

# Grundlagen zu Natural Language Processing und Analyse von GATE als NLP-Tool

*Aaron Schul, Felix Ritter*

## Contents

1	Einleitung . . . . .	3
2	Grundlagen von Natural Language Processing . . . . .	4
2.1	Einführung . . . . .	4
2.1.1	Motivation und Herkunft . . . . .	4
2.1.2	NLP als Annäherung an natürlichsprachliche Probleme . . . . .	5
2.1.3	Definition . . . . .	7
2.1.4	Ziel und funktionale Einordnung . . . . .	7
2.2	Linguistische Analyse . . . . .	7
2.3	Allgemeine NLP Architektur . . . . .	8
2.4	Morphologie in NLP . . . . .	10
2.4.1	Lexika in der Morphologie . . . . .	11
2.4.2	Buchstabierregeln mit endlichen Zustandswandlern . . . . .	12
2.4.3	Endliche Zustandswandler zur Umsetzung von morphologischen Regeln . . . . .	13
2.5	Part of speech tagging und Wortvorhersage . . . . .	15
2.5.1	N-gram Modelle zur Sprachvorhersage und -modellierung . . . . .	16
2.5.2	Stochastisches POS-tagging . . . . .	18
2.6	Parsen und Generieren . . . . .	20
2.6.1	Generative Grammatik . . . . .	20
2.6.2	Kontextfreie Grammatiken . . . . .	21
2.7	Parsen mit constraint-basierten Grammatiken . . . . .	22
2.7.1	Beispiel zu einfacher FS Grammatik . . . . .	22

## List of Tables

1	Wahrscheinlichkeitstabelle Bigram zur Wortvorhersage . . . . .	17
2	Annotationen gemäß [CLW7] (Auszug) . . . . .	18

## 1 Einleitung

Mit voranschreitender Globalisierung und immer größeren Mengen an übertragenen Informationen steigt auch die Menge an zu verarbeitender natürlicher Sprache. Während dies bisher manuell durch den Menschen geschah, kommt allmählich aufgrund der schier unendlichen Menge von Daten vor allem im unternehmerischen Kontext und im Internet die Notwendigkeit auf, die natürliche Sprache automatisiert durch Computer verarbeiten zu lassen.

Dies bezeichnet man als Natural Language Processing (kurz NLP). Es hilft dabei, Texte beispielsweise nach Schlagworten zu durchsuchen, strukturell zu analysieren, Muster zu erkennen und teilweise sogar die Bedeutung von Geschriebenem zu verstehen und diese Semantiken darzustellen.

Mit dem Aufkommen von NLP steigt auch das Interesse an NLP-Tools, also an Programmen, welche in der Lage sind, die für NLP notwendigen Funktionalitäten übersichtlich und praktikabel bereitzustellen. Eines der bekanntesten NLP-Tools ist das von der University Of Stanford entwickelte GATE, welches eine Vielzahl von Anwendungsbereichen im NLP abdeckt.

Das Ziel dieser Arbeit ist es, eine Anforderungsanalyse für solche Tools durchzuführen und diese dann mit dem GATE-Tool abzugleichen. Somit bietet sich die Möglichkeit GATE zu analysieren und daraufhin kriterienbasiert zu evaluieren. Funktionale wie nicht-funktionale Anforderungen werden definiert.

Zu diesem Zweck werden zunächst ausführlich die Grundlagen zu NLP und dessen Tools erläutert. Die Teilschritte der Textverarbeitung werden theoretisch wie praktisch anhand von Implementierungsbeispielen erläutert. Daraufhin können mit Hilfe von Use Cases Anforderungen an diese Tools erhoben werden, auf denen Analyse und Evaluation des GATE-Tools beruhen sollen. Abgeschlossen wird die Arbeit von einer kurzen Zusammenfassung der gewonnenen Erkenntnisse, gefolgt von einem Ausblick auf mögliche zukünftige Arbeiten und Entwicklungen für NLP-relevante Domänen.

## 2 Grundlagen von Natural Language Processing

### 2.1 Einführung

Der folgende Abschnitt befasst sich mit den Grundlagen zu Natural Language Processing (im folgenden kurz NLP). Dazu gehört neben dessen Herkunft bzw. Motivation eine inhaltliche Wissensgrundlage, welche für den weiteren Verlauf der Ausarbeitung relevant ist.

#### 2.1.1 Motivation und Herkunft

Ein Großteil der menschlichen Kommunikation findet durch natürliche Sprache statt. Dies ist bereits seit Jahrtausenden so, jedoch hat sich die Übertragung dieser Sprache mit der Zeit verändert und weiterentwickelt.

Damals lediglich von Mund zu Mund übertragen war der erste große Schritt die Entwicklung der Schrift. Ob mit Hieroglyphen, Alphabeten oder anderen Zeichen, war man in nun in der Lage natürliche Sprache über lange Zeit und/oder weite Strecken zu vermitteln. Dies erwies sich als sehr Vorteilhaft und so setzte sich die Entwicklung fort, bis man über das Morsen und das Telefon schließlich die elektronische Nachricht erfand. Im Rahmen der voranschreitenden Digitalisierung und Globalsierung stieg die Menge an übertragener natürlicher Sprache weiter rasant an, sodass heutzutage ein Leben ohne beispielsweise die E-Mail unvorstellbar scheint. Obgleich die Übertragung durch kabelgebundene oder drahtlose Kommunikation weitgehend automatisiert ist, erfolgt die Auswertung des Inhalts weiterhin manuell durch menschliche Empfänger.

Die Daten selbst sind jedoch inzwischen kaum noch nur durch den Menschen effizient zu verarbeiten, sodass man nach einer neuen Möglichkeit sucht, dem Menschen diese Arbeit zu erleichtern, oder sogar abzunehmen. Mit diese Aufgabe beschäftigt sich NLP.

Bereits in den 1960er Jahren versuchte man durch Sprachcomputer bzw. Chatbots wie *ELIZA* (Weizenbaum 1966) die Mensch-Maschine-Kommunikation umzusetzen, indem die Textnachricht eines Benutzers automatisch durch einen Computer verarbeitet wurde und eine plausible Antwort gegeben wurde, damals jedoch noch ohne echte Wissensbasis. Aus heutiger Sicht kann man ELIZA somit keine Möglichkeit zur "intelligenten" Kommunikation attestieren, da scheinbar der Sinn hinter den Nutzereingaben nicht gänzlich erfasst wird. Heutige Entwicklungen im Bereich Sprachassistenzsysteme sind alltäglicher

Begleiter jedes Smartphone-Nutzers geworden; sie analysieren die Spracheingaben mithilfe von NLP-Techniken bezüglich ihrer Bedeutung in Echtzeit über Clouds.

