nur Buchstaben in den verschiedensten Schreibungen enthalten, als auch Sonderformen, die Mischungen aus Ziffern, Sonderzeichen und Buchstaben enthalten. Für die Analyse der Wortformen wurde ein Segmentierungsalgorithmus entwickelt, der die komplexen Formen in ihre Bestandteile, die im DKL-EF verzeichnet sind, zerlegt. Eine wichtige Rolle beim Lemmatisierungsalgorithmus spielt auch die Disambiguierung von Mehrfachanalysen. Hier werden sowohl Verfahren, die auf Wortebene operieren angewendet, als auch Verfahren, die zur Disambiguierung eine Kontextanalyse vornehmen.

Die Basiseinheiten des vorgestellten Lemmatisierungsalgorithmus sind Wörter bzw. Formen, die zwischen Blanks oder Satzzeichen auftreten. Es hat sich jedoch gezeigt, daß für eine zufriedenstellende Lemmatisierung in Anwendungen wie Information Retrieval oder automatische Indizierung eine Lemmatisierung die als Gegenstand lediglich Wörter hat, nicht ausreicht. Daher ist für die Zukunft das Verfahren dahingehend zu erweitern, daß auch Einheiten, die aus mehreren Formen bestehen, als ein

Lemma analysiert werden können. Dabei ist sowohl an transparente Bildungen zu denken, wie etwa Datumsangaben, Maßangaben, zusammengesetzte Namen im allgemeinen sowie Personenbezeichnungen mit Titel und diversen Namensbestandteilen, usw., als auch mehr oder weniger stark lexikalisierte Einheiten, wie Kollokationen, Mehrwortlexeme und Idiome. In diesem Bereich ist sowohl eine geeignete Repräsentation dieser Elemente im Lexikon als auch eine Erkennungsprozedur im Rahmen der Lemmatisierung notwendig. Als Formalismus zur Beschreibung bieten sich hierfür die bereits in Abschnitt 3.5.2 auf Seite 126 beschriebenen lokalen Grammatiken in Form endlicher Automaten an.

Anhang A: Beispiele aus dem CISLEX-EF

Determinatoren:

jed,.DET3
jederart,.DET5
jederlei,.DET5
jedwed,.DET3
jen,.DET3
kein,.DET2
keinerlei,.DET5
manch,.DET1

Adjektive:

faschistoid,.ADJ11
faselbar,.ADJ0
faselig,.ADJ0
faserig,.ADJ0
fashionabel,.ADJ2
fashionable,.ADJ15
fasrig,.ADJ0
fassungslos,.ADJ11
fastidiös,.ADJ11

Adjektivsuffixe:

deutig,.ASUFF0
ehig,.ASUFF0
eiig,.ASUFF0
einig,.ASUFF0
fällig,.ASUFF0
fältig,.ASUFF0
förmig,.ASUFF0
fühlig,.ASUFF0
fach,.ASUFF0
farben,.ASUFF2
feldig,.ASUFF0
fellig,.ASUFF0

Nomen:

Borretsch;mask;NS1;NP6
Borschtsch;mask;NS0;NPSG
Borst;mask;NS1;NP12
Borst;mask;NS1;NP2
Borste;fem;NS0;NP4
Borstigkeit;fem;NS0;NP3
Borstwisch;mask;NS1;NP2
Borte;fem;NS0;NP4
Borusse;mask;NS4;NP4

Verben:

locken,.VSW1
lockern,.VSW4
lodern,.VSW4
logarithmieren,.VSW1s
loggen,.VSW1
logieren,.VSW1s
lohen,.VSW1
lohnen,.VSW1
kennen,.VUNR
mögen,.VUNR
leiden,.VST4#<leid,leid,litt,litt,litt>
leihen,.VST1#<leih,leih,lieh,lieh,lieh>
lesen,.VST3#<les,lies,las,läs,les>
liegen,.VST1#<lieg,lieg,lag,läg,leg>

Konjunktionen:

insofern,.KONJ
insoweit,.KONJ
inwiefern,.KONJ
inwieweit,.KONJ
je,.KONJ

Partikeln:

gar,.PART
garnicht,.PART
genau,.PART
genausowenig,.PART
gerade,.PART
geradezu,.PART
geschweige,.PART
gleich,.PART
gleichfalls,.PART
höchst,.PART

Präpositionen:

entgegen,.PREP5
entlang,.PREP4
entsprechend,.PREP3
exklusive,.PREP2
für,.PREP1

fern,.PREP2
gegenüber,.PREP6
gegen,.PREP1
gemäß,.PREP3
gen,.PREP1

Pronomen:

es,.Pron
etwas,.Pron
euch,.Pron
eueresgleich,.Pron
euresgleich,.Pron
ich,.Pron
ihr,.Pron
ihrergleich,.Pron
ihresgleich,.Pron
irgendetwas,.Pron
irgendw,.Pron
jedermann,.Pron
jemand,.Pron
kein,.Pron

Verbpartikeln:

abhanden,.VPART
drein,.VPART
inne,VPART
ran,.VPART
rein,.VPART
hintan,.VPART

Adverben:

vorher,.ADV
vorhin,.ADV
vorhinein,.ADV
vorkommendenfalls,.ADV
vorlings,.ADV
vormals,.ADV
vormittags,.ADV
vornüber,.ADV
vorn,.ADV
vornan,.ADV
vorne,.ADV
vornehmlich,.ADV

Interjektionen:

dalli,.INTJ
danke,.INTJ
dawai,.INTJ
denkste,.INTJ
dideldum,.INTJ
dideldumdei,.INTJ

Anhang B: Beispiele aus dem CISLEX-FLEX

Determinatoren:

%ihr,ihr.DET2:neN:aeN:neM
%welch,welch.DET5:X
%keiner,kein.DET2:gmM:geF:deF
%jedweden,jedwed.DET3:dmN:aeM
%diesem,dies.DET3:deM:deN
%ebendiejenige,ebend_jenig.DET9:aeF:neF
%irgendeine,irgendein.DET2:amM:nmM:aeF:neF
%jene,jen.DET3:nmN:amN:aeF:neF
%unsre,unsr.DET2:amM:nmM:aeF:neF
%jedem,jed.DET3:deM:deN

Pronomen:

%unsereinem,unserein.Pron:d3eU
%keiner,kein.Pron:n3eM:d3eF:g3eF
%ihn,er.Pron:a3eM
%dir,du.Pron:d2eU
%ihrer,sie.Pron:g3eF:g2mU:g3mU
%ihnen,sie.Pron:d3mU
%wir,wir.Pron:n1mU
%nix,nix.Pron:n3eU:a3eU:d3eU:g3eU
%sonstwem,sonstw.Pron:d3eU
%sonstjemandes,sonstjemand.Pron:g3eU
%unsresgleichen,unsresgleich.Pron:X
%niemanden,niemand.Pron:a3eU

Adjektive:

%bauschig,bauschig.ADJ0:up
%bauschige,bauschig.ADJ0:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp
%bauschigem,bauschig.ADJ0:deMxp:deNxp
%bauschigen,bauschig.ADJ0:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp:geUzp:deUyp:geUyp:aeMzp:aeMyp:amUzp:amUyp
%bauschiger,bauschig.ADJ0:neMzp:neMyp:geFxp:deFxp:dmUxp:uk
%bauschigere,bauschig.ADJ0:neFxk:nmUxk:aeFxk:amUxk:neUyk:aeFyk:aeNyk:neFzk:aeFzk
%bauschigerem,bauschig.ADJ0:deMxk:deNxk
%bauschigeren,bauschig.ADJ0:geMxk:geNxk:gmUxk:aeMxk:nmUzk:nmUyk:deUzk:geUzk:deUyk:geUyk:aeMzk:aeMyk:amUzk:amUyk
%bauschigerer,bauschig.ADJ0:neMzk:neMyk:geFxk:deFxk:dmUxk
%bauschigeres,bauschig.ADJ0:neNzk:neNxk:aeNzk:aeNxk
%bauschiges,bauschig.ADJ0:neNzp:neNxp:aeNzp:aeNxp
%bauschigste,bauschig.ADJ0:neFxs:nmUxs:aeFxs:amUxs:neUys:aeFys:aeNys:neFzs:aeFzs
%bauschigstem,bauschig.ADJ0:deMxs:deNxs
%bauschigsten,bauschig.ADJ0:us:geMxs:geNxs:gmUxs:aeMxs:nmUzs:nmUys:deUzs

:geUzs:deUys:geUys:aeMzs:aeMys:amUzs:amUys
%bauschigster,bauschig.ADJ0:neMzs:neMys:geFxs:deFxs:dmUxs
%bauschigstes,bauschig.ADJ0:neNzs:neNxs:aeNzs:aeNxs
%fashionable,fashionable.ADJ15:up:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp :neFzp:aeFzp
%fashionablem,fashionable.ADJ15:deMxp:deNxp
%fashionablen,fashionable.ADJ15:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp :geUzp:deUyp:geUyp:aeMzp:aeMyp:amUzp:amUyp
%fashionabler,fashionable.ADJ15:neMzp:neMyp:geFxp:deFxp:dmUxp:uk
%fashionablere,fashionable.ADJ15:neFxk:nmUxk:aeFxk:amUxk:neUyk:aeFyk:aeNyk :neFzk:aeFzk
%fashionablerem,fashionable.ADJ15:deMxk:deNxk
%fashionableren,fashionable.ADJ15:geMxk:geNxk:gmUxk:aeMxk:nmUzk:nmUyk :deUzk:geUzk:deUyk:geUyk:aeMzk:aeMyk:amUzk:amUyk
%fashionablerer,fashionable.ADJ15:neMzk:neMyk:geFxk:deFxk:dmUxk
%fashionableres,fashionable.ADJ15:neNzk:neNxk:aeNzk:aeNxk
%fashionables,fashionable.ADJ15:neNzp:neNxp:aeNzp:aeNxp
%fashionableste,fashionable.ADJ15:neFxs:nmUxs:aeFxs:amUxs:neUys:aeFys:aeNys :neFzs:aeFzs
%fashionablestem,fashionable.ADJ15:deMxs:deNxs
%fashionablesten,fashionable.ADJ15:us:geMxs:geNxs:gmUxs:aeMxs:nmUzs:nmUys :deUzs:geUzs:deUys:geUys:aeMzs:aeMys:amUzs:amUys
%fashionablester,fashionable.ADJ15:neMzs:neMys:geFxs:deFxs:dmUxs
%fashionablestes,fashionable.ADJ15:neNzs:neNxs:aeNzs:aeNxs

Nomen:

%Analytiken,Analytik.fem(NS0,NP3):nmF:gmF:dmF:amF
%Apport,Apport.mask(NS2,NP2):neM:deM:aeM
%Canasta,Canasta.neut(NS2,NPSG):neN:deN:aeN
%Dippeln,Dippel.mask(NS2,NP1):dmM
%Dreispitze,Dreispitz.mask(NS10,NP2):deM:nmM:gmM:amM
%Fötus,Fötus.mask(NS0,NP24):neM:geM:deM:aeM
%Feedbacks,Feedback.neut(NS2,NP6):geN:nmN:gmN:dmN:amN
%Finnlandisierung,Finnlandisierung.fem(NS0,NP3):neF:geF:deF:aeF
%Höfleins,Höflein.neut(NS2,NP0):geN
%Heimlichtuer,Heimlichtuer.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM
%Henna,Henna.fem(NS0,NPSG):neF:geF:deF:aeF
%Kommoden,Kommode.fem(NS0,NP4):nmF:gmF:dmF:amF
%Korolla,Korolla.fem(NS0,NP18):neF:geF:deF:aeF
%Pelzes,Pelz.mask(NS10,NP2):geM
%Pirat,Pirat.mask(NS3,NP3):neM
%Rolli,Rolli.mask(NS2,NP6):neM:deM:aeM
%Schindel,Schindel.fem(NS0,NP4):neF:geF:deF:aeF
%Skrubbers,Skrubber.mask(NS2,NP1):geM
%Spukereien,Spukerei.fem(NS0,NP3):nmF:gmF:dmF:amF
%Transliterationen,Transliteration.fem(NS0,NP3):nmF:gmF:dmF:amF
%Vertuschungen,Vertuschung.fem(NS0,NP3):nmF:gmF:dmF:amF

Verben:

%baumeln,baumeln.VSW6:OI:1mGi:3mGi:1mGc:3mGc
%begegneten,begegnen.VSW2s:1mVi:3mVi:1mVc:3mVc
%dribbele,dribbeln.VSW6:1eGi:1eGc:3eGc
%filtertet,filtern.VSW4:2mVi:2mVc
%frequentieret,frequentieren.VSW1s:2mGc
%haltre,haltern.VSW4:1eGi:1eGc:3eGc
%kastelten,kasteln.VSW6:1mVi:3mVi:1mVc:3mVc
%konvertierst,konvertieren.VSW1s:2eGi
%kuppelte,kuppeln.VSW6:1eVi:3eVi:1eVc:3eVc
%maischte,maischen.VSW1:1eVi:3eVi:1eVc:3eVc
%meditieret,meditieren.VSW1s:2mGc
%meißle,meißeln.VSW6:1eGi:1eGc:3eGc
%peitscht,peitschen.VSW1:3eGi:2mGi
%ratifizierend,ratifizieren.VSW1s:OE
%schnarcht,schnarchen.VSW1:3eGi:2mGi
%schnicket,schnicken.VSW1:2mGc
%schoberst,schobern.VSW4:2eGi:2eGc
%schuchtere,schuchtern.VSW5:1eGi:1eGc:3eGc
%schuldeten,schulden.VSW2:1mVi:3mVi:1mVc:3mVc
%schwanzte,schwanzen.VSW3:1eVi:3eVi:1eVc:3eVc
%zeiselst,zeiseln.VSW6:2eGi:2eGc
%zimmern,zimmern.VSW4:OI:1mGi:3mGi:1mGc:3mGc

Funktionswörter:

%keineswegs,keineswegs.ADV
%keinmal,keinmal.ADV
%kennzeichnenderweise,kennzeichnenderweise.ADV
%kieloben,kieloben.ADV
%kistenweise,kistenweise.ADV
%solang,solang.KONJ
%sondern,sondern.KONJ
%sooft,sooft.KONJ
%sosehr,sosehr.KONJ
%nämlich,nämlich.PART
%namens,namens.PART
%nicht,nicht.PART
%noch,noch.PART
%nun,nun.PART
%nur,nur.PART
%ohnehin,ohnehin.PART
%lang,lang.PREP0
%laut,laut.PREP3
%mang,mang.PREP4
%mangels,mangels.PREP5
%mit,mit.PREP3
%abhanden,abhanden.VPART
%drein,drein.VPART
%inne,inne.VPART

Fugenformen:

%Fötzel,Fötzel.mask(NS2,NP1):FF1
%Füßchen,Füßchen.neut(NS2,NP0):FF1
%Füßlein,Füßlein.neut(NS2,NP0):FF1
%Füßler,Füßler.mask(NS2,NP1):FF1
%Füßlerinnen,Füßlerin.fem(NS0,NP5):FF12
%Füßling,Füßling.mask(NS2,NP2):FF1
%Füchschen,Füchschen.neut(NS2,NP0):FF1
%Füchsinnen,Füchsin.fem(NS0,NP5):FF12
%Füchslein,Füchslein.neut(NS2,NP0):FF1
%Fügsamkeits,Fügsamkeit.fem(NS0,NP3):FF13
%Fügungs,Fügung.fem(NS0,NP3):FF13
%Fühlbarkeits,Fühlbarkeit.fem(NS0,NP3):FF13
%Fühler,Fühler.mask(NS2,NP1):FF1
%Fühllosigkeits,Fühllosigkeit.fem(NS0,NP3):FF13
%Fühlsamkeits,Fühlsamkeit.fem(NS0,NP3):FF13
%Fühlungnahme,Fühlungnahme.fem(NS0,NP4):FF1
%Fühlungs,Fühlung.fem(NS0,NP3):FF13
%Fühlungsnahme,Fühlungsnahme.fem(NS0,NP4):FF1
%Führen,Führe.fem(NS0,NP4):FF11
%Führer,Führer.mask(NS2,NP1):FF1
%Führerinnen,Führerin.fem(NS0,NP5):FF12
%Führertums,Führertum.neut(NS2,NP14):FF13

%beginn,beginnen.VST1s:FF
%beiß,beißen.VST11:FF
%berg,bergen.VST1:FF
%berst,bersten.VST8:FF
%beweg,bewegen.VST1s:FF
%bieg,biegen.VST1:FF
%biet,bieten.VST4:FF
%bind,binden.VST4:FF
%birg,bergen.VST1:FF
%birst,bersten.VST8:FF
%bitt,bitten.VST4:FF
%blas,blasen.VST3:FF

%abschlägigst.ADJ0:AFs
%analogst.ADJ0:AFs
%best.ADJu;gut:AFs
%hinterst.ADJ0:AFs
%höchst.ADJ;hoch:AFs
%langweiligst.ADJ0:AFs
%meist.ADJ;viel:AFs
%mittlerst.ADJ0:AFs
%nachst.ADJ;nah:AFs
%oberst.ADJ0:AFs
%unterst.ADJ0:AFs
%vorderst.ADJ0:AFs

Einträge in der EF-DBM:

acid: acid.mask(NS0,NPSG):FF1/acid.neut(NS0,NPSG):FF1/
acid.neut(NS0,NPSG):neN:geN:deN:aeN/acid.mask(NS0,NPSG):neM:geM:deM:aeM
alternativere: alternativ.ADJ0:neFxk:nmUxk:aeFxk:amUxk:neUyk:aeFyk:aeNyk:neFzk
:aeFzk
anästhesiertem: anästhesiert.ADJ0:deMxp:deNxp
anmähest: anmähen.VSWT1#2:2eGc
beherzte: beherzen.VSW3s:1eVi:3eVi:1eVc:3eVc/
beherzt.ADJ0:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp
betränend: betränen.VSW1s:OE/betranend.ADJ0:up
blockendere: blockend.ADJ0:neFxk:nmUxk:aeFxk:amUxk:neUyk:aeFyk:aeNyk:neFzk
:aeFzk
brutalitäten: brutalität.fem(NS0,NP3):FF5/brutalitat.fem(NS0,NP3):nmF:gmF:dmF:amF
chronographischem: chronographisch.ADJ0:deMxp:deNxp
durchgesintert: durchsintern.VSWT4#5:OZ
entproblematisierung: entproblematisierung.fem(NS0,NP3):neF:geF:deF:aeF
enttrümmernderer: enttrümmernd.ADJ0:neMzk:neMyk:geFxk:deFxk:dmNxk:dmMxk
:dmFxk
exkusiert: exkusiert.ADJ0:up/exkusieren.VSW1s:3eGi:2mGi:OZ
fondue: fondue.neut(NS2,NP6):FF1/fondue.neut(NS2,NP6):neN:deN:aeN
gelupfterer: gelupft.ADJ0:neMzk:neMyk:geFxk:deFxk:dmUxk
geluppterem: geluppt.ADJ0:deMxk:deNxk
generellstes: generell.ADJ0:neNzs:neNxs:aeNzs:aeNxs
goldhaltiges: goldhaltig.ADJ0:neNzp:neNxp:aeNzp:aeNxp
greißlereien: greißlerei.fem(NS0,NP3):FF5/
greißlerei.fem(NS0,NP3):nmF:gmF:dmF:amF
herumlaufe: herumlaufen.VSTT1#5:1eGc:3eGc:1eGi
hinwegkommend: hinwegkommen.VSTT1#6:OE
intarsiertest: intarsieren.VSW1s:2eVi:2eVc
kitzligerem: kitzlig.ADJ0:deMxk:deNxk
klitsch: klitsch.mask(NS1,NP2):FF1/klitsch.mask(NS1,NP2):neM:deM:aeM/
klitschen.VSW1:FF/klitsch.INTJ
koto: koto.fem(NS0,NP6):FF1/koto.neut(NS2,NP6):FF1/
koto.fem(NS0,NP6):neF:geF:deF:aeF/koto.neut(NS2,NP6):neN:deN:aeN
nestleins: nestlein.neut(NS2,NP0):geN
reitern: reiter.mask(NS2,NP1):dmM/reiter.fem(NS0,NP4):nmF:gmF:dmF:amF/
reitern.VSW5:OI:1mGi:3mGi:1mGc:3mGc/reiter.fem(NS0,NP4):FF11
schneuzenderes: schneuzend.ADJ0:neNzk:neNxk:aeNzk:aeNxk
stärksten: stark.ADJ4:us:geMxs:geNxs:gmUxs:aeMxs:nmUzs:nmUys:deUzs:geUzs
:deUys:geUys:aeMzs:aeMys:amUzs:amUys
stottererinnen: stottererin.fem(NS0,NP5):nmF:gmF:dmF:amF/
stottererin.fem(NS0,NP5):FF12
triumphier: triumphieren.VSW1s:FF
ulend: ulend.ADJ0:up/ulen.VSW1:OE
unartikuliertheiten: unartikuliertheit.fem(NS0,NP3):nmF:gmF:dmF:amF
vikariat: vikariat.neut(NS1,NP2):neN:deN:aeN/vikariat.neut(NS1,NP2):FF1

Anhang C: Beispiele aus dem CISLEX-AK und -EN

Eigennamen:

arman: arman.EN:X
basil: basil.EN:X
behaim: behaim.EN:X
blunk: blunk.EN:X
boiselle: boiselle.EN:X
bojarski: bojarski.EN:X
eder: eder.EN:X
engelberg: engelberg.EN:X
engeroff: engeroff.EN:X
ensinger: ensinger.EN:X
everac: everac.EN:X
fonseca: fonseca.EN:X
gasselink: gasselink.EN:X
gihl: gihl.EN:X
goldmann: goldmann.EN:X
gordons: gordons.EN:X
hagara: hagara.EN:X
hammill: hammill.EN:X
jochmann: jochmann.EN:X
jopp: jopp.EN:X
kelb: kelb.EN:X
knippler: knippler.EN:X
kopte: kopte.EN:X
krechels: krechels.EN:X
ledl: ledl.EN:X
lydia: lydia.EN:X
murcia: murcia.EN:X
neusel: neusel.EN:X
olson: olson.EN:X
oswald: oswald.EN:X
pallister: pallister.EN:X
paulkes: paulkes.EN:X
rompel: rompel.EN:X
rottier: rottier.EN:X
schenzilorz: schenzilorz.EN:X
schmalix: schmalix.EN:X
schwerbrok: schwerbrok.EN:X
sedlmeyer: sedlmeyer.EN:X
spellerberg: spellerberg.EN:X
viver: viver.EN:X
weesemann: weesemann.EN:X
zach: zach.EN:X

