Morphologische Analyse des Spanischen mithilfe von Bäumen

Hauptseminar: "Stringverarbeitung in den Geisteswissenschaften"
Dozent: Prof. Dr. Jürgen Rolshoven

vorgelegt von

Eva Hasler

15.04.2006

Inhaltsverzeichnis

1 Motivation **1**

2 Allgemeines zu Tries und Suffixbäumen **2**

2.1 Tries 2

2.2 Suffixbäume 2

2.3 Der Ukkonen-Algorithmus 3

3 Linguistischer Hintergrund **4**

4 Programm zur morphologischen Analyse von spanischen Korpora **5**

4.1 Hintergrund 5

4.2 Wahl der Datenstruktur 6

4.3 Programmablauf 6

4.3.1 Starten des Programms 6

4.3.2 Unterschied in der Analyse von Präfixen und Suffixen 7

4.3.3 Korpusauswahl 7

4.3.4 Erstellung einer Frequenzliste aller Blätter im Baum 8

4.3.5 Untersuchung der Affix-Kontexte 9

4.3.6 Umkehrung der Kontextsuche 10

4.3.7 Erstellung von Klassen 11

4.3.8 Suffixanalyse mit dem Ukkonen-Suffixbaum 15

4.4 Fazit und Ausblick 16

Literatur **18**

1 Motivation

Bei der maschinellen Verarbeitung von geschriebener Sprache tritt oft das Problem auf, dass die Verarbeitung Wissen über die betreffende Sprache erfordert, das entweder noch nicht gewonnen wurde oder zumindest maschinell noch nicht verfügbar ist. Zum Beispiel ist es für ein Übersetzungsprogramm unmöglich, eine korrekte Übersetzung zu generieren, wenn schon der Ausgangssatz mangels syntaktischer Informationen nicht richtig analysiert wurde. Bei Sätzen mit eher ungewöhnlicher Syntax lässt sich beim derzeitigen Stand von Übersetzungsprogrammen beobachten, dass falsche Übersetzungen generiert werden, weil spezielle und umfassende Regeln fehlen.

Es gibt auch Ansätze zur Übersetzung, die weitgehend auf linguistische Regeln verzichten und stattdessen komplizierte statistische Methoden anwenden, um die wahrscheinlichste Analyse zu bekommen. Auch wenn diese Verfahren bisher erfolgreicher sind als regelbasierte, besteht zumindest ein ideologisches Interesse daran, den Mangel an struktureller Information auszugleichen. Zudem stoßen auch statistische Verfahren früher oder später an ihre Grenzen.

Analog zu den syntaktischen Problemen beim Erstellen eines Strukturbaums in der maschinellen Übersetzung verhält es sich bei der morphologischen Analyse. Um beispielweise beim Taggen Grundform und Flektionsendung eines Verbs zu erkennen, muss Wissen über die paradigmatischen Ausprägungen von Verben vorhanden sein, auf das zugegriffen werden kann. Gesetzt den Fall, dieses morphologische Wissen sollte für alle bisher bekannten Sprachen erzeugt werden - was bisher von Hand erledigt werden muss - ist der Umfang dieser Arbeit kaum abzuschätzen und es kommt schnell der Gedanke auf, diese mühevolle Aufgabe an einen elektronischen Helfer abzugeben. Auch um sich der Analyse einer bisher unbekannten Sprache zu nähern, könnte eine automatische Lösung von Nutzen sein, um schnelle Ergebnisse zu bekommen.

Mit geeigneter Software linguistisches Wissen automatisch aus einem Korpus extrahieren zu können ist ein erstrebenswertes, aber auch noch weitgehend unerreichtes Ziel. Jedoch haben sich in den letzten Jahren die Voraussetzungen dafür extrem verbessert, da es immer mehr elektronische Texte in vielen verschiedenen Sprachen gibt, auf die relativ uneingeschränkt zugegriffen werden kann. Die gewünschte Information liegt schon vor, sie muss nur noch mit einer geeigneten Methode in eine geordnete Form gebracht werden, damit sie nutzbar ist.

Es könnte nun als Einwand die Behauptung aufgeworfen werden, dass das Wissen über Sprache immer an die menschliche Fähigkeit, Sprache zu verstehen gebunden ist und dass daher selbiges auch für die Analyse gilt. Dann wäre es aber immer nur möglich, eine Aussage über ein Wort zu treffen, wenn sein gesamtes Paradigma im Korpus vorgekommen wäre und dies würde ein recht beträchtliches Korpus voraussetzen. Beobachtet man jedoch, wie Kinder in der Phase des Spracherwerbs aus einer - verglichen mit riesigen Sprachkorpora - geringen Menge von Sprachmaterial die nötige strukturelle Information gewinnen, um die Sprache zu lernen, gibt

es Grund zur Hoffnung, dass man diesen Prozess auch mit Computerprogrammen nachahmen könnte. Schließlich ist dieser Prozess bei Kindern möglich, ohne dass sie jedes Lexem schon in jeder möglichen Ausprägung einmal gehört haben. Verglichen mit dem Spracherwerb ist das Auffinden von morphologischen Mustern natürlich ein sehr kleines Teilproblem. Dennoch ist es wichtig, sich auch mit trivial erscheinenden Fragestellungen zu beschäftigen, da sie die Basis sind, auf der man sich später auch größeren Problemen widmen kann.

2 Allgemeines zu Tries und Suffixbäumen

2.1 Tries

Ein Trie ist eine geordnete Baumstruktur, in der Zeichenketten gespeichert werden können. Die Kanten tragen Labels von je einem Buchstaben beim nicht kompakten Trie und beliebig lange Labels beim kompakten Trie. Strings mit gleichen Präfixen teilen sich die Kanten und Knoten, bis zu der Stelle, an der die Strings sich unterscheiden. Die Knoten sind nicht mit Zeichen gelabelt sondern mit Ganzzahlen, jedoch nur dann, wenn an dem Knoten ein Wort endet. Ansonsten tragen die Knoten kein Label, sondern sind je nach ihrer Stellung im Baum mit einem String assoziiert.
Eine gängige Anwendung von Tries ist die kompakte Speicherung von Lexika, da in ihnen schnelles Suchen, Einfügen und Löschen möglich ist. Sie werden zum Beispiel im Bereich der automatischen Rechtschreibüberprüfung benutzt, wo approximatives Matchen erforderlich ist. Außerdem eignen sie sich zur lexikographisch geordneten Ausgabe gespeicherter Strings (vgl. *Trie.* 2006, [1]).