### 2.1.2 NLP als Annäherung an natürlichsprachliche Probleme

Bereits bei der Mensch-Mensch-Kommunikation sind Verständigungsprobleme vorhanden und Teil der Kommunikation selbst. Die Intention einer Aussage ist etwa nicht nur abhängig von dem, was tatsächlich gesagt wird, sondern auch etwa von Gestik, Mimik und der Situation, in der kommuniziert wird. Bei der Untersuchung von geschriebenem Text in natürlicher Sprache treten diese Betrachtungen jedoch in den Hintergrund, da sich auf den Inhalt konzentriert wird und mitunter die Situation des Autoren unklar oder irrelevant ist. Wissen über die Problemdomäne ist jedoch zwingend für das Verständnis spezifischer Fachbegriffe von Nöten, daher müssen auch automatische NLP-Tools diese einbeziehen können. Zudem gibt es oft Veränderungen der Bedeutung von Sprache die stark kontextabhängig sind, wie zum Beispiel Sarkasmus oder Ironie. Hier soll nicht weiter auf Aspekte der Kommunikationswissenschaften eingegangen werden, jedoch ist beispielsweise die Ambiguität einer Aussage ganz alltäglich und hat gleichsam verschiedene Auswirkungen. Solche Mehrdeutigkeiten werden durch syntaktische und semantische Fehlinterpretationen verursacht.

Wo sich Menschen im Zweifel auf Erfahrungen und spezifische Fachkenntnisse verlassen oder bei ihrem Kommunikationspartner nachfragen können, müssen sich Computer allein auf die vorliegenden Dokumente in Schriftform verlassen; sie wissen nichts über den Autoren oder dessen Blickwinkel. Wie später gezeigt wird, kann jedoch etwa Fachwissen durch domänenspezifische Wissensbasen simuliert werden.

Die Frage nach einer intelligenten, algorithmen- und / oder wissensbasierten Auswertung durch Computer stellt sich im Angesicht der aufgezeigten Besonderheiten natürlicher Sprache. Es gilt also für NLP nicht nur die Probleme zu lösen, die generell mit präziser Automatisierung verbunden sind, sondern zusätzlich so zuverlässig wie möglich die oben genannten und der Sprache inhärenten Komplikationen zu bewältigen. Bereits einfache Anfragen können computergestützte Frage-Antwort-Systeme wie SQL-Datenbanken an ihre Grenzen bringen, wie die verschiedenen Formen von Mehrdeutigkeiten zeigen können:

Schon das Wort *überblicken* kann das Übersehen von etwas, ebenso wie

das im Blick haben (Gegenteil) bedeuten.

Der Satz *Die Betrachtung des Studenten* kann etwa als Student der (etwas) betrachtet, oder als ein Student, der (von jemand anderem) betrachtet wird, verstanden werden. Sätze mit solch einem Satzbau (bestehend aus nominalisiertem Verb, Bezugswort und Substantiv) verursachen unwillkürlich zwei mögliche Deutungen aufgrund der unklaren Syntax.

Die Frage *Verkaufen Sie Handys und Computer von Samsung?* kann als Frage nach allgemeinen Handys und nur Computern, spezifisch von Samsung oder nach Handys und Computern gleichermaßen nur von Samsung verstanden werden. Auch hier entsteht Unklarheit durch zwei mögliche Bezüge des einschränkenden Relativsatzes. Ob eine Aufzählung oder ein Unterschied (zwischen *von Samsung* und *nicht von Samsung* gemacht wird, ist syntaktisch unklar.)

*Wie schnell ist der Bus?* und *Wie schnell ist der Bus da?* mögen zunächst ähnlich aussehen, fragen jedoch inhaltlich anders und haben nichts miteinander zu tun. Außerdem könnte der zweite Satz kontextabhängig etwa nach der Fahrzeit bis zum Ziel oder nach der Ankunftszeit, bis an der man einsteigen kann, fragen. Die vage Formulierung von Sätzen stellt bei der Frage nach deren Bedeutung eine Herausforderung dar, durch Untersuchung des Kontextes der Frage kann hier jedoch meist recht treffsicher geantwortet werden.

Abgesehen von diesen syntaktischen Uneindeutigkeiten muss NLP auch Hürden der inhaltlichen Mehrdeutigkeit bzw. Semantik bewältigen. Aktuell beschäftigt man sich zum Beispiel mit der Aufgabe zu erkennen, ob ein Kommentar zu einem Video der Webseite Youtube eher positiv oder eher negativ gesinnt ist. Was hier für den Menschen schon beim ersten Lesen sofort erkennbar wird, stellt für den Computer ein ernstzunehmendes Problem dar. Tonalität und Stimmung des Kommentars, der unter einem Video als Reaktion entsteht, ist mitunter nicht direkt aus dem audiovisuellen Eindruck aus dem Video ersichtlich. Solche scheinbar simplen Probleme sind für den Menschen scheinbar simpel zu lösen. Wie oben erwähnt existiert NLP bereits seit ca. Mitte des 20. Jahrhunderts, jedoch stellt die Behandlung natürlichsprachlicher Probleme auch heute noch eine Herausforderung bei der Implementierung dar. Wenn Systeme die oben genannte Beispiele nicht explizit ausschließen, muss die Verarbeitungslogik besondere Rücksicht auf Uneindeutigkeiten nehmen.

### 2.1.3 Definition

NLP ist grob definiert als die automatische oder halb-automatische Verarbeitung von natürlicher Sprache. Manche schließen aus dieser Definition die Erfassung der Sprache aus, da diese nicht Teil der eigentlichen Verarbeitung ist. (Der Einfachheit halber wird im weiteren Verlauf davon ausgegangen, dass die Sprache als digitale Textdatei vorliegt, wenn nicht explizit anders angegeben. Es kann sich jedoch auch um gesprochenes Wort oder Gesten handeln)

NLP überschneidet sich mit vielen wichtigen Bereichen der Wissenschaft. Hauptsächlich sind dies Linguistik und Informatik, allerdings fließen auch Psychologie, Philosophie und Mathematik bzw. Logik stark mit ein.

### 2.1.4 Ziel und funktionale Einordnung

Das Ziel von NLP ist die Extraktion von Informationen aus gegebenen Texten, also aus einem natürlichsprachigen Input einen Wissensoutput über den Inhalt des Dokumentes zu generieren. Dies geschieht praktisch mithilfe von NLP-Tools wie GATE.