Abkürzungen:

acetv: AcetV.AK:X
aen: AEN.AK:X
agdir: AGDir..AK:X
aiag: AIAG.AK:X
ajle: AJLE.AK:X/AjLE.AK:X
aktr: AktR.AK:X
ama: AMA.AK:X
angv: AngV.AK:X
arci: ARCI.AK:X
are: ARE.AK:X
asym: asym..AK:X
bbm: BBM.AK:X
bdch: Bdch..AK:X
ccrrmm: CCRRMM.AK:X
chbk: ChBK.AK:X
dbr: DBR.AK:X
dprg: DPRG.AK:X
dst-a: Dst.-A..AK:X
dtm: DTM.AK:X
edu: EDU.AK:X
emcc: EMCC.AK:X
erbstva: ErbStVA.AK:X
ermv: ErmV.AK:X
ewre: EWRE.AK:X
fnf: FNF.AK:X
gvz: Gvz..AK:X
hbfl: hbfl..AK:X
hst: HSt..AK:X/Hst..AK:X
hundabgg: HundAbgG.AK:X
hvers: HVers..AK:X
icw: ICW.AK:X
ikj: IKJ.AK:X
inf-kr: Inf.-Kr..AK:X
intelsat: Intelsat.AK:X
plt: PLT.AK:X
pneu: Pneu.AK:X
r-dir: R.-Dir..AK:X
spr-beil: Spr.-Beil..AK:X
stadtver: Stadtver..AK:X
stnog: StNOG.AK:X
ungest: ungest..AK:X
vra: VRA.AK:X
vwj: VWJ.AK:X
wachtm: Wachtm..AK:X
zvl: ZVL.AK:X

Anhang D: Beispiele aus dem numerischen und dem Affixlexikon

Zahlwörter:

zwei,.NUM:0-2
drei,.NUM:0-3
vier,.NUM:0-4
fünf,.NUM:0-5
sechs,.NUM:0-6
sieben,.NUM:0-7
achtzig,.NUM:1-8
neunzig,.NUM:1-9
hundert,.NUM:2-1
tausend,.NUM:3-1
million,.NUM:6-1

Ordinalzahl-Affixe:

%rangig,rangig.OA:up
%rangige,rangig.OA:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp
%rangigem,rangig.OA:deMxp:deNxp
%rangigen,rangig.OA:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp:geUzp :deUyp:geUyp:aeMzp:aeMyp:amUzp:amUyp
%rangiger,rangig.OA:neMzp:neMyp:geFxp:deFxp:dmUxp:uk
%rangigere,rangig.OA:neFxk:nmUxk:aeFxk:amUxk:neUyk:aeFyk:aeNyk:neFzk:aeFzk
%rangigerem,rangig.OA:deMxk:deNxk
%rangigeren,rangig.OA:geMxk:geNxk:gmUxk:aeMxk:nmUzk:nmUyk:deUzk:geUzk :deUyk:geUyk:aeMzk:aeMyk:amUzk:amUyk
%rangigerer,rangig.OA:neMzk:neMyk:geFxk:deFxk:dmUxk
%rangigeres,rangig.OA:neNzk:neNxk:aeNzk:aeNxk
%rangiges,rangig.OA:neNzp:neNxp:aeNzp:aeNxp
%rangigste,rangig.OA:neFxs:nmUxs:aeFxs:amUxs:neUys:aeFys:aeNys:neFzs:aeFzs
%rangigstem,rangig.OA:deMxs:deNxs
%rangigsten,rangig.OA:us:geMxs:geNxs:gmUxs:aeMxs:nmUzs:nmUys:deUzs:geUzs :deUys:geUys:aeMzs:aeMys:amUzs:amUys
%rangigster,rangig.OA:neMzs:neMys:geFxs:deFxs:dmUxs
%rangigstes,rangig.OA:neNzs:neNxs:aeNzs:aeNxs
%kläßlern,kläßler.mask(ONS2,ONP1):dmM
%kläßler,kläßler.mask(ONS2,ONP1):neM:deM:aeM:nmM:gmM:amM

Kardinalzahl-Affixe:

%kant,kant.neut(NSUFFS1,NSUFFP2):neN:deN:aeN
%kants,kant.neut(NSUFFS1,NSUFFP2):geN
%master,master.mask(NSUFFS2,NSUFFP1):neM:deM:aeM:nmM:gmM:amM
%masters,master.mask(NSUFFS2,NSUFFP1):geM
%mastern,master.mask(NSUFFS2,NSUFFP1):dmM
%pfünder,pfünder.mask(NSUFFS2,NSUFFP1):neM:deM:aeM:nmM:gmM:amM
%pfünders,pfünder.mask(NSUFFS2,NSUFFP1):geM
%pfündern,pfünder.mask(NSUFFS2,NSUFFP1):dmM

%prozentig,prozentig.NASUFF0:up
%prozentige,prozentig.NASUFF0:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp
:neFzp:aeFzp
%prozentigem,prozentig.NASUFF0:deMxp:deNxp
%prozentigen,prozentig.NASUFF0:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp
:deUzp:geUzp:deUyp:geUyp:aeMzp:aeMyp:amUzp:amUyp
%prozentiger,prozentig.NASUFF0:neMzp:neMyp:geFxp:deFxp:dmUxp:uk
%prozentiges,prozentig.NASUFF0:neNzp:neNxp:aeNzp:aeNxp
%monatig,monatig.NASUFF0:up
%monatige,monatig.NASUFF0:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp
:neFzp:aeFzp
%monatigem,monatig.NASUFF0:deMxp:deNxp
%monatigen,monatig.NASUFF0:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp
:geUzp:deUyp:geUyp:aeMzp:aeMyp:amUzp:amUyp
%monatiger,monatig.NASUFF0:neMzp:neMyp:geFxp:deFxp:dmUxp:uk
%monatiges,monatig.NASUFF0:neNzp:neNxp:aeNzp:aeNxp
%malig,malig.NASUFF0:up
%malige,malig.NASUFF0:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp
:aeFzp
%maligem,malig.NASUFF0:deMxp:deNxp
%maligen,malig.NASUFF0:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp:geUzp
:deUyp:geUyp:aeMzp:aeMyp:amUzp:amUyp
%maliger,malig.NASUFF0:neMzp:neMyp:geFxp:deFxp:dmUxp:uk
%maliges,malig.NASUFF0:neNzp:neNxp:aeNzp:aeNxp

Präfixe:

%ant,ant.PREF
%chryso,chryso.PREF
%de,de.PREF
%exo,exo.PREF
%ferro,ferro.PREF
%hämo,hämo.PREF
%hekto,hekto.PREF
%ideo,ideo.PREF
%im,im.PREF
%super,super.PREF
%tri,tri.PREF

Suffixe:

%seits,seits.ADVSUFF
%bettige,bettig.ASUFF0:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp
:aeFzp
%förmigstes,förmig.ASUFF0:neNzs:neNxs:aeNzs:aeNxs
%fühligen,fühlig.ASUFF0:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp:geUzp
:deUyp:geUyp:aeMzp:aeMyp:amUzp:amUyp
%farbnerem,farben.ASUFF2:deMxk:deNxk
%mutzigerer,mutzig.ASUFF0:neMzk:neMyk:geFxk:deFxk:dmUxk
%prozentigem,prozentig.ASUFF0:deMxp:deNxp
%rippigem,rippig.ASUFF0:deMxp:deNxp

%schnauzigstes,schnauzig.ASUFF0:neNzs:neNxs:aeNzs:aeNxs
%schrötiges,schrötig.ASUFF0:neNzp:neNxp:aeNzp:aeNxp
%semestrigstes,semestrig.ASUFF0:neNzs:neNxs:aeNzs:aeNxs
%tägigem,tägig.ASUFF0:deMxp:deNxp
%turmiger,turmig.ASUFF0:neMzp:neMyp:geFxp:deFxp:dmUxp:uk
%ufriger,ufrig.ASUFF0:neMzp:neMyp:geFxp:deFxp:dmUxp:uk
%zahnigerer,zahnig.ASUFF0:neMzk:neMyk:geFxk:deFxk:dmUxk
%zylindrigster,zylindrig.ASUFF0:neMzs:neMys:geFxs:deFxs:dmUxs

Anhang E: Beispiele aus den Speziallisten

Reduktionsformen nach Apostroph am Wortanfang:

mal,einmal.ADV
nauf,hinauf.ADV
naus,hinaus.ADV
nen,ein.DET2:dmN:aeM
ne,ein.DET2:amM:nmM:aeF:neF
ner,ein.DET2:gmM:geF:deF
nem,ein.DET2:deM:deN
raus,heraus.ADV
rüber,herüber.ADV
rum,herum.ADV
runter,herunter.ADV
rein,herein.ADV
runter,herunter.ADV
s,es.Pron:n3eN:a3eN
zefix,kruzifix.INTJ
tschuldigung,entschuldigung.fem(NS0,NP3):neF:geF:deF:aeF

Lexikalisierte Apostrophformen:

rock'n'roll,rock'n'roll.mask(NS2,NP6):neM:deM:aeM
rock'n'rolls,rock'n'roll.mask(NS2,NP6):geM:nmM:gmM:dmM:amM
wies'n,wiesen.fem(NS0,NPSG):neF:geF:deF:aeF
heil'gen,heilig.ADJ0:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp:geUzp :deUyp:geUyp:aeMzp:aeMyp:amUzp:amUyp
heil'ge,heilig.ADJ0:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp
heil'ger,heilig.ADJ0:neMzp:neMyp:geFxp:deFxp:dmUxp:uk
heil'ges,heilig.ADJ0:neNzp:neNxp:aeNzp:aeNxp
heil'gem,heilig.ADJ0:deMxp:deNxp

Lexikalisierte BZX- und BSZX-Formen

1a,1A.AK:X
1plus,1Plus.EN:X
3-glocken-nudeln,3-Glocken-Nudeln.EN:X
3d,3D.AK:X
3sat,3SAT.EN:X
3sat,3satTextVision.EN:X
a-321,A-321.AK:X
a-340,A-340.AK:X
bündnis90,Bündnis90.EN:X
bel-20-index,Bel-20-Index.EN:X
boeing777,Boeing777.EN:X
channel3,Channel3.EN:X
christoph15,Christoph15.EN:X
d1,D1.AK:X
d2,D2.AK:X
dc9,DC9.AK:X

do328,Do328.AK:X
dornier328,Dornier328.EN:X
electronic2000,Electronic2000.EN:X
g7,G7.AK:X
gsg9,GSG-9.AK:X
k2,K2.EN:X
nikkei-225,Nikkei-225.EN:X
pro7,Pro7.EN:X
rtl2,RTL2.AK:X
sat1,SAT1.EN:X
sat1,Sat1.EN:X
v6,V6.AK:X

Liste der Nomenpaare mit häufiger Endgliedambiguität:

<salbe, albe>[1]
<samen, amen>
<amt, samt>
<sarg, arg>
<satzung, atzung>
<scharte, charte>
<schi, chi>
<egge, segge>
<ehe, sehe>
<eiche, seiche>
<eifer, seifer>
<ekel, sekel>
<enge, senge>
<enkel, senkel>
<ente, sente>
<erpel, serpel>
<esel, sesel>
<ester, sester>
<setter, etter>
<hit, shit>
<igel, sigel>
<kater, skater>
<ode, sode>
<ohr, sohr>
<oma, soma>
<spagat, pagat>
<span, pan>
<park, spark>
<part, spart>
<sparte, parte>
<speer, peer>
<pelz, spelz>
<pille, spille>
<spinne, pinne>
<spion, pion>
<podium, spodium>
<pore, spore>
<sport, port>
<spott, pott>
<tätigkeit, stätigkeit>
<stürmer, türmer>
<tabelle, stabelle>
<tafel, stafel>
<tag, stag>
<stand, tand>
<tank, stank>
<stau, tau>
<taucher, staucher>
<taufe, staufe>
<steak, teak>
<teig, steig>
<teilung, steilung>
<stein, tein>
<teller, steller>
<sticker, ticker>
<stop, top>
<stopfen, topfen>
<straß, traß>
<straps, traps>
<strebe, trebe>
<streber, treber>
<trecker, strecker>
<tritt, stritt>
<stummel, tummel>
<stute, tute>

1. Das jeweils erste Element wird präferiert.

Anhang F: Lemmatisierung eines Textbeispiels

Streiflicht, Süddt. Zeitung vom 5.1.1994:

Wir stehen hier mit unserem Streiflicht an der Schwelle des jung-dynamischen Jahres und blicken wie gebannt nach vorn - ahhh, ist doch was anderes als der alte zähe Knochen 1993. Zuerst . . . bitte, ist denn das die Möglichkeit?! Wollja, das muß es sein! Langvermißt und erotisierend: der Schweißgeruch der Leistungsbereitschaft. Über Nacht hat im ganzen Lande Besinnung eingesetzt. An jeder Ecke schießen Maso-Stationen aus dem Boden. Hungrige Mitarbeiter aus allen Branchen entblößen ihre Filetstücke und jubeln unter den gezielten Schlägen ihrer Verschlanker: Ich werde mich quälen quälen quälen! Vierzig Stunden und: mehr mehr mehr! - Maschinenlaufzeiten: süß süß süß! - Feiertage: weg weg weg! Macht richtig Spaß, zuzusehen. Aber was glänzt da vorn und blendet jeden Gedanken? Das muß der Gipfel der Macht sein - oder bloß ihr Zipfel? Achtzehn Wahlen führen dorthin. Zunächst einmal verbieten wir jeder Frau und jedem Mann in unserem gescheiten Lande die Dummwörter: Superwahljahr und Megawahljahr. Verstanden? Beide hiermit verboten! Da vorn, am Fuße der Macht in Camp A, lauert übrigens Bundeskanzler (Der Breite) Kohl auf Sherpa (Der Drahtige) Scharping, der soll ihn irgendwie nach oben schleppen, weil man annimmt, daß er's allein nicht mehr schafft. Aber wie? Geschultert? Untergeschnallt? Fleißig üben sie am Nordhang des Bonner Venusberges. Nebenan, in Camp B, brüllen die Wahlmanager aller Parteien übend aufeinander ein; man hört Fetzen: . . . Einheit gewollt . . . Verhindert . . . Selbstbediener . . . Sie persönlicher Arbeitsloser . . . politikverdrossene Memme . . . Ossi . . . Wessi . . . Bonnmörder . . . Berlinhure! Mittendrin, an Marterpfähle gefesselt, etliche ehemals stolze Stimmbürger. An ihnen sollte ausprobiert werden, ob der mit Enthüllung, Lüge und Rechthaber-Video monatelang traktierte Stopf-Mensch überhaupt bis zur Bundestagswahl im Oktober wird durchhalten können. [...]

Ausgabe des Lemmatisierers:

Wir stehen hier mit unserem Streiflicht an der Schwelle des jung-dynamischen Jahres und blicken wie gebannt nach vorn - ahhh, ist doch was anderes als der alte zähe Knochen 1993 .

WB Wir: wir.Pron:n1mU
WB stehen: stehen.VUNR:OI:1mGi:3mGi:1mGc:3mGc
WB hier: hier.ADV
WB mit: mit.PREP3
WB unserem: unser.DET2:deM:deN
WB Streiflicht: streif||licht.neut(NS1,NP8):neN:deN:aeN##N+V,N
WB an: an.PREP4
WB der: d.DET4:gmM:neM:geF:deF
WB Schwelle: schwelle.fem(NS0,NP4):neF:geF:deF:aeF
WB des: d.DET4:geM:geN
BSX jung-dynamischen: jung-dynamisch.ADJ0:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp:geUzp:deUyp
:geUyp:aeMzp:aeMyp:amUzp:amUyp###ADJ+V-ADJ
jung-dynamische.NA:geMx:geNx:gmUx:aeMx:nmUz:nmUy:deUz:geUz:deUy:geUy:aeMz:aeMy:amUz:amUy
###ADJ+V-NA
WB Jahres: jahr.neut(NS1,NP2):geN
WB und: und.KONJ
WB blicken: blicken.VSW1:OI:1mGi:3mGi:1mGc:3mGc
WB wie: wie.KONJ wie.ADV wie.PREP0 wie.Pron:X
WB gebannt: bannen.VSW1:OZ

WB nach: nach.PREP3
WB vorn: vor.PDET:aeM vorn.ADV
SX -: SX(-).CC
WB ahhh: WB(ahhh).UK
SX ,: SX(,).CC
WB ist: sein.VUNR:3eGi
WB doch: doch.PART doch.KONJ doch.ADV
WB was: was.ADV w.Pron:n3eU:a3eU:d3eU:g3eU
WB anderes: ander.ADJ2:neNzp:neNxp:aeNzp:aeNxp
WB als: als.KONJ als.PREP0
WB der: d.DET4:gmM:neM:geF:deF
WB alte: alt.ADJ6:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp
WB zähe: zäh.ADJ12:up:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp
WB Knochen: knochen.mask(NS2,NP0):neM:deM:aeM:nmM:gmM:dmM:amM
ZX 1993: Integerzahl(1993).NUM
SX .: SX(.).CC

Zuerst . . . bitte, ist denn das die Möglichkeit?!

WB Zuerst: zuerst.ADV
SX .: SX(.).CC SX .: SX(.).CC SX .: SX(.).CC
WB bitte: bitten.VST4:1eGi:1eGc:3eGc
SX ,: SX(,).CC
WB ist: sein.VUNR:3eGi
WB denn: denn.PART denn.KONJ
WB das: d.DET4:neN:aeN
WB die: d.DET4:nmN:amN:aeF:neF
WB Möglichkeit: möglichkeit.fem(NS0,NP3):neF:geF:deF:aeF
SX ?: SX(?).CC SX !: SX(!).CC

Wollja, das muß es sein!

WB Wollja: WB(wollja).UK
SX ,: SX(,).CC
WB das: d.DET4:neN:aeN
WB muß: müssen.VUNR:1eGi:3eGi
WB es: es.Pron:n3eN:a3eN
WB sein: sein.DET2:neN:aeN:neM sein.VUNR:OI
SX !: SX(!).CC

Langvermißt und erotisierend: der Schweißgeruch der Leistungsbereitschaft .

WB Langvermißt: lang||vermißt.ADJ0:up##ADJ+PREP+V,ADJ
lang||vermissen.VSW8s:3eGi:OZ:2mGi:2eGi##ADJ+PREP+V,V
WB und: und.KONJ
WB erotisierend: erotisieren.VSW1s:OE
SX :: SX(:).CC
WB der: d.DET4:gmM:neM:geF:deF
WB Schweißgeruch: schweiß||geruch.mask(NS1,NP12):neM:deM:aeM##N+V,N
WB der: d.DET4:gmM:neM:geF:deF
WB Leistungsbereitschaft: leistungs||bereitschaft.fem(NS0,NP3):neF:geF:deF:aeF##N,N
SX .: SX(.).CC

Über Nacht hat im ganzen Lande Besinnung eingesetzt .

WB Über: über.PREP4
WB Nacht: nacht.fem(NS0,NP12):neF:geF:deF:aeF
WB hat: haben.VUNR:3eGi
WB im: in.PDET:deM:deN
WB ganzen: ganz.ADJ11:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp:geUzp:deUyp:geUyp:aeMzp
:aeMyp:amUzp:amUyp
WB Lande: land.neut(NS1,NP14):deN land.neut(NS1,NP2):deN:nmN:gmN:amN
WB Besinnung: besinnung.fem(NS0,NP3):neF:geF:deF:aeF

WB eingesetzt: einsetzen.VSWT3#3:OZ
SX .: SX(.).CC

An jeder Ecke schießen Maso-Stationen aus dem Boden .

WB An: an.PREP4
WB jeder: jed.DET3:gmM:neM:geF:deF
WB Ecke: ecke.fem(NS0,NP4):neF:geF:deF:aeFeck.neut(NS1,NP2):deN:nmN:gmN:amN
WB schießen: schießen.VST11:OI:1mGi:3mGi:1mGc:3mGc
BSX Maso-Stationen: maso-station.fem(NS0,NP3):nmF:gmF:dmF:amF###UK-N
WB aus: aus.PREP3
WB dem: d.DET4:deM:deN
WB Boden: boden.mask(NS2,NP13):neM:deM:aeM
SX .: SX(.).CC

Hungrige Mitarbeiter aus allen Branchen entblößen ihre Filetstücke und jubeln unter den gezielten Schlägen ihrer Verschlanker: Ich werde mich quälen quälen quälen!

WB Hungrige: hungrig.ADJ0:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp
hungrige.NA:neFx:nmUx:aeFx:amUx:neUy:aeFy:aeNy:neFz:aeFz
WB Mitarbeiter: mit||arbeiter.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM##PREP,N
WB aus: aus.PREP3
WB allen: alle.DET1:dmN:geM:geN:aeM
WB Branchen: branche.fem(NS0,NP4):nmF:gmF:dmF:amF
WB entblößen: entblößen.VSW3s:OI:1mGi:3mGi:1mGc:3mGc
WB ihre: ihr.DET2:amM:nmM:aeF:neF
WB Filetstücke: filet||stück.neut(NS1,NP2):deN:nmN:gmN:amN##N,N
filets||tücke.fem(NS0,NP4):neF:geF:deF:aeF##N,N
WB und: und.KONJ
WB jubeln: jubeln.VSW6:OI:1mGi:3mGi:1mGc:3mGc
WB unter: unter.PREP4 unter.ADJ0:up
WB den: d.DET4:dmN:dmM:dmF:aeM
WB gezielten: gezielt.ADJ11:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp:geUzp:deUyp:geUyp
:aeMzp:aeMyp:amUzp:amUyp
WB Schlägen: schlag.mask(NS1,NP12):dmM
WB ihrer: sie.Pron:g3eF:g3mU ihr.DET2:gmM:geF:deF
WB Verschlanker: verschlanker.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM
SX :: SX(:).CC
WB Ich: ich.Pron:n1eU
WB werde: werden.VUNR:1eGi:1eGc:3eGc
WB mich: ich.Pron:a1eU
WB quälen: quälen.VSW1:OI:1mGi:3mGi:1mGc:3mGc WB quälen: quälen.VSW1:OI:1mGi:3mGi:1mGc:3mGc
WB quälen: quälen.VSW1:OI:1mGi:3mGi:1mGc:3mGc
SX !: SX(!).CC

Vierzig Stunden und: mehr mehr mehr!