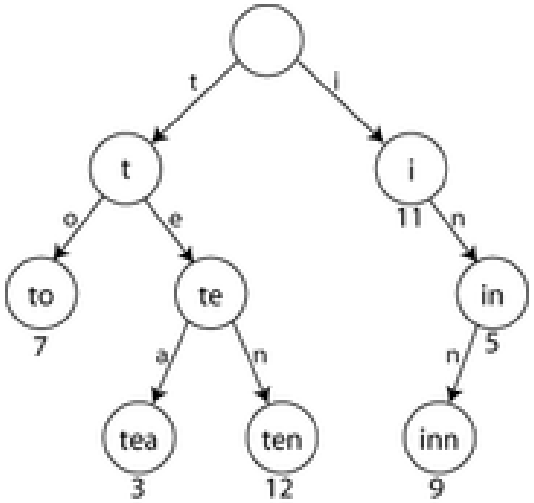

Bild 1: Trie für die Wörter *to*, *tea*, *ten*, *i*, *in*, and *inn*.

2.2 Suffixbäume

In einem Suffixbaum werden alle Suffixe eines Textes gespeichert, angefangen mit dem kürzesten Suffix, dem leeren String, bis hin zum kompletten Text als einem einzigen String. Dies ist zwar

eine sehr redundante Art der Informationsspeicherung, hat aber den Vorteil, dass jeder Teilstring des Textes an der Wurzel des Baums beginnt. Der Teilstring kann also im Baum gefunden werden, ohne dass zuerst der dem Teilstring im Text vorangehende String gesucht werden muss. Suffixbäume bieten eine Möglichkeit der Präprozessierung von Texten, mit der unterschiedliche String-Probleme lösbar sind. Zum Beispiel können in linearer Zeit Teilstrings eines Textes gefunden werden, was mit anderen Algorithmen wesentlich aufwendiger ist. Suffixbäume arbeiten hierbei so effizient, weil die Dauer der Präprozessierung linear zur Textlänge zunimmt und die Dauer der Teilstringsuche sogar nur noch linear zur Länge des Suchstrings steigt. Interessanter noch als die Suche nach exakten Teilstrings ist die Suche nach Übereinstimmungen mit Fehlern. Viele Problemstellungen erfordern es, dass ähnliche Strings mit einer bestimmten Zahl von Abweichungen gefunden werden. Mit Suffixbäumen ist auch diese kompliziertere Suche in linearer Zeit durchführbar (vgl. Gusfield, [2]).

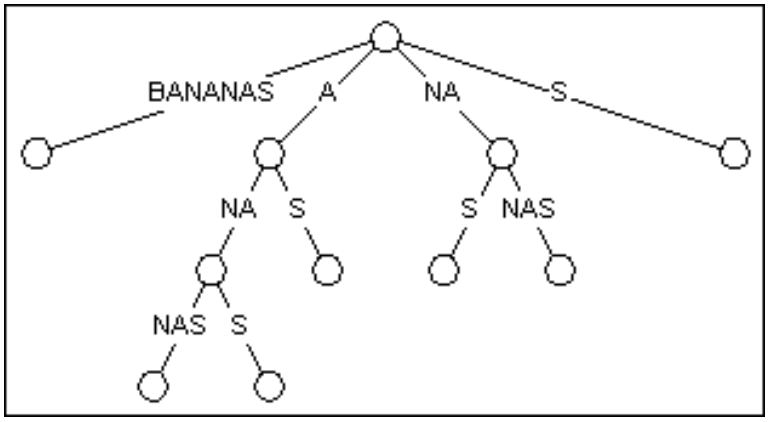

Bild 2: Suffixbaum für *BANANAS*

2.3 Der Ukkonen-Algorithmus

Die in Kapitel vier beschriebene Programmoberfläche *Suffixe(Ukkonen)* verwendet einen Suffixbaum, dessen Erzeugungsalgorithmus auf den Finnen Esko Ukkonen zurückgeht. Dieser Algorithmus ist angeblich der einfachste lineare Algorithmus zur Suffixbaumerzeugung ("Esko Ukkonen [..] devised a linear-time algorithm for constructing a suffix tree that may be the conceptually easiest linear-time construction algorithm", Gusfield, S. 94, [2]).
Der Baum wird hierbei in linearer Zeit zur Textlänge aufgebaut, indem nacheinander alle Präfixe des Textes, angefangen beim kürzesten, in den Baum eingefügt werden. Der in jedem Schritt hinzukommende neue Endbuchstabe des Präfixes wird an jedes schon im Baum vorhandene Suffix angehängt, angefangen beim längsten Suffix. Endet ein Suffix an einem impliziten oder expliziten Knoten, so wird ein neuer Knoten mit dem neuen Endbuchstaben des Präfixes hinter dem Suffix eingefügt. Dazu wird entweder der implizite Knoten zu einem expliziten gemacht, an den der neue Knoten gehängt wird, oder dieser wird an den bestehenden expliziten Knoten angehängt (vgl. Nelson, [3]).
Der Ukkonen-Suffixbaum hat gegenüber dem kompakten Trie einen Vorteil in der Dauer der Erzeugung. Der Trie ist aber in seinem Informationsgehalt eher auf das Ziel der in der vorliegen-

den Arbeit angestrebten Untersuchung zugeschnitten. Soll für denselben Zweck der Ukkonen-Suffixbaum benutzt werden, müssen zusätzliche Bearbeitungsschritte unternommen werden, auf die an entsprechender Stelle eingegangen wird.

3 Linguistischer Hintergrund

Die in dieser Arbeit angestrebte Analyse beruht auf der Methodik des Strukturalismus, einer in der Sprachwissenschaft anerkannten Forschungsmethode, begründet durch den Genfer Sprachwissenschaftler Ferdinand der Saussure. Durch Betrachtung von Phänomenen in ihrem Kontext sollen die Kombinationsregeln der Phänomene beziehungsweise ihre zugrundeliegende Struktur erschlossen werden. Außerdem ist es "Ziel einer Strukturanalyse [..], sämtliche Einheiten eines Systems (einer Struktur) herauszuarbeiten und zu klassifizieren sowie die Regeln ihrer Kombination zu beschreiben" (Eagleton, 1994, [4]). Der Hintergrund dieser Absicht ist die "Grundannahme, dass Phänomene nicht isoliert auftreten, sondern in Verbindung mit anderen Phänomenen stehen" (*Strukturalismus*, [5]).