Dazu ist, wie im nächsten Abschnitt beschrieben, die linguistische Analyse der Dokumente der Sprachstruktur in Teilschritten erforderlich. Auf Basis der syntaktischen Analyse kann dann eine semantische Analyse durch Einsatz sogenannter Ontologien (Sammlung von Wissen aus einer Domäne) erfolgen. Letztendlich kann somit tatsächlich Wissen über die Bedeutung des Dokumenteninhalts gewonnen werden.

## 2.2 Linguistische Analyse

Der folgende Abschnitt beschäftigt sich mit der Analyse der sprachlichen Struktur eines Dokumentes. Am Anfang von NLP steht die linguistische Analyse des Textes etwa bezüglich Satzstruktur und Differenzierung von Wörtern aus der Sprache und Eigennamen. Das Ziel ist die sogenannte Annotation der Textbausteine bezüglich ihrer Bedeutung. Sachzusammenhänge und Relationen von Wörtern sollen dargestellt werden.

NLP kann, wie später in dieser Arbeit beschrieben, als Zusammenfassung von 6 wesentlichen Teilbereichen verstanden werden. Begonnen bei der einfachen Wortanalyse, werden später Bedeutungen und Verknüpfungen zur Anwendbarkeit der Aussage eines Satzes oder Wortes in der realen Welt

analysiert. Die Linguistik befasst sich seit jeher mit der Untersuchung von natürlicher Sprache im Hinblick auf die dahinterliegenden Konstrukte. NLP spaltet bei der Analyse der Sprachbestandteile weiter auf, etwa müssen die folgenden Teilbereiche, die daraus bekannt sind, bei der natürlichsprachigen Analyse identifiziert werden:

1. Morphologische Analyse - Die Zerlegung von Wörtern bezüglich ihrer Struktur. Die Komposition aus Präfix, Wortstamm und Suffix von Wörtern wird erkannt, um Fall, Tempus, Numerus etc. des Wortes zu bestimmen. Im Englischen zeigt so die Endung -ed bei Verben die Vergangenheitsform an. Dabei ist zu beachten, dass Mehrdeutigkeiten entstehen können.
2. Syntax - Zuordnung von Bedeutungen anhand sprachlicher Phrasen. Jede Sprache besitzt syntaktische Regelungen bezüglich der auftauchenden Wörter; im Deutschen folgt etwa auf einen Artikel stets ein Nomen oder bestimmte Signalwörter geben Satztyp und Tempus an. Auf Basis dieses Wissens können formale und strukturelle Analysen der Textbestandteile erfolgen.
3. Semantiken - Die Identifikation einer Bedeutung von Sätzen und Begriffen. Die Semantik eines Textabschnittes wird häufig als "Logik" bezeichnet und fragt nach dessen Bedeutung bzw. Thema. Bedeutungen können anhand von Kompositionen einzelner Wörter und Sätze auf Basis der semantischen Analyse erkannt werden.
4. Kontext - Darstellung von Sachzusammenhängen aus der Vereinigung von ähnlichen Bedeutungen. Textbausteine, die sich mit dem gleichen Thema beschäftigen, werden dem gleichen Sachzusammenhang zugeordnet und als Teil dessen annotiert.

## 2.3 Allgemeine NLP Architektur

Die Verarbeitung eines Dokumentes mittels NLP erfolgt nacheinander in einzelnen Schritten, die jeweils einen Bereich oder Teilbereich der Linguistik des Textes abbilden. Die Analyse erfolgt dabei nacheinander und wird immer feingranularer, begonnen bei einfachen grammatikalischen Analysen bis hin zu komplexen Wissensbasierten verfahren. Die folgenden Schritte stellen laut EINIGEN QUELLEN eine typische komponentenweise Architektur dar.



Es ist anzumerken, dass auch mehrere Schritte gemeinsam in einem Schritt durchgeführt oder gar ganz entfallen können, abhängig von der Umsetzung anderer Stufen:

1. Input Processing - Erkennung der Dokumentensprache und Normalisierung des Textes. Im ersten Schritt geht es um das korrekte Format des Eingabedokumentes für die Verarbeitung. Dieser Vorgang stellt jedoch noch keine Analyse dar, sondern dient lediglich der Vorbereitung.
2. Morphologische Analyse - Die meisten Sprachen sind im Bezug auf ihre Grammatik und syntaktische Struktur der Wörter in Systemen abbildbar, etwa durch Modellierung mittels Automaten zur Erzeugung der Worte. Auch können Vokabeln in Wörterbüchern/Lexika gesammelt und Regeln auf ihnen formuliert und gespeichert werden.
3. Part-of-Speech-Tagging - Die einzelnen Wörter eines Satzes werden im Hinblick auf den Sach- oder Satzzusammenhang, wie auch auf die Stellung der Stellung im Satz hin analysiert. Basierend auf Kontext und/oder Erfahrungswerten bzw. Heuristiken können die Wörter korrekt erfasst und deren Fall getaggt werden. Subjekte, Prädikate usw. werden identifiziert. Dazu ist eine Wissensbasis mit Trainingsdaten und die Einordnung des Kontextes darin von Nöten.
4. Parsing - Die Ergebnisse der der vorherigen Schritte werden weiter verarbeitet und in ein standardisiertes Format gebracht. Syntaktische Zusammenhänge, wie etwa das Zusammenfassen von Wörtern zu einer gemeinsamen Bedeutungsphrase und das Zuordnen von Verben zu einem Nomen können nach dem Parsing dargestellt werden.
5. Disambiguation - Die Entfernung von Mehrdeutigkeiten auf Basis der Ergebnisse des Parsings stellt einen entscheidenden Schritt für die semantischen Analysen dar. Nur wenn die Aussage und der Kontext eines Satzes klar erkennbar sind, kann dessen Semantik gezielt abgeleitet werden.
6. Context Module - Textbausteine, deren Interpretation semantisch auf dem Kontext anderer Sätze oder Wörtern beruhen, können erfasst werden. Bei Anaphern ist etwa das Verständnis des aktuellen Kontext abhängig von einer vorherigen Deutung.