WB Vierzig: vierzig.ADJ
WB Stunden: stunde.fem(NS0,NP4):nmF:gmF:dmF:amF stunden.neut(NS2,NPSG):neN:deN:aeN
WB und: und.KONJ
SX :: SX(:).CC
WB mehr: viel.ADJu4:uk mehr.ADV WB mehr: viel.ADJu4:ukmehr.ADV WB mehr: viel.ADJu4:ukmehr.ADV
SX !: SX(!).CC

- Maschinenlaufzeiten: süß süß süß!

SX -: SX(-).CC
WB Maschinenlaufzeiten: maschinen||lauf||zeit.fem(NS0,NP3):nmF:gmF:dmF:amF##N,N+V,N
SX :: SX(:).CC
WB süß: süß.ADJ11:up WB süß: süß.ADJ11:up WB süß: süß.ADJ11:up
SX !: SX(!).CC

- Feiertage: weg weg weg!

SX -: SX(-).CC
WB Feiertage: feier||tag.mask(NS1,NP2):deM:nmM:gmM:amM##N+V,N
SX :: SX(:).CC
WB weg: weg.KONJ WB weg: weg.KONJ WB weg: weg.KONJ
SX !: SX(!).CC

Macht richtig Spaß, zuzusehen .

WB Macht: machen.VSW1:3eGi:2mGi macht.fem(NS0,NP12):neF:geF:deF:aeF
WB richtig: richtig.ADJ0:up richtig.PART
WB Spaß: spaß.mask(NS10,NP12):neM:deM:aeM
SX ,: SX(,).CC
WB zuzusehen: zusehen.VSTT1#2:OI
SX .: SX(.).CC

Aber was glänzt da vorn und blendet jeden Gedanken?

WB Aber: aber.PART aber.KONJ aber.neut(NS2,NP1):neN:deN:aeN:nmN:gmN:amN
WB was: was.ADV w.Pron:n3eU:a3eU:d3eU:g3eU
WB glänzt: glänzen.VSW3:2eGi:3eGi:2mGi
WB da: da.KONJda.ADV
WB vorn: vor.PDET:aeM vorn.ADV
WB und: und.KONJ
WB blendet: blenden.VSW2:3eGi:2mGi:2mGc
WB jeden: jed.DET3:dmN:aeM
WB Gedanken: gedanke.mask(NS6,NP4):deM:aeM:nmM:gmM:dmM:amM
SX ?: SX(?).CC

Das muß der Gipfel der Macht sein - oder bloß ihr Zipfel?

WB Das: d.DET4:neN:aeN
WB muß: müssen.VUNR:1eGi:3eGi
WB der: d.DET4:gmM:neM:geF:deF
WB Gipfel: gipfel.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM
WB der: d.DET4:gmM:neM:geF:deF
WB Macht: macht.fem(NS0,NP12):neF:geF:deF:aeF
WB sein: sein.DET2:neN:aeN:neM sein.VUNR:OI
SX -: SX(-).CC
WB oder: oder.KONJ
WB bloß: bloß.ADJ11:up bloß.PART bloß.ADV
WB ihr: ihr.DET2:neN:aeN:neM sie.Pron:d3eF
WB Zipfel: zipfel.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM
SX ?: SX(?).CC

Achtzehn Wahlen führen dorthin .

WB Achtzehn: acht||zehn.ADJ##N+ADJ+V,ADJ 18.NUM
WB Wahlen: wahl.fem(NS0,NP3):nmF:gmF:dmF:amF
WB führen: führen.VSW1:OI:1mGi:3mGi:1mGc:3mGc fahren.VST1:1mVc:3mVc
WB dorthin: dorthin.ADV
SX .: SX(.).CC

Zunächst einmal verbieten wir jeder Frau und jedem Mann in unserem gescheiten Lande die Dummwörter: Super-wahljahr und Megawahljahr .

WB Zunächst: zunächst.ADV zunächst.PREP2
WB einmal: einmal.PART einmal.ADV
WB verbieten: verbieten.VST4s:OI:1mGi:3mGi:1mGc:3mGc
WB wir: wir.Pron:n1mU
WB jeder: jed.DET3:gmM:neM:geF:deF
WB Frau: frau.fem(NS0,NP3):neF:geF:deF:aeF

WB und: und.KONJ
WB jedem: jed.DET3:deM:deN
WB Mann: mann.mask(NS1,NP14):neM:deM:aeM
WB in: in.PREP4
WB unserem: unser.DET2:deM:deN
WB gescheiten: gescheit.ADJ11:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp:geUzp:deUyp:geUyp:aeMzp
:aeMyp:amUzp:amUyp
WB Lande: land.neut(NS1,NP14):deN land.neut(NS1,NP2):deN:nmN:gmN:amN
WB die: d.DET4:nmN:amN:aeF:neF
WB Dummwörter: dumm||wort.neut(NS1,NP14):nmN:gmN:amN##ADJ,N
SX :: SX(:).CC
WB Superwahljahr: super||wahl||jahr.neut(NS1,NP2):neN:deN:aeN##ADJ+N,N,N
WB und: und.KONJ
WB Megawahljahr: mega||wahl||jahr.neut(NS1,NP2):neN:deN:aeN##PREF,N,N
SX .: SX(.).CC

Verstanden?

WB Verstanden: verstehen.VUNRs:1mVi:OZ:3mVi
SX ?: SX(?).CC

Beide hiermit verboten!

WB Beide: beid.DET2:amN:nmN
WB hiermit: hiermit.ADV
WB verboten: verbieten.VST4s:1mVi:OZ:3mVi
SX !: SX(!).CC

Da vorn, am Fuße der Macht in Camp A, lauert übrigens Bundeskanzler (Der Breite) Kohl auf Sherpa (Der Drahtige) Scharping, der soll ihn irgendwie nach oben schleppen, weil man annimmt, daß er's allein nicht mehr schafft .

WB Da: da.KONJda.ADV
WB vorn: vor.PDET:aeM vorn.ADV
SX ,: SX(,).CC
WB am: an.PDET:deM:deN
WB Fuße: fuß.mask(NS10,NP12):deM
WB der: d.DET4:gmM:neM:geF:deF
WB Macht: macht.fem(NS0,NP12):neF:geF:deF:aeF
WB in: in.PREP4
WB Camp: camp.neut(NS2,NP6):neN:deN:aeN
CB A: a.AK:X
SX ,: SX(,).CC
WB lauert: lauern.VSW4:3eGi:2mGi:2mGc
WB übrigens: übrigens.PARTübrigens.ADV
WB Bundeskanzler: bundes||kanzler.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM##N,N
SX (: SX(().CC
WB Der: d.DET4:gmM:neM:geF:deF
WB Breite: breite.NA:neFx:nmUx:aeFx:amUx:neUy:aeFy:aeNy:neFz:aeFz
SX): SX()).CC
WB Kohl: kohl.mask(NS1,NP2):neM:deM:aeM
WB auf: auf.PREP4
WB Sherpa: sherpa.mask(NS2,NP6):neM:deM:aeM
SX (: SX(().CC
WB Der: d.DET4:gmM:neM:geF:deF
WB Drahtige: drahtige.NA:neFx:nmUx:aeFx:amUx:neUy:aeFy:aeNy:neFz:aeFz
SX): SX()).CC
WB Scharping: scharping.EN:X
SX ,: SX(,).CC
WB der: d.DET4:gmM:neM:geF:deF
WB soll: sollen.VUNR:1eGi:3eGi:eb
WB ihn: er.Pron:a3eM
WB irgendwie: irgendwie.ADV
WB nach: nach.PREP3

WB oben: oben.ADV
WB schleppen: schleppen.VSW1:OI:1mGi:3mGi:1mGc:3mGc
SX ,: SX(,).CC
WB weil: weil.KONJ
WB man: man.PART man.Pron:n3eU:a3eU:d3eU:g3eU
WB annimmt: annehmen.VUNRT#2:3eGi
SX ,: SX(,).CC
WB daß: daß.KONJ
BSX er's: (er+es/sie).Pron:n3eM
WB allein: allein.ADJ:X allein.PART allein.ADV allein.KONJ
WB nicht: nicht.PART nicht.ADV
WB mehr: viel.ADJu4:uk mehr.ADV
WB schafft: schaffen.VST1:2mGi:3eGi
SX .: SX(.).CC

Aber wie?

WB Aber: aber.PART aber.KONJ aber.neut(NS2,NP1):neN:deN:aeN:nmN:gmN:amN
WB wie: wie.KONJ wie.ADV wie.PREP0 wie.Pron:X
SX ?: SX(?).CC

Geschultert?

WB Geschultert: schultern.VSW4:OZ
SX ?: SX(?).CC

Untergeschnallt?

WB Untergeschnallt: unter||schnallen.VSW1:OZ##N+PREP+ADJ,V unter||geschnallt.ADJ0:up##N+PREP+ADJ,ADJ
SX ?: SX(?).CC

Fleißig üben sie am Nordhang des Bonner Venusberges .

WB Fleißig: fleißig.ADJ0:up
WB üben: üben.VSW1:OI:1mGi:3mGi:1mGc:3mGc
WB sie: sie.Pron:n3eF:a3eF
WB am: an.PDET:deM:deN
WB Nordhang: nord||hang.mask(NS1,NP12):neM:deM:aeM##N+ADV,N
WB des: d.DET4:geM:geN
WB Bonner: bonner.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM
WB Venusberges: venus||berg.mask(NS1,NP2):geM##EN,N
SX .: SX(.).CC

Nebenan, in Camp B, brüllen die Wahlmanager aller Parteien übend aufeinander ein; man hört Fetzen: . . .

WB Nebenan: nebenan.ADV
SX ,: SX(,).CC
WB in: in.PREP4
WB Camp: camp.neut(NS2,NP6):neN:deN:aeN
CB B: b.AK:X
SX ,: SX(,).CC
WB brüllen: brüllen.VSW1:OI:1mGi:3mGi:1mGc:3mGc
WB die: d.DET4:nmN:amN:aeF:neF
WB Wahlmanager: wahl||manager.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM##N,N
WB aller: alle.DET1:gmM:neM:geF:deF
WB Parteien: partei.fem(NS0,NP3):nmF:gmF:dmF:amF
WB übend: üben.VSW1:OE
WB aufeinander: aufeinander.ADV
WB ein: ein.DET2:neN:aeN:neM
SX ;: SX(;).CC
WB man: man.PART man.Pron:n3eU:a3eU:d3eU:g3eU
WB hört: hören.VSW1:3eGi:2mGi

WB Fetzen: fetzen.mask(NS2,NP0):neM:deM:aeM:nmM:gmM:dmM:amM fetzen.neut(NS2,NPSG):neN:deN:aeN
SX :: SX(:).CC SX .: SX(.).CC SX .: SX(.).CC SX .: SX(.).CC

Einheit gewollt . . .