Die Einheiten, die durch das nachfolgend beschriebene Programm gefunden werden sollen, sind Präfixe und Suffixe des Spanischen. Der zur Analyse vorliegende spanische Text wird in einer Baumstruktur abgespeichert, weil durch diese Art der Speicherung die Zusammenhänge zwischen den Wörtern, beziehungweise das, was zwei Wörtern gemeinsam ist, sichtbar gemacht wird. Im Suffixbaum wie im Trie teilen sich Wörter mit gleichen Präfixen die Knoten und Kanten, die mit diesem Präfix gelabelt sind. Diese signifikante Stelle - der Knoten, bis zu dem die Wörter gleich sind und ab dem sie sich unterscheiden - ist im Idealfall eine Morphemgrenze. Wenn dies der Fall ist, sind die Labels der abzweigenden Kanten gute Kandidaten für Präfixe oder Suffixe (je nachdem, ob die Wörter vorwärts oder rückwärts im Baum gespeichert wurden). Deswegen wird bei der Analyse des Baums der Fokus zunächst nur auf die Blätter gelegt. Die Analyse könnte auch ausgeweitet werden, indem nicht nur die Blätter betrachtet werden, sondern auch innerhalb des Baums nach signifikanten Stellen gesucht wird.

Es gibt eine Eigenschaft von Tries und Suffixbäumen, die dazu führt, dass die potentiellen Morpheme, die im Baum gefunden werden, unweigerlich von denen abweichen, die die Sprachwissenschaft postuliert. Die Bäume sind deterministisch, das heißt zwei Kanten, die von demselben Knoten abzweigen, können nicht mit demselben String gelabelt sein, beziehungsweise bei Labels länger als ein Buchstabe dürfen die Anfangsbuchstaben der Labels nicht gleich sein. Dies führt dazu, dass bei vollständig gespeichertem Paradigma (am Beispiel eines Verbs) die Flexionsendungen eines spanischen Verbs wie *alcanzar* ('erreichen') immer anders abgespeichert werden als ein Spanisch-Lehrbuch sie beschreiben würde.

Beispiel Suffix-Suche Die Flexionsendungen für 2. Person Singular (*-ar*) und 3. Person Plural (*-an*) beginnen beide mit *a*. Es kann aber im Baum unterhalb des Knotens für *alcanz* nicht zwei verschiedene Blatt-Labels geben, die mit *a* beginnen. Daher gibt es einen Knoten, zu

dem die Kante mit dem Label a führt und von diesem Knoten abzweigend zwei Blätter mit den Labels n und r.

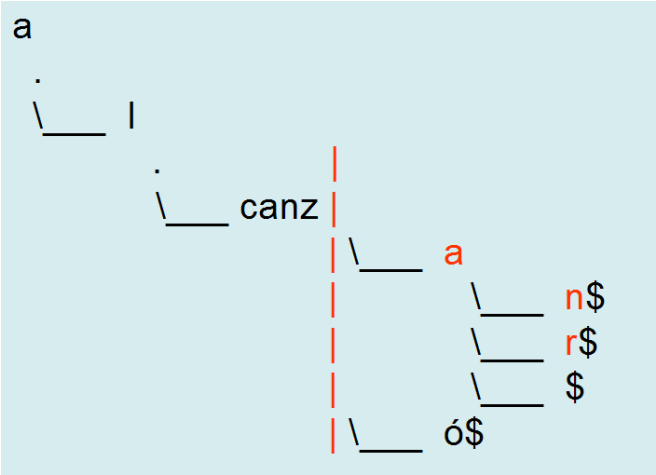

Bild 3: Baumausschnitt des Verbs *alcanzar*

An diesem Punkt stellt sich die Frage, welche der vorgeschlagenen Flexionsendungen richtig sind. Ausgehend von der Annahme, dass "die Struktur [..] nicht auf der Ebene der Wirklichkeit [existiert], sondern nur auf der Ebene des Modells" (*Strukturalismus*, [5]), ist diese Frage eigentlich irrelevant. Es geht dann vielmehr darum, welches Modell mit seinen Elementen die Phänomene am besten beschreibt. Würde eine neue Menge von Elementen mit Kombinationsregeln eingeführt, die die Produktion von Verben oder anderen Wörtern besser beschreibt, wäre das legitim.

4 Programm zur morphologischen Analyse von spanischen Korpora

4.1 Hintergrund

Das Programm soll dazu dienen, anhand einer relativ kleinen Zusammenstellung von spanischen Texten Informationen über die spanische Morphologie zu gewinnen. Der Fokus kann hierbei auf Präfix- oder Suffix-Morpheme gelegt werden, Morpheme im Wortinnern werden bisher nicht untersucht. Es ist wichtig zu erwähnen, dass keinerlei linguistisches Wissen zugrunde gelegt wird, sondern lediglich die Strukturen, die im Korpus vorhanden sind, herausgefiltert und sichtbar gemacht werden. Die Ergebnisse weichen daher mitunter von den linguistischen Vorstellungen eines Präfixes oder Suffixes ab.

Da die Untersuchung morphologischer Natur ist, soll die Grundeinheit das graphische Wort sein. Daher wird der Text in Teiltexte aufgesplittet, die jeweils einem Wort entsprechen, und nacheinander in den Baum eingefügt. Die ursprüngliche Motivation zur Erstellung des Programms lag in der Frage, ob ein Suffixbaum dazu geeignet ist, strukturelle Informationen aus einem Korpus zu gewinnen. Im Folgenden wird erläutert, warum sich ein kompakter Trie ebenso eignet, um zur selben Lösung zu kommen.

4.2 Wahl der Datenstruktur

Die verwendete Datenstruktur der ersten drei Programmoberflächen ist ein kompakter Trie, bei der vierten Oberfläche wird ein Suffixbaum verwendet. Beim Trie werden nicht alle Suffixe eines Textes (hier: Wortes) eingefügt, sondern nur der komplette Text als String. Beispielsweise würden in einen Suffixbaum für das Wort *Haus* die Strings *Haus*, *aus*, *us*, *s* und der leere String eingefügt. Der kompakte Trie enthält jedoch nur den String *Haus*.

Inhaltlich betrachtet hat der kompakte Trie zwei Vorteile gegenüber dem Suffixbaum. Zum einen verfälscht der Suffixbaum die Häufigkeitsverteilung der Blattknoten im Baum und zwar abhängig von der Länge der eingefügten Wörter. Wird zum Beispiel das Wort *aproximadamente* (in einen leeren Baum) eingefügt, enthält der Suffixbaum danach elfmal den String *mente*. Wird das Wort *claramente* eingefügt, ist der String *mente* nur sechsmal enthalten. Dies ist problematisch, da das Programm die Häufigkeiten der Blätter auswerten soll, um die Suche nach potentiell bedeutsamen Strings einzugrenzen. Es ist unwahrscheinlich, dass Blattlabels, die im Baum nur einmal vorkommen, eine bedeutungs- oder funktionstragende Einheit darstellen. Wenn die Struktur der Baums aber derart ist, dass das Auftreten eines Suffixes im Baum durch die Länge der eingefügten Wörter gewichtet wird, ist diese Art der Filterung nutzlos.