7. Text Planning - Die Sachzusammenhänge und Semantiken, die aus dem zugrundeliegenden Text extrahiert wurden, werden für die Darstellung im fertigen Text definiert. Es wird festgelegt, welche der Bedeutungen übertragen und vermittelt werden sollen.
8. Tactical Generation - Die Bedeutungen werden in Form konkreter Zeichenketten generiert, die die gewünschte Bedeutung enthalten. Diese Textbausteine basieren häufig direkt auf den Ergebnissen des vorherigen Parsings, da dort schon Bedeutungen zusammengefasst werden können.
9. Morphological Generation - Die erzeugten Sätze werden morphologisch an die Rahmenbedingungen der verwendeten Sprache angepasst. Grammatiken und Regeln der Satzkonstruktion werden auf die erzeugten Wörter angewendet, sodass ein lesbarer und nachvollziehbarer Text entsteht.

## 2.4 Morphologie in NLP

Der folgende Abschnitt befasst sich genauer mit der morphologischen Analyse von Wörtern und den dazu verwendeten Zustandswandlern. Da zu der englischen Sprache bereits die meisten Fortschritte erzielt wurden und sie verhältnismäßig wenig komplex im Vergleich zu anderen Sprachen ist, werden im Folgenden viele der Beispiele auf Englisch behandelt.

Wie bereits erwähnt befasst sich die Morphologie mit der Struktur, Zusammensetzung und Korrektheit einzelner Wörter. Jedes Wort der englischen Sprache besteht aus mindestens einem Wortstamm und beliebig vielen Affixen. Affixe sind unterteilt in Präfixe und Suffixe, abhängig davon, ob sie vor oder hinter dem Wortstamm auftauchen. Es gibt Affixe, welche auch in der Mitte des Wortes eingeordnet werden. Diese existieren jedoch nicht in der englischen Sprache und sind daher künftig zu vernachlässigen.

Man betrachte beispielsweise das Wort "believe" von dem englischen Verb "to believe" (zu Deutsch: "glauben"). Hängt man das Suffix "-able" an so erhält man das Adjektiv "believable" ("glaubhaft"), fügt man nun das Präfix "un-" hinzu erhält man die Negation "unbelievable" ("unglaubhaft"). Die morphologische Analyse erkennt dies nicht nur als ein Wort, sondern als Präfix-Stamm-Suffix-Konstrukt.

Es wird unterschieden zwischen ableitender und flexionaler Morphologie,

wobei die Grenze jedoch nicht ganz einfach zu setzen ist. Affixe wie "anti-", "re-" und das oben verwendete "un-" gelten als ableitend und können mehrfach und sogar rekursiv zu Stämmen auftauchen. Sie fügen in der Regel eine Zusatzinformation an den Stamm an, wie z.B. Negation. Es kann auch, wie in dem Beispiel oben demonstriert, zur Änderung der Wortart kommen. Flexionale Affixe werden normalerweise wegen grammatikalischer Regeln wie Pluralisierung hinzugefügt. Hierzu gehören das Plural-"s" und die Endung "-ed" in der Vergangenheitsform.

Die morphologische Analyse ist üblicherweise eine Vorarbeit für das Parsen von Texten, welches in Abschnitt 2.6 genauer beschrieben ist. Dafür sind jedoch genauere syntaktische und semantische Informationen nötig. So würde beispielsweise das Wort "fishing" eindeutig durch die "-ing"-Endung als Verb in bestimmter Form erkannt, während bei dem Wort "fish" jedoch eine Mehrdeutigkeit zwischen dem Verb "fischen" und dem Nomen "Fisch" die Verarbeitung verkomplizieren würde. Dies wird durch das sogenannte Part-of-Speech-Tagging gelöst, welches in Abschnitt 2.5 behandelt wird.

### 2.4.1 Lexika in der Morphologie

Für eine präzise morphologische Analyse bedarf es 3 verschiedenen Lexika um mögliche Interpretationen zu speichern:

1. Liste von Affixen (zusammen mit den Information, die damit verbunden sind, z.B. Negation)
2. Liste von unregelmäßigen Wörtern (zusammen mit den Information, die damit verbunden sind, z.B. "went": simple past go)
3. Liste von Wortstämmen (zusammen mit syntaktischer Kategorie, z.B. "believe": verb)

Die Affix-Liste könnte in einen Präfix- und einen Suffix-Teil getrennt werden, welche jeweils den Affix und die oben beschriebenen Information als formatierten Text enthalten, in etwa wie folgt (Suffix-Teil):

```
ed PAST_VERB
ed PSP_VERB
s PLURAL_NOUN
```

PAST\_VERB, PSP\_VERB und PLURAL\_NOUN beinhalten die durch den jeweiligen Suffix hinzugefügten Informationen über das Wort.

Ähnlich verhält es sich mit der Auflistung der unregelmäßigen Formen. Der Affix wird hier durch die unregelmäßige Form ersetzt und es ist zu beachten, dass es nötig ist zu den in 1. vorhandenen Informationen auch den Referenz-Stamm anzugeben, in etwa wie folgt:

began PAST\_VERB begin  
 begun PSP\_VERB begin

"Began" und "begun" sind hier jeweils die "PAST\_VERB"- und "PSP\_VERB"(past participle)-Form des Referenzstammes "begin".

Um unnötige Fehler zu vermeiden ist es ratsam zusätzlich ein Lexikon mit Wortstämmen zu verwenden, welches die Wortstämme und eine grammatikalische Einordnung beinhaltet. Dies hat hauptsächlich zwei Gründe:

1. Die Form ist wichtig im Zusammenhang mit vielen syntaktischen (näher erläutert in Abschnitt 2.5) und morphologischen Regeln. Die "able"-Anhangsregel, kann beispielsweise nur bei Verben verwendet werden, welche dann zu einem Adjektiv werden(diese Eigenschaft würde in der Affix-Liste beschrieben).
2. Die möglichen Wortstämme überprüfen zu können ist wichtig, da somit viele Mehrdeutigkeiten ausgeschlossen werden können. So könnte "bus" z.B. als "bu<sup>h</sup>s" (Plural von "bu") interpretiert werden. Dies ist jedoch nicht möglich wenn kein Eintrag zu dem Wortstamm "bu" existiert.