WB Einheit: einheit.fem(NS0,NP3):neF:geF:deF:aeF
WB gewollt: gewollt.ADJ10:up wollen.VUNR:OZ
SX .: SX(.).CC SX .: SX(.).CC SX .: SX(.).CC

Verhindert . . .

WB Verhindert: verhindern.VSW4s:3eGi:2mGc:OZ:2mGi verhindert.ADJ10:up
SX .: SX(.).CC SX .: SX(.).CC SX .: SX(.).CC

Selbstbediener . . .

WB Selbstbediener: selbst||bediener.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM##PART+N+ADV,N
SX .: SX(.).CC SX .: SX(.).CC SX .: SX(.).CC

Sie persönlicher Arbeitsloser . . . politikverdrossene Memme . . .

WB Sie: sie.Pron:n3eF:a3eF:n2mU:a2mU
WB persönlicher: persönlich.ADJ0:neMzp:neMyp:geFxp:deFxp:dmUxp:uk
WB Arbeitsloser: arbeitslose.NA:neMz:neMy:geFx:deFx:dmUx
SX .: SX(.).CC SX .: SX(.).CC SX .: SX(.).CC
WB politikverdrossene: politik||verdrossen.ADJ0:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp
##N,ADJ
WB Memme: memme.fem(NS0,NP4):neF:geF:deF:aeF
SX .: SX(.).CC SX .: SX(.).CC SX .: SX(.).CC

Ossi . . .

WB Ossi: ossi.mask(NS2,NP6):neM:deM:aeM
SX .: SX(.).CC SX .: SX(.).CC SX .: SX(.).CC

Wessi . . .

WB Wessi: wessi.mask(NS2,NP6):neM:deM:aeM
SX .: SX(.).CC SX .: SX(.).CC SX .: SX(.).CC

Bonnmörder . . .
WB Bonnmörder: bonn||mörder.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM##EN,N
SX .: SX(.).CC SX .: SX(.).CC SX .: SX(.).CC

Berlinhure!

WB Berlinhure: berlin||hure.fem(NS0,NP4):neF:geF:deF:aeF##EN,N
SX !: SX(!).CC

Mittendrin, an Marterpfähle gefesselt, etliche ehemals stolze Stimmbürger .

WB Mittendrin: mittendrin.ADV
SX ,: SX(,).CC
WB an: an.PREP4
WB Marterpfähle: marter||pfahl.mask(NS1,NP12):nmM:gmM:amM##N+V,N
WB gefesselt: fesseln.VSW6:OZ
SX ,: SX(,).CC
WB etliche: etlich.ADJ0:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp
WB ehemals: ehemals.ADV
WB stolze: stolz.ADJ11:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp
WB Stimmbürger: stimm||bürger.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM##V,N
SX .: SX(.).CC

An ihnen sollte ausprobiert werden, ob der mit Enthüllung, Lüge und Rechthaber-Video monatelang traktierte Stopf-Mensch überhaupt bis zur Bundestagswahl im Oktober wird durchhalten können .

WB An: an.PREP4
WB ihnen: sie.Pron:d3mU
WB sollte: sollen.VUNR:1eVi:3eVi:1eVc:3eVc
WB ausprobiert: ausprobieren.VSWT1s#3:3eGi:OZ:2mGi
WB werden: werden.VUNR:OI:1mGi:3mGi:1mGc:3mGc
SX ,: SX(,).CC
WB ob: ob.PREP5
WB der: d.DET4:gmM:neM:geF:deF
WB mit: mit.PREP3
WB Enthüllung: enthüllung.fem(NS0,NP3):neF:geF:deF:aeF
SX ,: SX(,).CC
WB Lüge: lüge.fem(NS0,NP4):neF:geF:deF:aeF
WB und: und.KONJ
BSX Rechthaber-Video: rechthaber-video.neut(NS2,NP6):neN:deN:aeN###N-N
WB monatelang: monate||lang.ADJ17:up##N,ADJ
WB traktierte: traktiert.ADJ0:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp
traktieren.VSW1s:1eVi:3eVi:1eVc:3eVc
BSX Stopf-Mensch: stopf-mensch.mask(NS3,NP3):neM###V-N
WB überhaupt: überhaupt.PART überhaupt.ADV
WB bis: bis.KONJ bis.PREP7
WB zur: zu.PDET:deF
WB Bundestagswahl: bundes||tags||wahl.fem(NS0,NP3):neF:geF:deF:aeF##N,N+ADV,N
WB im: in.PDET:deM:deN
WB Oktober: oktober.mask(NS2,NP1):neM:deM:aeM:nmM:gmM:amM
WB wird: werden.VUNR:3eGi
WB durchhalten: durchhalten.VSTT6#5:3mGi:OI:1mGc:3mGc:1mGi
WB können: können.VUNR:OI:1mGi:3mGi:1mGc:3mGc
SX .: SX(.).CC

Anhang G: Lemmatisierung von Sonderformen

ZSX '68:
    ApoJahr(NN68).NUM

BSX 'ne:
    ein.DET2:amM:nmM:aeF:neF

ZSX +11%:
    Prozentzahl(signIntegerzahl(11)).NUM

ZSX -100:
    signIntegerzahl(100).NUM

ZSX -12%:
    Prozentzahl(signIntegerzahl(12)).NUM

BSZX 0190-Telephonnummern:
    0190-telephon||nummer.fem(NS0,NP4):nmF:gmF:dmF:amF##N,N###NUM-N,N

ZSX 1-5:
    Differenzangabe(Integerzahl(1),Integerzahl(5)).NUM

BSZX 1-Megabit-Chip:
    1-megabit-chip.mask(NS2,NP6):neM:deM:aeM###NUM-PREF,N-N

BSZX 10%ige:
    10||%ig.ZA0:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp

BSZX 100%-Beteiligung:
    100%-beteiligung.fem(NS0,NP3):neF:geF:deF:aeF###NUM-N

BSZX 1000er-Grenze:
    1000er-grenze.fem(NS0,NP4):neF:geF:deF:aeF###NUM,N-N

BSZX 10e-Abschreibung:
    10e-abschreibung.fem(NS0,NP3):neF:geF:deF:aeF###UK-N

BZX 16jährige:
      16||jährig.ZA0:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp##NUM,A
    16||jährige.NA:neFx:nmUx:aeFx:amUx:neUy:aeFy:aeNy:neFz:aeFz##NUM,NA

BZX 1976er:
    1976||er.mask(ZNS2,ZNP1):neM:deM:aeM:nmM:gmM:amM##NUM,N

BSZX A8-Modell:
    a8-modell.neut(NS2,NP2):neN:deN:aeN###AK-N

BSZX Audi-V6-Modell:
    audi-v6-modell.neut(NS2,NP2):neN:deN:aeN###EN-AK-N

BSZX BMW-3er-Reihe:
    bmw-3er-reihe.fem(NS0,NP4):neF:geF:deF:aeF###AK-NUM,N-N

BSX Ben's-Reis:
    ben's-reis.neut(NS10,NP8):neN:deN:aeN###EN-N
    ben's-reis.mask(NS10,NP2):neM:deM:aeM###EN-N

```
BSX C&A-Handelsmarke:
    c&a-handels||marke.fem(NS0,NP4):neF:geF:deF:aeF##N,N###AK&AK-N,N

BSX CD's:
    CD.AK:g:m

BSX Delors':
    delors.EN:g

BSX Fichtel&Sachs:
    pfolge(fichtel,sachs).EN:X###EN&EN

BSX Thyssen'sche:
    thyssensch.END:neFxp:nmUxp:aeFxp:amUxp:neUyp:aeFyp:aeNyp:neFzp:aeFzp

BSX auf'n:
    (auf+den).PREP4

BSX bring':
    bringen.VUNR:1eGi:1eGc:3eGc

BSX z'viel:
    zuviel.neut(NS2,NPSG):neN:deN:aeN
    zuviel.Pron:n3eU:a3eU:d3eU:g3eU
```

Anhang H: Beispiel für einen Programmdurchlauf

Unten abgedruckt ist ein vollständiger Programmdurchlauf des Wortes *Verbündeten*. Um auch die Funktionen der Tokenisierung zu illustrieren wurde ein minimaler Kontext miteingegeben, dessen Lemmatisierung allerdings ausgelassen wurde. Der Ausdruck enthält jeweils die Funktionsaufrufe mit den entsprechenden Argumenten (z.B. ws_vektor(die):) und das Ergebnis des Funktionsaufrufs, das in der Form "$f(x_1,...,x_n) = y_1...y_m$" (z.B. ws_vektor(die) = die@WB:die:3:0:0:0:::1:SA) dargestellt ist. Liefert ein Funktionsaufruf kein Ergebnis, so ist der Rückgabewert leer (z.B. check_zuInf(verbündeten) =).

Eingabe: die Verbündeten, der

```
splitte(die Verbündeten, der):
   del_satzzeichen(die):
   del_satzzeichen(die) = die
   del_satzzeichen(Verbündeten,):
      del_satzzeichen(Verbündeten):
      del_satzzeichen(Verbündeten) = Verbündeten
         del_satzzeichen(Verbündeten,) = Verbündeten|,
      del_satzzeichen(der):
      del_satzzeichen(der) = der
   ws_vektor(die):
   ws_vektor(die) = die@WB:die:3:0:0:0:::1:SA
   ws_vektor(Verbündeten):
   ws_vektor(Verbündeten) = verbündeten@WB:Verbündeten:11:0:1:0:::0:SX
   ws_vektor(,):
   ws_vektor(,) = ,@SX::,:1:0:0:0:::0:WB
   ws_vektor(der):
   ws_vektor(der) = der@WB:der:3:0:0:0:::1:SX
splitte(die Verbündeten, der) = die@WB:die:3:0:0:0:::1:SA verbündeten@WB:Verbündeten:11:0:1:0:::0:WB
    ,@SX:,:1:0:0:0:::0:WB der@WB:der:3:0:0:0:::1:SX

lemmatisiere(die@WB:die:3:0:0:0:::1:SA):
   ...
lemmatisiere(die@WB:die:3:0:0:0:::1:SA) = [die]:d.DET4:nmN:amN:aeF:neF

lemmatisiere(verbündeten@WB:Verbündeten:11:0:1:0:::0:WB):
   lemmatisiere_WB(verbündeten,WB Verbündeten 11 0 1 0    0 WB):
      look_up_ef(verbündeten):
         $LEX{verbündeten} = verbünden.VSW2s:1mVi:3mVi:1mVc:3mVc/
            verbündet.ADJ0:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp:geUzp:deUyp:geUyp:aeMzp:aeMyp
            :amUzp:amUyp
         check_ef(verbünden.VSW2s:1mVi:3mVi:1mVc:3mVc/
            verbündet.ADJ0:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp:geUzp:deUyp:geUyp:aeMzp:aeMyp
            :amUzp:amUyp):
         check_ef(verbünden.VSW2s:1mVi:3mVi:1mVc:3mVc/verbündet.ADJ0:geMxp:geNxp:gmUxp:aeMxp:nmUzp
            :nmUyp:deUzp:geUzp:deUyp:geUyp:aeMzp:aeMyp:amUzp:amUyp)
            = verbünden.VSW2s:1mVi:3mVi:1mVc:3mVc verbündet.ADJ0:geMxp:geNxp:gmUxp:aeMxp
            :nmUzp:nmUyp:deUzp:geUzp:deUyp:geUyp:aeMzp:aeMyp:amUzp:amUyp
         check_zuInf(verbündeten):
         check_zuInf(verbündeten) =
         check_GenInf(verbündeten):
         check_GenInf(verbündeten) =
         check_nomInf(verbündeten):
         check_nomInf(verbündeten) =
         check_nomAdj(verbündeten):
         check_nomAdj(verbündeten) = verbündete.NA:geMx:geNx:gmUx:aeMx:nmUz:nmUy
            :deUz:geUz:deUy:geUy:aeMz:aeMy:amUz:amUy
```