Zum anderen enthält der Suffixbaum viele Informationen, die im vorliegenden Programm nicht verwendet werden und deswegen für den angestrebten Zweck überflüssig sind. Wird ein Suffix im Baum gefunden, das mit einer bestimmten Häufigkeit auftritt und daher näher betrachtet werden soll, ist von Interesse, zu welchem Wort dieses Suffix gehört. Wird zum Beispiel das Suffix *-mente* gefunden, soll auch dessen Präfix-String *clara-* gefunden werden. Es bringt jedoch keinen Informationsgewinn zu erfahren, dass im Suffixbaum auch *a-*, *ra-*, *ara-* und *lara-* Präfix-Strings von *-mente* sind. Man kann diese überflüssigen Informationen zwar herausfiltern, indem man die Strings mit einer Liste aller im Baum enthaltenen graphischen Wörter vergleicht, es ist aber übersichtlicher, wenn der Baum erst gar keine überflüssigen Informationen enthält.
(Da die Wörter *Präfix*, *Suffix* und *Affix* ambig sind, werden im Folgenden die nicht unter morphologischen Gesichtspunkten betrachteten Affixe als *Präfix-String*, *Suffix-String* und*Affix-String* bezeichnet).

4.3 Programmablauf

4.3.1 Starten des Programms

Alle Klassen des Programms wurden in eine ausführbare jar-Datei mit dem Namen *MorphologischeAnalyse.jar* exportiert. Die Programmausführung beginnt mit einem Auswahldialog (siehe Bild 4), über den der Fokus der Benutzeroberfläche ausgewählt werden kann. Dies ist nötig, da sich die Oberflächen für die Analyse von Präfixen und Suffixen unterscheiden und die Beschriftungen der Fenster nicht nachträglich geändert werden können.

Bei den Programmoberflächen *Suffixe* und *Praefixe* muss jeder einzelne Bearbeitungsschritt vom Benutzer durch Knopfdruck initiiert werden. Die Knöpfe sind mit Zahlen versehen, die die Reihenfolge der Verarbeitungsschritte anzeigen. Bei der Oberfläche *Suffixe und Präfixe* werden der Einfachheit halber mehrere Schritte zusammengefasst und gleichzeitig die Bäume zur Präfix- und zur Suffixanalyse verarbeitet. Hierbei müssen die Bäume jedoch schon in serialisierter Form vorliegen. Die modularen Oberflächen haben den Vorteil, dass der Benutzer die Zwischenschritte einsehen und prüfen kann. Die Ergebnisse sind somit besser nachzuvollziehen. Als vierte Option kann eine Oberfläche aufgerufen werden, bei der ein nach dem Ukkonen-Algorithmus erstellter Suffixbaum verwendet wird (*Suffixe(Ukkonen)*). Mit diesem Baum ist bisher nur eine Oberfläche für die Untersuchung von Suffixen implementiert.

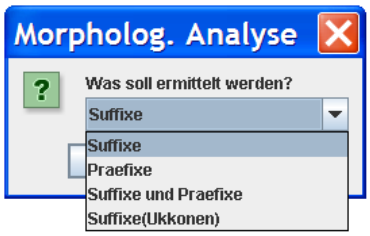

Bild 4: Auswahldialog

4.3.2 Unterschied in der Analyse von Präfixen und Suffixen

Der einzige Unterschied in der Analyse von Präfixen und Suffixen beruht darauf, dass bei der Präfixanalyse jedes Wort gespiegelt wird, bevor es in den Baum eingefügt wird. Dadurch wird erreicht, dass man im Baum nach Blättern suchen kann, aber anstatt der Suffixe die Präfixe der Wörter erhält. Die gefundenen Strings müssen wiederum gespiegelt werden, da sie in den Blattlabels in umgedrehter Form stehen (dies macht das Programm automatisch, bevor die Strings angezeigt werden). Für Präfixe und Suffixe werden unterschiedliche Klassen verwendet, da der Quellcode lesbarer und leichter veränderbar ist, wenn die Variablennamen zur aktuellen Aufgabe passen.

4.3.3 Korpusauswahl

Der Benutzer kann zu Testzwecken manuell einen Text eingeben und daraus einen Baum erzeugen lassen oder über einen File-Dialog die gewünschte Textdatei auswählen, aus der der Baum erstellt werden soll. Durch Betätigen des Knopfes *1) Suffixbaum aus Textdatei* wird der jeweilige Baum erzeugt und im Textfenster dargestellt.

Es ist wichtig zu erwähnen, dass die Bearbeitungszeit zur Erstellung der Bäume bei einer Textdatei von 32 Seiten (Beispieldatei spanisch_groß.txt) je nach Leistungsfähigkeit des Rechners zwischen zehn und 20 Minuten betragen kann. Dies mag sowohl am Erstellungsalgorithmus des Tries liegen als auch an der Programmierweise selbst, denn es gibt Java-Klassen, mit der die Erzeugung von Bäumen schneller implementiert ist. Damit der Leser das Programm auch ohne Wartezeiten ausprobieren kann, gibt es die Möglichkeit, bereits erstellte Bäume zu serialisieren und später wieder einzulesen. Das Serialisieren erfolgt über den Menüpunkt *Datei → Baum spei-*

chern, das Einlesen über den Knopf *1) Gespeicherten Baum einlesen*. Über den Knopf *(Baum zeigen)*, wird der Baum im Textfeld angezeigt (siehe Bild 5).
Außerdem kann jederzeit der aktuell im Textfenster angezeigte Text in einer Datei abgespeichert werden. Dies erfolgt über den Menüpunkt *Datei → Aktuellen Text speichern*.