### 2.4.2 Buchstabierregeln mit endlichen Zustandswandlern

In der Morphologie gibt es eine Vielzahl von Regeln die bei dem Unterteilen, Interpretieren und Zusammensetzen von Wörtern zu beachten sind. Wenn man das "believe"-Beispiel aus Abschnitt 2.4 erneut betrachtet, fällt auf, dass bei dem hinzufügen des Suffixes "-able" der letzte Buchstabe des Wortstammes entfällt. Dies ist natürlich kein Tippfehler, sondern beruht auf einer der besagten Regeln. Die oben verwendete Regel lässt sich wie folgt darstellen: ===== In der Morphologie gibt es eine Vielzahl von Regeln die bei dem Unterteilen, Interpretieren und Zusammensetzen von Wörtern zu beachten sind. Wenn man das "believe"-Beispiel aus dem obigen Abschnitt erneut betrachtet, fällt auf, dass bei dem hinzufügen des Suffixes "-

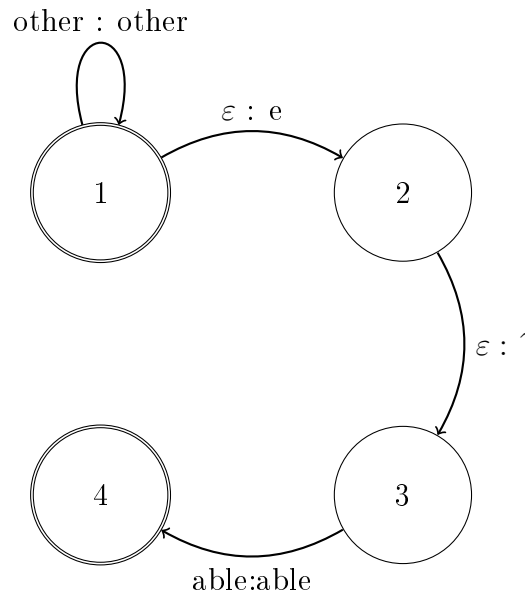
"able" der letzte Buchstabe des Wortstammes entfällt. Dies ist natürlich kein Tippfehler, sondern beruht auf einer der besagten Regeln. Die oben verwendete Regel lässt sich wie folgt darstellen:  $e \rightarrow \varepsilon\_ \text{able}$

Das  $e$  ist in dem Falle der in Frage stehende Buchstabe, welcher, symbolisiert durch den Pfeil, in das  $\varepsilon$  (das leere Wort) gewandelt wird, falls ein Affix angehängt wird ( $\wedge$ ) und es an dieser Stelle ( $\_$ ) steht, also vor dem "able".

Solche Regeln lassen sich zu morphologischen Zwecken gut mit endlichen Zustandswandlern (finite state transducers) realisieren, was im Folgenden demonstriert werden soll.

### 2.4.3 Endliche Zustandswandler zur Umsetzung von morphologischen Regeln

Ein endlicher Zustandswandler ist ein Graph aus einer endlichen Anzahl von Knoten (Zuständen), welche durch Kanten verbunden sind. Knoten sind unterteilt in einen Start-Knoten, eine beliebige Anzahl an normalen Knoten und mindestens einen Ziel-Knoten. Der Start-Knoten als Input das zu verarbeitende Wort, welches nun in eine Abbildung der Aufbaustruktur des Wortes übertragen werden soll. Das Erreichen eines Ziel-Knotens heißt, dass das Wort korrekt ist und akzeptiert werden kann. Die Kanten repräsentieren die Abbildung einer Zeichenkette auf eine andere. Dies funktioniert bidirektional, wodurch sie sowohl zum Erstellen, als auch zum Auflösen von Wörtern verwendet werden können. Die folgende Abbildung zeigt den endlichen Zustandswandler, der die Umsetzung der besagten "able"-Anhangs-Regel ermöglicht:



Der oben abgebildete Zustandswandler ist in der Lage die beschriebene Regel umzusetzen wie folgt: In Zustand 1 erfolgt der Input des eingelesenen Wortes, also in diesem Falle "believable". Über die Schleife mit "other:other" ist die Abbildung aller nicht speziell angegebenen Zeichenfolgen auf sich selbst beschrieben. Es kann jederzeit das leere Wort auf ein "e" übertragen werden, dies geschieht jedoch nur, falls auch ein "able" folgt. In dem Fall wird zuvor noch aus dem leeren Wort ein Affix-Zeichen (^) generiert (Es ist anzumerken, dass der Zustandswandler nicht deterministisch ist, sich jedoch über die Potenzmengenkonstruktion immer ein äquivalenter deterministischer Zustandswandler erstellen ließe). So wird also das Wort "believable" durch diesen Automaten als "believe^able" Interpretiert, und somit das "able"-Affix erkannt. Andersherum ließe sich aus dem Wortaufbau "believe^able" das korrekte Wort "believable" generieren.

«««< HEAD Üblicherweise wird für jede Regel ein eigener Zustandswandler erstellt, welche dann bei der Verarbeitung von Wörtern parallel ablaufen. Dies ist gut für die Performanz, birgt allerdings Probleme, wenn man mit komplexen Wörtern arbeitet, bei denen mehrere Regeln gleichzeitig anzuwenden sind, bei denen verschieden valide Ergebnisse erzielt werden können. So wäre es z.B. problematisch zwischen den Wörtern "un^ion^ise^ed" (Chemie) und "union^ise^ed" (gewerkschaftlich organisiert) zu unterscheiden.

## 2.5 Part of speech tagging und Wortvorhersage

Dieses Kapitel befasst sich mit der Transformation der Ergebnisse der morphologischen Analyse einzelner Wörter in eine sinnvolle Kategorisierung der Wörter im Sachzusammenhang aus mehreren Wörtern. Part-of-speech tagging klassifiziert einzelne Worte eines Textes so, dass erfasst wird, welchen Teil der Sprache sie ausmachen. Die einzelnen Wörter des Textes werden im Englischen bezüglich ihrer grammatischen Funktion und Anzahl markiert (*annotiert*). Die Identifikation muss derart erfolgen, dass sie den Kontext des Satzes erfasst, also Sinn ergibt. Diese Kennzeichnung geht über die Analyse einzelner Wörter hinaus, sie erfasst auch die Syntax umgebender Wörter eines Satzes.

Im Deutschen ist es etwa recht wahrscheinlich, dass auf ein Nomen am Satzanfang als nächstes Wort ein Verb folgt, da dies einen typischen Satzbau darstellt. Ein part-of-speech tagger sollte daher beispielsweise im Satz "*Ich fahre Fahrrad.*" korrekt das "Ich" als Subjekt, das "fahre" als Verb und das "Fahrrad" als Akkusativobjekt darstellen; Die isolierte "Betrachtung von "Fahrrad" ließe aber auch die Kategorisierung als Subjekt zu, da sich die zwei Möglichkeiten bezüglich der Morphologie nicht unterscheiden. Diese Mehrdeutigkeit einzelner Worte im Satz wird durch Analyse des syntaktischen Kontextes eliminiert, sodass POS-tagging eine Eindeutigkeit der syntaktischen Deutung erzielt.