look_up_ef(verbündeten) = verbünden.VSW2s:1mVi:3mVi:1mVc:3mVc
verbündet.ADJ0:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp:geUzp:deUyp:geUyp:aeMzp
:aeMyp:amUzp:amUyp
verbündete.NA:geMx:geNx:gmUx:aeMx:nmUz:nmUy:deUz:geUz:deUy:geUy:aeMz:aeMy:amUz:amUy
filter_gross(verbünden.VSW2s:1mVi:3mVi:1mVc:3mVc
verbündet.ADJ0:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp:geUzp:deUyp:geUyp:aeMzp:aeMyp
:amUzp:amUyp
verbündete.NA:geMx:geNx:gmUx:aeMx:nmUz:nmUy:deUz:geUz:deUy:geUy:aeMz:aeMy:amUz:amUy
filter_gross(verbünden.VSW2s:1mVi:3mVi:1mVc:3mVc
verbündet.ADJ0:geMxp:geNxp:gmUxp:aeMxp:nmUzp:nmUyp:deUzp:geUzp:deUyp:geUyp:aeMzp:aeMyp
:amUzp:amUyp
verbündete.NA:geMx:geNx:gmUx:aeMx:nmUz:nmUy:deUz:geUz:deUy:geUy:aeMz:aeMy:amUz:amUy)
= verbündete.NA:geMx:geNx:gmUx:aeMx:nmUz:nmUy:deUz:geUz:deUy:geUy:aeMz:aeMy:amUz:amUy
filter_ef(verbündete.NA:geMx:geNx:gmUx:aeMx:nmUz:nmUy:deUz:geUz:deUy:geUy:aeMz:aeMy:amUz:amUy,
1):
filter_ef(verbündete.NA:geMx:geNx:gmUx:aeMx:nmUz:nmUy:deUz:geUz:deUy:geUy:aeMz:aeMy:amUz:amUy,
1) = verbündete.NA:geMx:geNx:gmUx:aeMx:nmUz:nmUy:deUz:geUz:deUy:geUy:aeMz:aeMy:amUz:amUy
lemmatisiere_WB(verbündeten,WB Verbündeten 11 0 1 0 0 WB,1) =
[verbündeten]:verbündete.NA:geMx:geNx:gmUx:aeMx:nmUz:nmUy:deUz:geUz:deUy:geUy:aeMz:aeMy
:amUz:amUy
lemmatisiere(verbündeten@WB:Verbündeten:11:0:1:0:::0:WB,1) =
[verbündeten]:verbündete.NA:geMx:geNx:gmUx:aeMx:nmUz:nmUy:deUz:geUz:deUy:geUy:aeMz:aeMy
:amUz:amUy

lemmatisiere(,@SX:,:1:0:0:0:::0:WB):
...
lemmatisiere(,@SX:,:1:0:0:0:::0:WB) = [,]:SX(,).CC

lemmatisiere(der@WB:der:3:0:0:0:::1:SX):
...
lemmatisiere(der@WB:der:3:0:0:0:::1:SX) = [der]:d.DET4:gmM:neM:geF:deF

6 Literatur

Allén, S. (1967): Förhållandet mellan skrift och tal. In: Allén, S. , K.H. Dahlstedt u.a. (Hrsg.): Språk, språkvård och kommunikation. Lund. S.76ff.

Allén, S. (1978): Lexical Entry, Linguistic Sign and Lexical Data Base. In Proc. Coling 78. S.

Augst, G. (1975): Untersuchungen zum Morpheminventar der deutschen Gegenwartsprache. Tübingen: Narr.

Barton, G., R. Berwick & E. Ristad (1987): Computational Complexity of Natural Language. Cambridge/M. MIT Press.

Belica, C. (1994): WP2 - Lemmatizer. Deliverable D5, MLAP93-21 MECOLB. Institut für deutsche Sprache, Mannheim.

Bergenholtz, H. (1976): Zur Morphologie deutscher Substantive, Verben und Adjektive. Bonn.

Bergenholtz, H. (1985): Vom wissenschaftlichen Wörterbuch zum Lernerwörterbuch. In: Bergenholtz/Mugdan (1985): Lexikographie und Grammatik. Akten des Essener Kolloquiums zur Grammatik im Wörterbuch.

Bergenholtz, H. (1989): Probleme der Selektion im allgemeinen einsprachigen Wörterbuch. In: Steger, H. & H. Wiegand (Hrsg): Handbücher zur Sprach und Kommunikationswissenschaft. Bd. 5. Wörterbücher.

Bergenholtz, H. & J. Mugdan (1984): Grammatik im Wörterbuch von ja bis Jux. In: Studien zur neuhochdeutschen Lexikographie V. Hrsg von H.E. Wiegand. Hildesheim, Zürich, New York. S. 47-102

Bergenholtz, H. & B. Schaeder (1977): Die Wortarten des Deutschen. Stutttgart: Klett..

Boons, J.-P., A. Guillet & C. Leclère (1976): La structure des phrases simples en français: classes de constructions transitives. Rapport de Recherches du LADL No 6, Paris: Université Paris 7.

Brill, E. (1994): Some Advances in Transformation-Based Part of Speech Tagging. Erscheint in: Proc. of the 12th National Conference on Artificial Intelligence (AAAI-94).

Büttel, I., G. Niedermair, G. Thurmair & A. Wessel (1986): MARS: Morphologische Analyse für Retrieval-Systeme. In: Schwarz/Thurmair (1986).

Church, K. (1988): A Stocastic Parts Program and Noun Phrase Parser for Unrestricted Text. In: Proc. of the Second ACL Conference on Applied Natural Language Processing.

[CONDOR] (1975): CONDOR. Bericht der Siemens-Datenverarbeitung München.

Courtois, B. (1984): DELAS: Dictionnaire électronique du LADL pour les mots simples du français. Rapport Technique,Laboratoire d'Automatique Documentaire et Linguistique, Université Paris 7.

Courtois, B. (1989): Un système de dictionnaire électronique pour les mots simples du français. Report No. 9, Laboratoire d'Automatique Documentaire et Linguistique, Université Paris 7.

Courtois, B. & M. Silberztein (1989): Les dictionnaires électroniques DELAS et DELAC. Actes du colloque sur les langues romanes, Université Laval: Quebec.

Cutting, D., J. Kupiec, J. Pedersen & P. Sibun (1992): A Practical Part-of-Speech Tagger. In: Proc. of the 3rd Conference on Applied Natural Language Processing, ACL.

Dagan, I., F. Pereira & L. Lee (1994): Similarity-Based Estimation of Word Coocurrence Probabilities. In: Proc. of the 32nd Annual Meeting of the Association for Computational Linguistics.

DeRose, S.J. (1988): Grammatical Category Disambiguation by Statistical Optimization. In: Computational Linguistics 14/1. S. 31-39.

Dickmann, L., J. Heine, J. Klein, F. Oberhauser, H. Pirker & J. Simon (1993): Abschlußbericht zur Kodierung und Bewertung des SADAW-Lexikons. Saarbrücken.

Dietrich, R. (1973): Automatische Textwörterbücher. Studien zur maschinellen Lemmatisierung verbaler Wortformen des Deutschen. Tübingen, Niemeyer.

Dietrich, R. & W. Klein (1974): Computerlinguistik. Eine Einführung. Stuttgart, Urban.

Domenig, M. (1987): Entwurf eines dedizierten Datenbanksystems für Lexika: Problemanalyse und Software-Entwurf anhand eines Projektes für maschinelle Sprachübersetzung. Tübingen. Niemeyer.

Domenig, M., T. Domenig, D. Holz, A. Hsiung & S. Pedrazzini (1993): Werkzeuge zur Akquisition und Verwaltung von morphologischem und phradealem Wissen. In: Pütz/Haller (1993).

DUDEN-Redaktion (Hrsg.) (1986): DUDEN “Rechtschreibung der deutschen Sprache und der Fremdwörter. Mannheim/Wien/Zürich: Bibliographisches Institut.

Eggers, H. (Hrsg.) (1980): SALEM: Ein Verfahren zur Lemmatisierung deutscher Texte. Tübingen: Niemeyer.

Feldweg, H. (1993): Stochastische Wortartendisambiguierung für das Deutsche. Untersuchungen mit dem robusten System LIKELY. SfS-Report-08-93, Universität Tübingen.

Fleischer, W. & I. Barz (1992): Wortbildung der deutschen Gegenwartssprache. Tübingen: Niemeyer.

Francis, W. & H. Kucera (1982): Frequency Analysis of English Usage. Boston: Houghton Mifflin.

Gale, W. & K. Church (1991): A Program fot Aligning Sentences in Bilingual Corpora. In: Proc. of the 29th Annual Meeting of the ACL.

Garside, R., G. Leech & G. Sampson (Hrsg.) (1987): The Computational Analysis of English. Longman.

Giry-Schneider, J. (1978): Les nominalisations en français. L'opérateur "faire" dans le lexique. Genève: Droz.

Giry-Schneider, J. (1987): Les prédicats nominaux en français. Les phrases simples à verbe support. Genève: Droz..

Gonnet, G. (1984): Handbook of Algorithms and Data Structures. Addison-Wesley.

Gross, G. (1991): La forme d'un dictionnaire électronique. LADL-Report, Laboratoire d'Automatiuqe Documentaire et Linguistique, Université Paris 7.

Gross, M. (1986): Grammaire transformationelle du français: 2. Syntaxe du nom. Paris: Cantilène.

Gross, M.(1988): Sur les phrases figées complexes du français. In: Langue Française 77, 47-70. Paris: Larousse.

Gross, M. & D. Perrin (eds.) (1989): Electronic Dictionaries in Computational Linguistics, LITP Spring School on Theorertical Computer Science Saint-Pierre d'Oléron, France, May 1987 Proceedings. Berlin, Heidelberg, New York.

Gross, M. (1989): The Use of Finite Automata in the Lexical Representation of Natural Lnaguage. In: Gross, M. & D. Perrin (eds.) 1989.

Gross, M. (1990): La caractérisation des adverbes dans un lexique-grammaire. In: Langue Française 86, 90-102. Paris: Larousse.

Guenthner, F. & P. Maier (1994): Das CISLEX-Wörterbuchsystem. CIS-Bericht 94-76. CIS, Universität München.

Hanrieder, G. (1994): MORPH. Ein modulares und robustes Morphologieprogramm für das Deutsche in Common Lisp. In: LDV-Forum 11/1, 1994. S. 30-38.