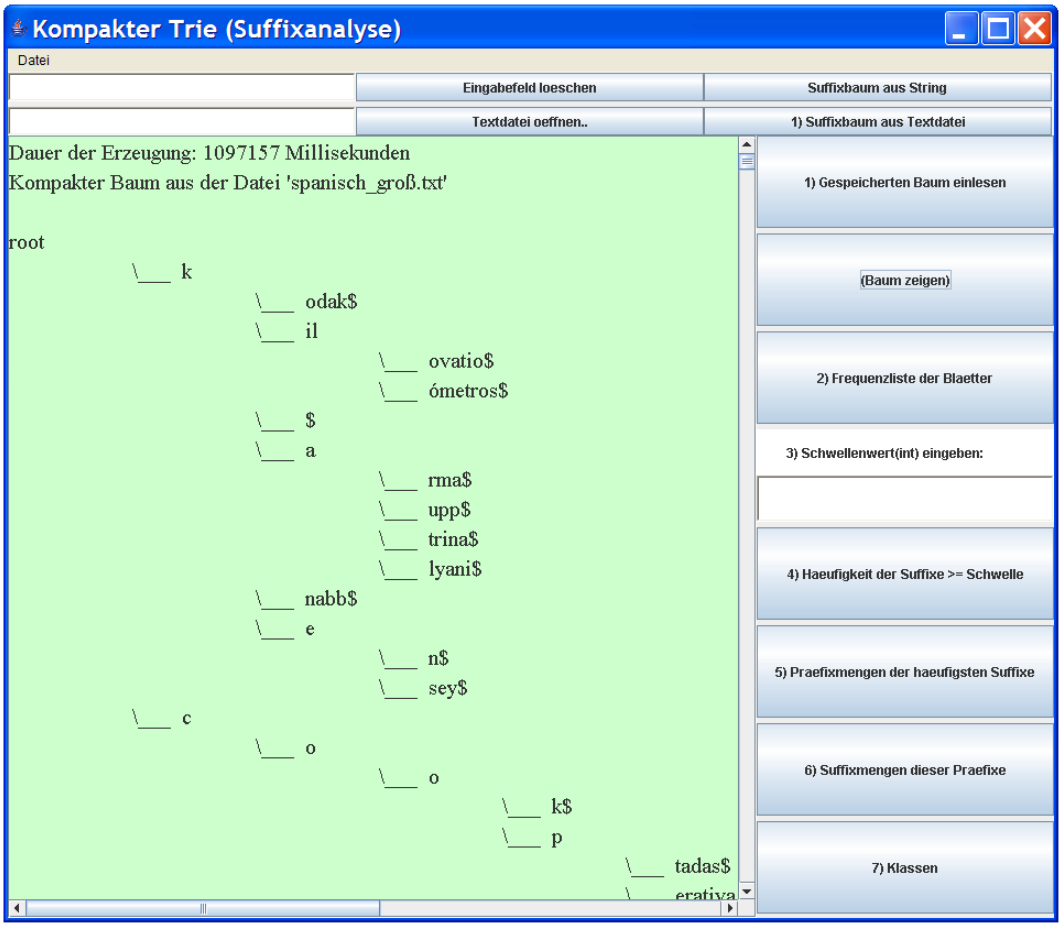

Bild 5: Oberfläche zur Suffix-Analyse

Die Datei, aus der der Baum eingelesen werden soll, muss *Baum_praefixe.tmp* beziehungsweise *Baum_suffixe.tmp* heißen und muss im Programmverzeichnis liegen.
Der Ukkonen-Suffixbaum konnte bisher noch nicht serialisiert werden. Seine Erstellung benötigt bei Verwendung derselben Textdatei aber erheblich weniger Zeit und kann daher auch ohne serialisierten Baum gut getestet werden.

4.3.4 Erstellung einer Frequenzliste aller Blätter im Baum

Unter der Annahme, dass die Häufigkeit des Auftretens eines Blatts im Baum bestimmt, ob das Label dieses Blatts ein guter Kandidat für ein Affix des Spanischen ist, macht es Sinn, die potentiellen Affixe bezüglich ihrer Häufigkeit zu filtern. Dieses Verfahren deckt keine distributionellen Unterschiede der Affixe auf, denn aufgrund der Struktur des Baumes können nur Wort-Types betrachtet werden. Es könnte also sein, dass ein Affix in der Frequenzliste einen relativ hohen Wert aufweist, aber bei Betrachtung des gesamten Textes im Gegensatz zu anderen Wörtern

eher selten vorkommt.
Dies wäre der Fall, wenn das Affix zwar in unterschiedlichen Kontexten auftritt, wovon jedoch jeder einzelne im Text nur selten vorkommt. Ein Affix, dass nur in wenigen Kontexten vorkommt, die aber sehr häufig sind (zum Beispiel in Funktionswörtern), hätte in der Frequenzliste einen niedrigeren Rang, obwohl es häufig im Text zu finden ist.

Nach Betätigen des Knopf *2) Frequenzliste der Blaetter* wird eine absteigend geordnete Frequenzliste angezeigt, in der alle Strings, die als Blattknoten vorkommen, mit ihrer Häufigkeit eingetragen sind. Der Benutzer muss selbst eine Schwelle für die weitere Verarbeitung der Affixe festlegen, indem er zum Beispiel angibt, dass im weiteren Verlauf nur Affixe betrachtet werden sollen, die mindestens zwanzigmal als Blätter im Baum vorgekommen sind. Die Schwelle muss je nach Größe des eingelesenen Textes gewählt werden.
Die Struktur der Bäume macht diese Eingrenzung nötig, denn je seltener ein Wort oder eine morphologische Ausprägung dieses Wortes im Text vorkommt, umso länger sind die Strings, die es im Baum repräsentieren. Gibt es kein anderes Wort im Text, das mit dem Wort einen Präfix-String gemeinsam hat, so steht es komplett als Label eines Blattes unter der Wurzel. Es ist aber nicht sinnvoll, ein ganzes Wort darauf zu untersuchen, ob es ein Affix des Spanischen ist, daher sollen solche Strings vorher aussortiert werden.

Der Schwellenwert für die Weiterverarbeitung der Affixe wird im Textfeld *Schwellenwert(int) eingeben:* angegeben. Durch Betätigen des Knopfs *3) Haeufigkeit der Suffixe >= Schwelle* (bzw. *3) Haeufigkeit der Praefixe >= Schwelle*) werden die Affixe ausgewählt, deren Mindesthäufigkeit im Baum gleich dem Schwellenwert ist. Alle anderen Affixe werden nun nicht mehr betrachtet.