Solche Regelmäßigkeiten werden in Form von sogenannten *corpora* geliefert; sie stellen Trainingsdaten in Form von tatsächlicher Sprache dar. Im Englischen wird etwa häufig die Zeitung The Wall Street Journal genutzt, um die Repräsentation bestimmter Satzstrukturen und Wortfolgen statistisch zu erfassen. Alternativ können auch standardisierte Corpora wie der Lancaster-Oslo-Bergen Corpus verwendet werden, der etwa eine Million Wörter in Sprache enthält. Corpora sind etwa für NLP auf Basis von Machine-Learning-Algorithmen erforderlich, können darüber hinaus aber auch etwa zur Rechtschreibprüfung oder Textunterteilung benutzt werden. Die meisten Textverarbeitungsprogramme können heutzutage etwa erkennen, wenn sich syntaktische Fehler in einem Satz ereignen (Kommasetzung, falscher Kasus, etc.). Es ist anzumerken, dass part-of-speech tagging nicht unbedingt vorher annotierten Texten bedarf, aber ohne diese fällt eine syntaktische Einordnung ungleich schwerer und die Fehlerquote steigt massiv. Mehrdeutigkeiten aus der Syntax heraus können somit nicht aufgeschlüsselt werden.

Die Anwendung eines Corpus auf ein natürlichsprachliches Dokument er-

möglichst sogleich auch eine Vorhersage der nächsten Wörter bzw. Sprachbausteine, nachdem schon ein Teilsatz gelesen wurde. Diese *prediction* kann auf Basis bekannter Regelmäßigkeiten also schon vor der Wortanalyse recht genau die nächste grammatikalische Komponente bzw. den Worttyp vorhersagen.

### 2.5.1 N-gram Modelle zur Sprachvorhersage und -modellierung

Auf Basis der vorliegenden Corpora existieren eine Vielzahl an Regeln bezüglich des Satzbaus und bekannter Folgen von Worttypen, wenn ein neues Dokument annotiert werden soll. Zur Veranschaulichung einer möglichen Annotation soll in diesem Abschnitt die Notation von CLAWS 7 (<http://ucrel.lancs.ac.uk/claws7tags.html>) für Annotationen genutzt werden.

Bei erneuter Betrachtung des obigen Beispiels "*Ich fahre Fahrrad.*" könnte, wenn dieser Satz als minimaler Corpus vorliegt, neben der Regel Subjekt-Prädikat-Objekt eines Satzbaus auch beinhalten, dass der Punkt ein Satzende markiert und das Substantive und Personalpronomen mit einem Großbuchstaben beginnen ("Ich" und "Fahrrad" können Nomen sein und sind groß geschrieben). Sprachvorhersage nutzt diese bekannten Regeln und würde etwa aus der obigen Regel herleiten, dass, wenn ein Verb gelesen wurde, darauf immer ein Substantiv bzw. das Satzobjekt folgt. POS-Tagging macht sich diese Beobachtungen durch Berechnung von Wahrscheinlichkeiten bei der Annotation von Wortfolgen zunutze.

Kontextfreie Betrachtung einzelner Wörter, also ohne andere Taggings mit einzubeziehen, wird als Unigram bezeichnet. Dabei werden jedoch Regeln über den Zusammenhang von Wortketten vollständig ignoriert und das Ergebnis ist ähnlich dem der morphologischen Analyse. Vorhersage-Systeme, die sich der Annotierung des vorigen Wortes bedienen, werden als Bigramme (englisch *bigrams*) bezeichnet, Trigramme dementsprechend bei Berücksichtigung der zwei vorigen Wörter und N-gramme, wenn alle  $n-1$  zuvor gelesenen Wörter, also die gesamte Terminologie des Textes bis zum  $n$ -ten Wort, berücksichtigt werden. Die Wahrscheinlichkeit des Typen des Folgewortes auf Basis eines Bigrams lässt sich berechnen, indem aus einem Corpus die Worttypen an der jeweiligen Satzstelle gezählt und statistisch erfasst werden. Als Beispiel dazu soll hier zunächst ein Bigram zur tatsächlichen Wortvorhersage dienen, anschließend wird diskutiert, inwiefern solche Techniken auch den nächsten part-of-speech in einem Text vorhersagen können. Aus folgendem Corpus vierer Aussagen lässt sich dies wie folgt ableiten:



<s> ich mag Züge <s> ich mag guten Tee  
 <s> guten Morgen <s> guten Abend <s>

Zwischen einzelnen Äußerungen wird der Platzhalter <s> als Indikator einer neuen Aussage verwendet, sodass dem Bigram nun eine Zeichenkette mit den obigen Aussagen, getrennt durch <s> zur Verfügung steht. Nun wird die mögliche Wortkombinatorik aus dem Corpus extrahiert, indem Kombinationen gezählt werden. Berechnet werden nun die Wahrscheinlichkeiten des Folgewortes auf Basis der vorherigen, formal ausgedrückt:  $\frac{C(w_{n-1}w_n)}{\sum w C(w_{n-1}w)}$ . Es lassen sich nun die Wahrscheinlichkeiten des jeweils nächsten Folgewortes

Sequenz	Anzahl	$P(w_n P(w_{n-1}))$
<s>	4	
<s> ich	2	0,50
<s> guten	2	0,50
ich mag	2	1,0
mag Züge	1	0,5
mag guten	1	0,5
Züge	1	
Züge <s>	1	1
guten	3	
guten Tee	1	$\frac{1}{3}$
guten Morgen	1	$\frac{1}{3}$
guten Abend	1	$\frac{1}{3}$
Tee	1	
Tee <s>	1	1
Morgen	1	
Morgen <s>	1	1
Abend	1	
Abend <s>	1	1

Tab. 1: Wahrscheinlichkeitstabelle Bigram zur Wortvorhersage

nach einem Wort ablesen. Auf "ich mag" folgt insg. einmal "Züge" und einmal "guten". Daher kann nun genau die Wahrscheinlichkeit des nächsten Wortes "Züge" mit  $P("Züge"|"ich mag") = 50\%$  berechnen.