Hausmann, F. (1989): Wörterbücher Dictionaries Dictionnaires. Ein linguitisches Handbuch zur Lexikographie. Berlin. DeGruyter.

Hausser, R. (1994a): LA-Morph. Ein linksassoziatives Morphologiesystem. Darstellung für die ersten Morpholympics 1994. In: LDV-Forum 11/1, 1994. S. 39-53.

Hausser, R. (1994b): The Coordinator's Final Report on the First Morpholympics. In: LDV-Forum 11/1, 1994. S. 54-64..

Heidolph, K., W. Fläming & W. Motsch (1984): Grundzüge einer deutschen Grammatik. Berlin: Akademie-Verlag.

Hellberg, S. (1972): Computerized Lemmatization without the Use of a Dictionary. In: Computers and the Humanities 6. S 209-212.s

Heß, K., J. Brustkern & W. Lenders (1983): Maschinenlesbare deutsche Wörterbücher: Dokumentation, Vergleich, Integration. Tübingen: Niemeyer.

Heyn, M. (1992): Zur Wiederverwendung maschinenlesbarer Wörterbücher: Eine computergestützte metalexikographische Studie am Beispiel der elektronischen Edition des "Oxford Advanced Learner's Dictionary of Current English". Tübingen: Niemeyer.

Hippelein, M. (1994): Probleme der morphosyntaktischen Annotation deutscher Textkorpora: Untersuchungen zu "irregulären" Wortformen. Magisterarbeit, Universität Stuttgart.

Kandler, G. & S. Winter (1992): Wortanalytisches Wörterbuch. Deutscher Wortschatz nach Sinn-Elementen in 10 Bänden. München.

Kay, M. & M Röscheisen (1988): Text-Translation Alignment. Techn. Report, Xerox Palo Alto Research Center.

Koskenniemi, K. (1983): Two-Level Morphology. A General Theory for Word-Form Recognition and Production. Dept. of General Linguistics, Univ. of Helsinki, Publ. No. 11.

Kühnhold, I., O. Putzer & H. Wellmann (1978): Deutsche Wortbildung. Das Adjektiv. Düsseldorf: Schwann.

Kühnhold, I. & H. Wellmann (1973): Deutsche Wortbildung. Das Verb. Düsseldorf: Schwann.

Labelle, J. (1983): Verbes supports et opérateurs dans les constructions en "avoir" à un ou deux compléments. In: Lingvisticae Investigationes 7:2. 237-260. Amsterdam/Philadelphia.

Langer, S. (1994): Kodierung von Fugenmorphemen bei N+N-Komposita. Abschlußarbeit zum Aufbaustudiengang Computerlinguistik. CIS, Universität München.

Leech, G., R. Garside & E. Atwell (1983): The Automatic Grammatical Tagging of the LOB Corpus. In: ICAME News 7, S. 13-33.

Lenders, W. (1989): Segmentierung in der Computerlinguistik. In: Handbuch Computerlinguistik.

Lenders, W. (1993): Tagging - Formen und Tools. In: Pütz/Haller (1993).

Liberman, M. (Hrsg.) (1991): Association for Computational Linguistics - Data Collection Initiative. CD-ROM I. Univ. of Pennsylvania.

Lingsoft Oy (1994): GERTWOL. Questionnaire for Morpholympics 1994. In: LDV-Forum 11/1, 1994. S. 17-29.

Mackensen, L. (1955): Deutsches Wörterbuch. Laupheim.

Marshall, I. (1987): Tag Selection Using Probabilistic Methods. In: Garside, R., G. Leech & G. Sampson (Hrsg.): The Computational Analysis of English. London/New York: Longman. 1987.

Meunier, A. (1981): Nominalisations d'adjectifs par verbes supports. Thèse de troisième cycle, LADL, Université Paris 7.

Müller, S. (1991): Zur Systematik morphologischer Paradigmen. Die deutschen Verben in der Computermorphologie. Magisterarbeit. Heidelberg,

Ott, N. (1992): Statistische Untersuchungen an einsprachigen Zeitungstexten. Technischer Bericht, IBM Deutschland, Heidelberg.

Petöfi, J. & J. Bredemeier (1977): Das Lexikon in der Grammatik. Die Grammatik im Lexikon. Papiere zur Textlinguistik, Bd 13. Hamburg.

Poirier, C. (1989): Les différents supports du dictionnaire: livre, microfiche, dictionnaire électronique. In: Steger, H. & H. Wiegand (Hrsg): Handbücher zur Sprach- und Kommunikationswissenschaft. Bd. 5. Wörterbücher.

Pütz, H.P. & J. Haller (Hg.) (1993): Sprachtechnologie: Methoden, Werkzeuge, Perspektiven. Hildesheim: Olms.

Rackow, U. (1992): On the Treatment of Compounds in Machine Translation. A Study. IWBS Report, IBM Deutschland.

Rackow, U., I. Dagan & U. Schwall (1992): Automatic Translation of Noun Compounds. In: Proceedings of Coling'92.

Revuz, D. (1991): Dictionnaires et lexiques, méthodes et algorithmes. Thèse de doctorat, CERIL, Université Paris 7.

Roche, E. (1991): Une représentation par automates des textes et des propriétés syntaxiques. Rapport L2/91 du Programme de Recherches Coordonnées "Informatique Linguistique". LADL, CERIL, Unversité Paris 7.

Rosengren, I. (1969): Wort und Wortform. in: Studia Linguistica XXIII, 1969, 103 ff.

Rosengren, I. (1972): Ein Frequenzwörterbuch der deutschen Zeitungssprache I und II. Lunder germanistische Forschungen 41, Lund 1972.

Schaeder, B. & K. Schweisthal (1968/70): Untersuchungen zu den deutschen Präpositionen. Manuskripte der Forschungsgruppe LIMAS. Die deutschen Präpositionen Bd. I (1968) und Bd. II (1970). Bonn.

Schaeder, B. (1981): Lexikographie als Theorie und Praxis. Tübingen.

Schicht, G. (1994a): Bindestrichwörter. Abschlußarbeit des Aufbaustudiengangs Computerlinguistik. CIS, Universität München.

Schicht, G. (1994b): Probleme der Satzende-Erkennung. CIS-Bericht-94-81 CIS, Universität München.

Schnelle, H., C. Kunze, U. Heid, M. Heyn (1990): Lexical and terminological Resources for German and for the Scandinavian Languages. Internes Papier der Eurotra-7 Study: Beitrag zur DOC-3, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart.

Schoenherr, L. (1991): Quantitative Analysis of German Compounds (QUAGC). Interner Bericht, IBM Heidelberg.

Schott, G. (1972): Automatic Analysis of Inflectional Morphemes in German Nouns. An Algorithm for Automatic Indexing. Acta Informatica I.

Schott, G. (1974): Datenverarbeitung und Großschreibung. In: Digeser, A. (Hrsg.): Probleme der Rechtschreibreform. Göttingen, Vandenhoeck. S. 67-95.

Schott, G. (1978): Automatische Deflexion deutscher Wörter unter Verwendung eines Minimalwörterbuchs. In: Sprache und Datenverarbeitung 1/1978. S. 62-77.

Schulze, W. & G. Willée (1983): Noch eine Variation über das Thema: "LEMMA". In: Sprache und Datenverarbeitung 1/1983. S. 40-46.

Schwartz, R. (1993): Learning Perl. Sebastopol/Ca.: O’Reilly.

Schwarz, C. & G. Thurmair (Hrsg.) (1986): Informationslinguistische Texterschließung. Hildesheim: Olms.

Seewald, U. (1993): Automatische Wortformerkennumg im Deutschen, Analyseverfahren und Systeme. In: LDV-Forum 10/2, 1993. S. 5-16.

Silberztein, M. (1989): Dictionnaires électroniques et reconnaissance lexicale automatique. Thèse de doctorat, Unversité Paris 7.

Stepanowa, M. & W. Fleischer (1985): Grundzüge der deutschen Wortbildung. Leipzig: BEB Bibiliographisches Institut.

Strauß, G. (1989): Angabe traditioneller Wortarten oder Beschreibung nach funktionalen Wortklassen im allgemeinen einsprachigen Wörterbuch? In: Steger, H. & H. Wiegand (Hrsg): Handbücher zur Sprach- und Kommunikationswissenschaft. Bd. 5. Wörterbücher.

Tapainen, P. & A. Voutilainen (1994): Tagging accurately - Don't guess if you know. Cmp-lg/9408009.

Thurmair, G. (1986): Eine maschinelle morphologische Analyse des Deutschen. In: Schwarz/Thurmair (1986).

Wahrig, G. (1975): Deutsches Wörterbuch. Gütersloh 1966. (Neuauflage 1975).

Wahrig, G. (1968): Neue Wege in der Wörterbucharbeit. Hamburg.

Wahrig, G. (1973): Anleitung zur grammatisch-semantischen Beschreibung lexikalischer Einheiten. Tübingen.

Wahrig, G. (1978): dtv-Wörterbuch der deutschen Sprache. München 1978.

Wall, L. & R. Schwartz (1993): Programmieren in perl. München/Wien: Hanser.

Wallner, M. (1994): Untersuchung von Text-Alignment im Rahmen computergestützter Textübersetzung auf Basis des Kay-Röscheisen-Algorithmus. Diplomarbeit Fachhochschule München, Fachbereich Informatik.

Weber, H.J. (1976): Automatische Lemmatisierung - Zielsetzung und Arbeitsweise eines linguistischen Identifikationsverfahrens. In: Linguistische Berichte 44/1976. S. 30-47.

Weber, N. (1990): Maschinelle Lexikographie und Wortbildungsstrukturen. Tübingen: Niemeyer.

Wellmann, H. (1975): Deutsche Wortbildung. Das Substantiv. Düsseldorf: Schwann.

Wiegand, H.E. (1983): Was ist eigentlich ein Lemma? Ein Beitrag zur Theorie der lexikographischen Beschreibung. In: Studien zur Neuhochdeutschen Lexikographie III, 1983.

Willée, G. (1977): Ein Verfahren zur automatischen Verbformenanalyse. In: Sprache und Datenverarbeitung 1/1977. S.160-162.

Willée, G. (1979): LEMMA - Ein Programmsystem zur automatischen Lemmatisierung deutscher Wortformen. In: Sprache und Datenverarbeitung 1/1979. S. 45-60.

Willée, G. (1993): Erfahrungen mit morphologischem Tagging am Beispiel des LIMAS-Korpus. In: Pütz/Haller (1993).

Wolski, W. (1989): Das Lemma und die verschiedenen Lemmatypen. In: Hausmann et al. (1989). S. 360-371.

Wothke, K. (1993): Statistisch basiertes Wortklassentagging. In: Pütz/Haller (1993).

Wothke, K., I. Weck-Ulm, J. Heinecke, O. Mertineit & J. Pachunke (1993): Statistically Based Automatic Tagging of German Text Corpora with Parts-of-Speech - Some Experiments. Technical Report 75.93.02, IBM Deutschland, Heidelberg.

Zehnder C.A. (1983): Informationssysteme und Datenbanken. Verlag der Fachvereine an den Schweiz. Hochschulen und Techniken, Zürich, 2.

Zimmermann, P. (1993): Epelle: Un logiciel de détection de fautes d’orthographe. INRIA-Rapport No. 1030. Institut national de recherche en informatique et en automatique.