4.3.5 Untersuchung der Affix-Kontexte

Als nächster Schritt (*5) Praefixmengen der haeufigsten Suffixe* bzw. *5) Suffixmengen der haeufigsten Praefixe*) werden zu jedem der ausgewählten Affixe diejenigen Strings im Baum gesucht, die an der Wurzel beginnen und mit diesem Affix enden (siehe Bild 6). Im Struktursystem sind das die Einheiten, die mit den gefundenen Affixen kombinierbar sind. Beispielsweise könnte für das Suffix *-mente* herausgefunden werden, dass es die möglichen Präfix-Strings *clara-* und *aproximada-* hat. Für das Präfix *re-* könnte festgestellt werden, dass es unter anderem mit den Suffix-Strings *-presenta*, *-servado* und *-nombre* kombinierbar ist.
Das Auffinden von paradigmatischen Relationen dient dazu, die Affixe danach zu klassifizieren, auf welche Wortarten sie hindeuten oder um allgemein eine Zusammenstellung der Wörter zu bekommen, die sich aus ihnen bilden lassen. Es gibt zum Beispiel bestimmte Endungen, die immer Verb-Endungen sind und daher in eine Klasse gehören. Andererseits kann man über die damit verbundenen Präfix-Strings Klassen von Wörtern bilden, die vermutlich derselben Wortart angehören. Zunächst wird also für jedes ausgewählte Affix die Menge der im Baum vorangehenden Strings gespeichert.

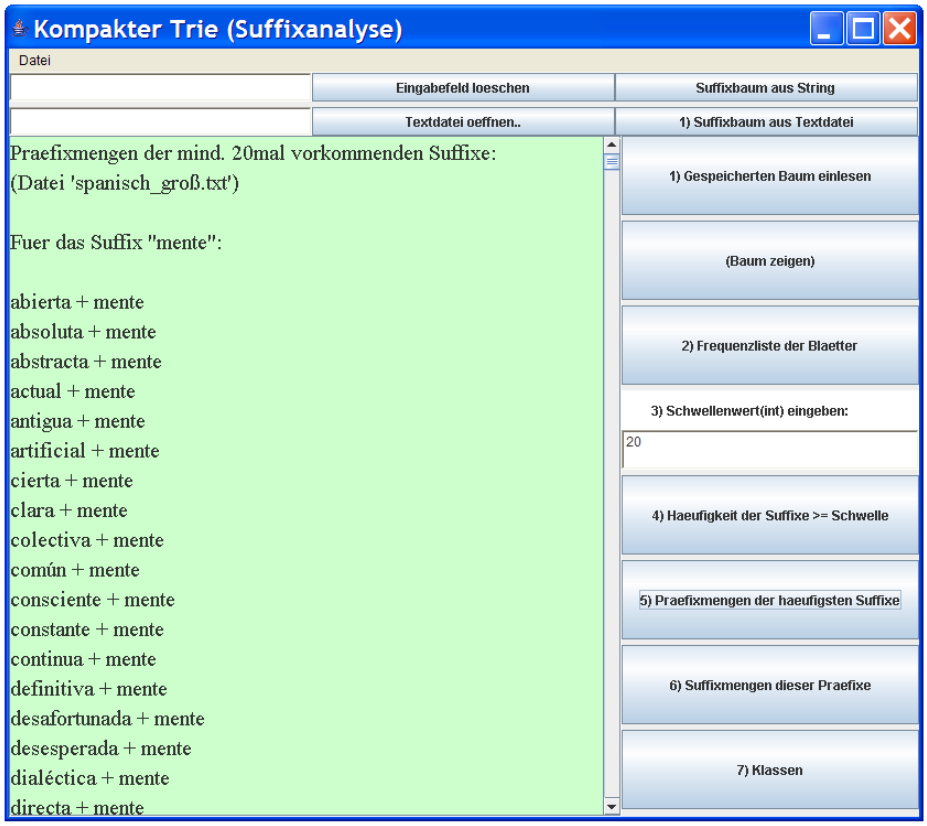

Bild 6: Präfixmengen der Suffixe

4.3.6 Umkehrung der Kontextsuche

Das Verfahren wird im folgenden Schritt *(6) Suffixmenge dieser Praefixe*, für die Präfixanalyse analog) umgekehrt, indem für jeden gefundenden Präfix-String wiederum alle möglichen Endungen im Baum gesucht werden (siehe Bild 7). Diese müssen zusätzlich zu den schon vorher ausgewählten Suffixen gehören, denn sonst würden auch wieder Suffixe betrachtet werden, die absichtlich vorher verworfen wurden. Die potentiellen Stämme sollen nur anhand der Suffixe verglichen werden, die zuvor als Morphemkandidaten bestimmt wurden.

Beispiel Suffix-Suche Für den Stamm *activ-* werden im Baum die Endungen *-a*, *-idad*, *-o* und *-os* gefunden. So erhält man die Suffixparadigmen für die gefundenen Stämme und kann Aussagen darüber treffen, welche Wortarten aus den Stämmen gebildet werden können.

Beispiel Präfix-Suche Für den String *-trás* werden im Baum die Präfixe *a-* und *de-* gefunden, für den String *-duce* werden dieselben Präfixe gefunden. Anhand dieser Daten kann man vermuten, dass *a-* und *de-* tatsächlich Präfixe des Spanischen sind. Allerdings lässt sich dadurch keine Gemeinsamkeit für die Strings *-trás* und *-duce* ableiten (*-duce* ist ein flektiertes Verb, *-trás* nicht).