### 2.5.2 Stochastisches POS-tagging

Neben der Möglichkeit, einzelne Wörter vorherzusagen, eignen sich n-Gramme wie oben erwähnt auch zur Darstellung der Wahrscheinlichkeiten von Satzbausteinen, also der von Platzhaltern für alle möglichen Wörter einer Kategorie bzw. Komponente. Stochastisches Part-of-speech tagging kann anhand dieser die Wahrscheinlichkeiten vorhersagen, welcher Worttyp wahrscheinlich noch folgen wird. Die Trainingsdaten sind, nach Best-Practices aus dem maschinellen Lernen, in der Regel manuell und durch Benutzer markiert worden, damit dem Programm ein korrekter Datensatz zur Verfügung steht. Auf Basis eines Corpus können somit Mehrdeutigkeiten durch den syntaktischen Zusammenhang, den der Corpus beschreibt, eindeutig aufgelöst werden. In einem Beispiel mit syntaktischer Ambiguität werden jedoch Wörter zunächst verschieden annotiert. Das tagset von CLAWS 7 annotiert Wortkategorien wie folgt:

PPIS2	1st person plural subjective personal pronoun
PPHO1	3rd person sing. objective personal pronoun
NN1	singular common noun
VV0	base form of lexical verb
VVD	past tense of lexical verb
PUN	punctuation

Tab. 2: Annotationen gemäß [CLW7] (Auszug)

Als Beispiel dient hier der englische Satz *"We saw her duck."*. Dieser enthält Mehrdeutigkeiten, da "duck" entweder als *"Ente"*, also als Substantiv *"ihre Ente"* oder als "ducken", also als Verb (dt. *"sie (sich) ducken"*) interpretiert werden kann. Zusätzlich kann "we saw" als *"sahen"*, oder als *"sägen"*, also zeitformabhängig verstanden werden. Äquivalent im Deutschen sind also die drei Sätze *"Wir sahen ihre Ente."*, *"Wir sahen sie (sich) ducken."* und *"Wir zerschneiden ihre Ente"* mit völlig unterschiedlicher Aussage. (Semantisch betrachtet stellt sich ferner die Frage, ob die Deutung von "her duck" als Nomen mit "ihr Haustier" etwas anderes bedeutet als "für sie zum Essen")

Diese sinnvollen Annotationen könnten durch einen CLAWS 7-POS-tagger wie folgt erfasst werden:

1. We(PPIS2) saw(VVD) her(PPH01) duck(VV0) .(PUN)
2. We(PPIS2) saw(VVD) her(PPH01) duck(NN1) .(PUN)
3. We(PPIS2) saw(VV0) her(PPH01) duck(NN1) .(PUN)

Problematisch ist, dass auch syntaktisch falsche Deutungen entstehen können, wenn, wie oben erwähnt, unzureichende Trainingsdaten und Regeln für die Wortfolgen existieren. Dann werden alle möglichen Teildeutungen der einzelnen Wörter als Tagging ausgegeben, etwa mit der Bedeutung "*Wir zerschneiden sie sich ducken*" aus der Syntax.

4. We(PPIS2) saw(VV0) her(PPH01) duck(VV0) .(PUN)

Durch das Verwenden von manuell erstellten Corpora kann zumindest diese eindeutig falsche Deutung der Syntax eliminiert werden, da solch ein Sprachkonstrukt in der englischen Sprache nicht auftreten kann. Das Tagging des folgenden minimalen Korpus könnte dann wieder als Eingabe für ein n-Gram dienen, welches auf Basis der höchsten Wahrscheinlichkeit ein Wort annotiert.

People use to see a duck. Most people duck in order to avoid danger.

Der Corpus könnte annotiert sein:

We(NN2) use(JK) to(T0) see(VV0) a(AT1) duck(NN1).(PUN) Most(DAT) people(NN2) duck(VV0) in(BCL11) order(BCL12) to(BCL13) avoid(VV0) danger(NN1).

Daraus ergibt sich unter Verwendung eines n-Grams umgangssprachlich formuliert, dass etwa in einem Satz, in welchem bereits ein Wort als Verb im Präsens getaggt wurde, ein Objekt, also ein als Nomen getaggt Wort, folgen muss. Im ersten Satz ergibt sich dies etwa aus der Stellung des "see" als Verb vor dem "duck". Dementsprechend kann so schon die vorhin beschriebene Deutung 4. ausgeschlossen werden, da diese Regel des Satzbaus nicht vorkommt. Weitere mögliche Regelmäßigkeiten, die ein n-Gram statistisch erfasst, wäre die Möglichkeit einer Before-Clause (BCL) nur nach einem Nomen oder die hohe Wahrscheinlichkeit eines Punktes nach einem Nomen im Singular. Auffällig ist, dass die Deutung von saw als Verb in Past Tense entfällt, da keine Regel auftritt, die ein Verb in Past-Tense vor einem Nomen enthält. Daher erfasst ein N-Gram auf Basis dieses Korpus nur Verben im Präsens korrekt (im Corpus taucht keine Deutung eines Verbs als Vergangenheitsform auf).

Abschließend lässt sich daher sagen, dass die Trainingsdaten möglichst repräsentativ für die möglichen Konstruktionen in einer Sprache bzw. in der Anwendungsdomäne des POS-taggers sein müssen, um akkurate Ergebnisse

zu erzielen. Dies betrifft sowohl die Qualität der im Corpus enthaltenen Wörter und Sätze, als auch deren Korrektheit im Hinblick auf die quantitative Repräsentation einzelner Konstrukte. Häufig anzutreffende Hauptsatz-Nebensatz-Konstruktionen müssen etwa für die Analyse eines stilistisch einfachen Textes, der zum Großteil aus solchen besteht, häufiger vorhanden sein, als kompliziert verschachtelte Kausalsätze und eingeschobene Relativsätze mit beispielsweise vielen beschreibenden Aufzählungen.

## 2.6 Parsen und Generieren

Wie bereits in 2.4 angekündigt befasst sich dieser Abschnitt mit dem Parsen und Generieren von Sätzen auf Grundlage der vorhergegangenen Verarbeitungsschritte. Fokus liegt dabei auf der Auflösung von Uneindeutigkeiten, welche in nicht durch die morphologische Analyse und Part-of-Speech-Tagging bewältigt werden konnten. Dies wird anhand verschiedener kontextfreier Grammatiken genauer erläutert.

### 2.6.1 Generative Grammatik

Der Begriff der generativen Grammatik beruht auf der Arbeit von Chomsky in den 1950er Jahren. Man versteht darunter eine formale Grammatik, welche in der Lage ist alle möglichen Sätze einer natürlichen Sprache und nur genau diese zu generieren. Hier ist anzumerken, dass dem tatsächliche Entwurf einer solchen Grammatik bisher nicht einmal nahe kommen konnte. Jedoch ist für Linguisten der prinzipielle Aufbau von solchen Grammatiken höchst interessant (vor allem von solchen Grammatiken, welche sich auf alle natürlichen Sprachen anwenden lassen), während sich NLP-Forscher eher mit dem Bau und der Nutzung von möglichst umfangreichen Grammatiken befassen.

Ähnlich zu der ableitenden Morphologie versucht man hier hier Zeichenketten eine interne Struktur zu verleihen, welche es ermöglicht Informationen zu den einzelnen Bestandteilen zu extrahieren, jedoch tut man dies mit Sätzen (oder Satzteilen) anstelle von einzelnen Wörtern. Darstellen lässt sich diese Struktur beispielsweise durch Klammerung wie folgt:

((the (fisherman)) fishes) ((the (old fisherman)) fishes)

Durch das obige Beispiel wird die Bindung der einzelnen Wörter untereinander deutlich: "old" gehört eindeutig zu "fisherman" und das "the" bezieht sich im zweiten Satz auf "old fisherman" zusammen. Die Klam-

merung (the old) ist also auf keinen Fall möglich, da "old" mit dem "fisherman" eingeklammert ist.

Bei zwei Grammatiken spricht man von schwacher Äquivalenz, falls sie die gleichen Zeichenketten generieren können und von starker Äquivalenz, wenn sie diese auch gleich klammern.

Die meisten Ansätze verleiht man der internen Struktur nun verschiedene Bezeichnungen abhängig von der oben gezeigten Klammerung. Somit würde man "the fisherman" (ebenso wie "the old fisherman") im Englischen als "noun-phrase" (NP) bezeichnen. "fishes" würde (zusammen mit jeglichen Zusätzen wie "fishes very good" oder "fishes fish" als "verb-phrase" (VP) gekennzeichnet. Diese Bezeichnungen dienen in kontextfreien Grammatiken als linke Seite von Produktionsregeln, wie im folgenden Abschnitt genauer dargestellt wird.

### 2.6.2 Kontextfreie Grammatiken

Kontextfreie Grammatiken stammen aus dem Bereich der Theoretischen Informatik und werden dargestellt durch ein Tupel mit 4 verschiedenen Komponenten:

1. Eine Menge nicht-terminaler Symbole (wie z.B. NP oder VP aus dem vorherigen Abschnitt)
2. Eine Menge terminaler Symbole (die Wörter aus denen später die Zeichenkette bestehen wird, also "the", "old", "fisherman" und "fishes")
3. Eine Menge von sogenannten Produktionsregeln mit folgenden Eigenschaften:
  - (a) Eine Regel besteht aus einer linken und einer rechten Seite, welche durch einen Pfeil ( $\rightarrow$ ) getrennt sind.
  - (b) Auf der linken Seite befindet sich genau ein nicht-terminales Symbol.
  - (c) Auf der rechten Seiten befindet sich mindestens mindestens ein Symbol (terminal oder nicht-terminal).
4. Das Startsymbol (konventionell "S"), welches zu den nicht-terminalen Symbolen gehört

Es ist zu beachten, dass in kontextfreien Grammatiken generell das leere Wort  $\varepsilon$  als terminales Symbol mit eingeschlossen ist. Aus linguistischer Sicht verkompliziert dies jedoch nur die Grammatiken und kann daher bezüglich NLP ausgeschlossen werden (es ließe sich außerdem auch zu jeder kontextfreien Grammatik eine schwach-äquivalente Grammatik erzeugen, welche das leere Wort nicht beinhaltet).

Man spricht von einer linksbündigen bzw. rechtsbündigen Grammatik, wenn alle nicht-terminalen Symbole ganz links bzw. rechts auf der rechten Seite der Regel stehen (also in der Form " $NT \rightarrow NT \ t$ " bzw. " $NT \rightarrow t \ NT$ "). Diese Grammatiken sind jedoch schwach-äquivalent zu regulären Grammatiken, welche sich mit endlichen Automaten implementieren lassen, was jedoch für den Aufbau von natürlichen Sprachen nicht ausreichend ist. Der Ansatz sich auf eine rechts- bzw. linksbündige Grammatik zu beschränken, um damit eine natürliche Sprache zu generieren entfällt also sofort.

## 2.7 Parsen mit constraint-basierten Grammatiken

Wie zuvor an verschiedenen Beispielen demonstriert wurde, weisen die kontextfreien Grammatiken beim Parsen und Generieren einige Schwächen auf. Daher wird im folgenden Abschnitt ein neuer, komplexerer Ansatz vorgestellt um sich mit den gleichen Problemen zu befassen. Kontextfreie Grammatiken können als constraint-basiert interpretiert werden, diese Bezeichnung wird jedoch üblicherweise für mächtigere Grammatiken verwendet. Constraint-basierte Grammatiken sind in NLP weit verbreitet und werden mit sogenannten feature-structures (Eigenschaftsstrukturen) dargestellt, daher auch oft feature-structure (FS) Grammatik genannt. Im folgenden Abschnitt soll die Umsetzung solch einer Grammatik durch feature-structures demonstriert werden.

### 2.7.1 Beispiel zu einfacher FS Grammatik

FSs weisen große Ähnlichkeit zu Baumstrukturen auf. Sie haben genau eine Wurzel und sind von dieser aus gerichtet. Ein Knoten kann keinen, einen oder mehrere Kindesnoten besitzen. Blätterknoten verfügen über Werte und Kanten sind mit Eigenschaften versehen. Ein Weg entlang von Eigenschaften wird als Pfad bezeichnet. Eigenschaften sind entweder atomar oder zusammengesetzt, abhängig davon ob sie auf ein Blatt des Baumes oder auf einen weiteren nicht-Blattknoten zeigen. In dem folgenden dem Paper

entnommenen Beispiel ist eine FS abgebildet, die sich in zwei Eigenschaften, "*CAT*" für die Wortkategorie und "*AGR*" für agreement (Stimmigkeit), teilt, welche jeweils auf die Blattknoten mit den Werten "*noun*" (für Nomen) und "*sg*" (für Singular) zeigen. Die Eigenschaften "*CAT*" und "*AGR*" sind also atomar.

Fügt man wie in der nächsten Abbildung vor der Wurzel eine weitere Eigenschaft "*Head*" hinzu, so ist diese zusammengesetzt, da sie auf einen nicht-Blattknoten zeigt. Dieser wird wiederum in die beiden Eigenschaften "*CAT*" und "*AGR*" aufgeteilt.

**References**

- [CLW7] University Centre for Computer Corpus Research on Language, Lancaster University, "CLAWS part-of-speech tagger for English", <http://ucrel.lancs.ac.uk/claws/>, abgerufen am 19.08.2018 16.43 Uhr