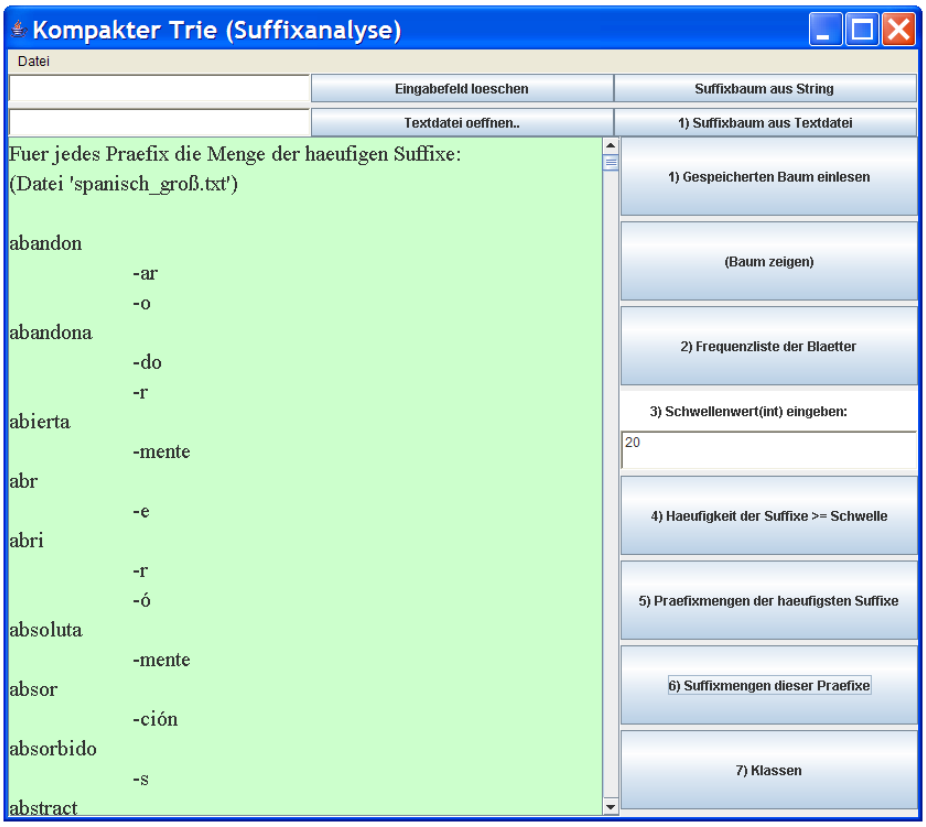

Bild 7: Suffixmengen der Präfixe

4.3.7 Erstellung von Klassen

Im folgenden Schritt (*7) Klassen*) werden die Ergebnisse aus 6) verwendet, um Klassen von Wörtern beziehungsweise Affixen zu erstellen (siehe Bild 8, Bild 9). In der beschriebenen Liste, die zu ausgewählten Strings die möglichen Endungen im Baum enthält, werden diese Endungen (Präfixe oder Suffixe) verglichen. Jede Gruppierung von Endungen ergibt eine Klasse und die zugehörigen Strings bilden somit auch eine Klasse.

Beispiel Die Strings *explic-* und *aclar-* treten im Baum beide mit den Endungen *-a*, *-ar* und *-ó* auf. Daher gibt es eine Klasse von Wörtern, in der *explica*, *explicar*, *explicó*, *aclara*, *aclarar* und *aclaró* enthalten sind sowie eine Klasse von Suffixen, die *-a*, *-ar* und *-ó* umfasst, je nachdem, welche Perspektive man betrachten möchte.

Der Sinn dieses Verfahrens ist, die Wörter in Wortarten zu unterteilen und die Affixe danach zu gruppieren, mit welchen Wortarten sie kombinierbar sind. Da das Programm kein linguistisches Wissen zu Verfügung hat, gibt es auch noch kein Konzept von Wortarten oder gar Namen dafür. Es entstehen einfach Gruppen von Strings, denen man zunächst nicht ansieht, ob die paradigmatische Ähnlichkeit zufällig oder systematisch ist. Erst die Prüfung unter Einbezug der gängigen Definition von Wortarten gibt eine Grundlage zur Bewertung der Ergebnisse.

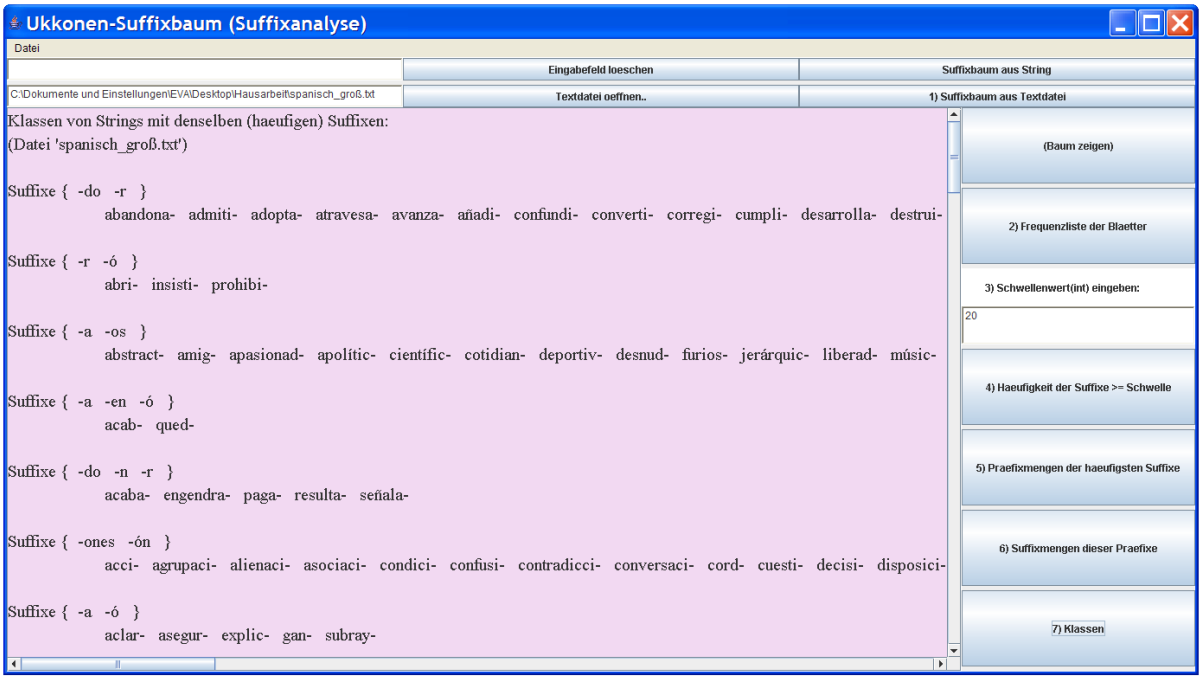

Bild 8: Suffixklassen

Einerseits ist es wichtig, eine neue Untersuchungsmethode zu bewerten, denn es muss ja entschieden werden, ob sie verworfen oder beibehalten wird. Andererseits dürfen möglicherweise neue Erkenntnisse nicht behindert werden, indem bisher vorhandene und für richtig befundene Informationen als gegebene Grundlage betrachtet werden, von der nicht abgewichen werden darf. Vor der Auswertung neuer Ergebnisse muss Klarheit darüber bestehen, ob lediglich das bisher vorhandene Wissen durch eine automatische Analyse bestätigt werden soll oder nach neuem Wissen gesucht wird, das mit dem bestehenden nicht unbedingt konform sein muss.
Es könnten Wortklassen entstehen, wie es sie in der traditionellen Linguistik nicht gibt, die aber trotzdem sinnvoll sind. Nomen und Verben zum Beispiel haben gemeinsam, dass sie flektierbar sind, unterscheiden sich aber in ihrer Funktion (Bezeichnung einer Sache oder einer Tätigkeit). Aufgrund ihrer gemeinsamen Eigenschaft, der Flektierbarkeit, könnten sie auch in eine gemeinsame Klasse aufgenommen werden.

Beipiele zu Suffix-Klassen

1 Suffix(e) { -mente } abierta- absoluta- abstracta- clara- colectiva- común- constante- definitiva- desafortunada- desesperada- dialéctica- directa- doble- efectiva- espiritual- estética- exacta- extremada- final- fuerte- fundamental- igual- inevitable- intelectual- necesaria- precisa- separada- subjetiva- única-

2 Suffix(e) { -a -idad -o -os } activ- negativ- subjetiv-

3 Suffix(e) { -se -te } enrollar- quedar-

4 Suffix(e) { -r -ó } abri- insisti- prohibi-

5 Suffix(e) { -ones -ón } acci- agrupaci- alienaci- asociaci- condici- confusi- contradicci- conversacicord- cuesti- decisi- disposici- diversi- elecci- excarcelaci- excepci- expresi- formaci- frustraciilusi- innovaci- intervenci- manifestaci- mistificaci- ocupaci- opci- oposici- organizaci- pasipoblaci- pretensi- raz- realizaci- rebeli- regulaci- relaci- religi- repercusi- represi- revoluciseparaci- soluci- tradici-

6 Suffix(e) { -as } bols- camarad- fl- liberador- palabr- plac- tercer-

7 ∗ Suffix(e) { -a -o } acordad- afect- afectiv- ajen- anunciad- aprobad- apropiad- arbitrariasegurad- autónom- cerrad- comienz- complet- comprenderl- concret- confiad- cualitativdebid- desesperad- desmistificad- detest- dich- efectiv- expresad- fotográfic- fundamenthacerl- impuest- increment- iniciad- intrínsec- junt- larg- llevad- mediátic- mer- misteriosmágic- místic- necesari- ocupad- opuest- orgiástic- pensad- plen- psicológic- pur- reclamreducirl- rein- respet- retorn- sanitari- secret- sencill- tant- teng- vist- vuelt- vuestr- y-

8 ∗ Suffix(e) { -r -s } acto- alcanza- aprende- baja- dominado- escrito- extraña- liberado- luchameno-

Die erste Klasse enthält Wörter, die auf *-mente* enden, was im Spanischen das Suffixmorphem für Adverben ist. Dieses Morphem ist nicht ambig, deshalb sind alle Wörter dieser Klasse tatsächlich Adverben und das gefundene Morphem deckt sich mit dem, was die Sprachwissenschaft als Morphem betrachtet. Bei der zweiten Klasse fällt auf, dass die Stämme mit den angegebenen Endungen sowohl zu Nomen als auch zu Adjektiven kombinierbar sind.
Actividad, *negatividad* und *subjetividad* sind Nomen, die Stämme *activ-*, *negativ-*, und *subjetiv-* ergeben jeweils mit den Endungen *-a*, *-o*, *-os* Adjektive. Somit wurde zwar keine Unterteilung in Wortarten vorgenommen, aber da alle Stämme sowohl zu Nomen als auch zu Adjektiven gehören, ist die Zusammenstellung nicht zufällig sondern systematisch und zeigt eine wichtige Gemeinsamkeit dieser Wörter.

Die Klassen drei, vier und fünf enthalten jeweils nur Wörter derselben Wortart. In Klasse drei sind es Verb-Infinitive mit angeschlossenen Personalpronomen (reflexive Verben), zum Beispiel *quedar-se* ('bleiben'), in Klasse vier sind es ebenfalls Verben, einmal mit der Infinitiv-Endung (*-r*), einmal mit der Endung für die 3. Person Singular Indefinido (*-ó*). Zur Infinitiv-Endung wird normalerweise der vorangehende Vokal hinzugenommen (hier das *i*), aber aus schon beschriebenen Gründen erzeugt der Baum ein anderes Ergebnis. Klasse fünf enthält Nomen, die im Singular (*-ón*) und Plural (*-ones*) aufgetreten sind. Die Endung *-ones* besteht eigentlich aus zwei Morphemen, *-on* und -es. Da im Plural das *o* aber keinen Akzent hat, gibt es an dieser Stelle im Baum eine Verzweigung, an der der komplette String *-ones* hängt.

Die Endung *-as* in Klasse sechs besteht aus zwei Mophemen, *-a* für weiblich und *-s* für Plural. Welche Endung bei Wörtern, die auf *-as* enden, erkannt wird, hängt davon ab, ob es im Baum noch eine andere Endung am selben Stamm gibt, die mit *a* beginnt. Wäre im spanischen Text neben dem Wort *bolsas* ('Taschen', Plural) auch *bolsa* ('Tasche', Singular) vorgekommen, wäre

nur *-s* als Endung gefunden worden.

Die letzten beiden Klassen sind Beispiele dafür, wie die Sortierung fehlschlagen kann, wenn Suffixmorpheme ambig sind oder im gefundenen Wort das Suffix zwar als String aber nicht als Morphem vorkommt. Die Suffixe aus Klasse sieben (*-a*, *-o*) können entweder Adjektiv- und Nomen-Endungen für männlich und weiblich sein oder auch Verb-Endungen (*-o* → 1. Person Singular Präsens, *-a* → 3. Person Singular Präsens). *Comienzo* und *comienza* sind Ausprägungen des Verbs *comenzar* ('anfangen'), die Wörter *retorno* und *retorna* vom Verb *retornar* ('zurückgeben'). Eine Mischform sind die Wörter *fundamento* ('Fundament') und *fundamenta* ('er begründet'). Die meisten anderen Wörter dieser Klasse sind Adjektive mit männlicher oder weiblicher Endung. Ebenso können die Endungen aus Klasse acht (*-r*, *-s*) entweder Nomen-Endungen sein wie bei *actor* ('Schauspieler') oder Verb-Endungen wie bei *aprender* ('lernen'). Diese beiden Klasse postulieren zwar keine falschen Morpheme, bringen aber auch keine Erkenntnisse über deren unterschiedliche Verwendung.

Die angegebenen Klassen sind bei der Untersuchung des Textes spanisch_groß.txt mit dem Schwellenwert 20 entstanden. Von 87 Klassen haben bei 70 die Suffixe dieselbe Funktion für die Präfix-Strings (z.B. hat *-o* bei allen Wörtern die Funktion 1. Person Singular Präsens), bei 17 Klassen stehen die Suffixe in unterschiedlichem Zusammenhang zu den Präfix-Strings (wie in Beispiel sieben).

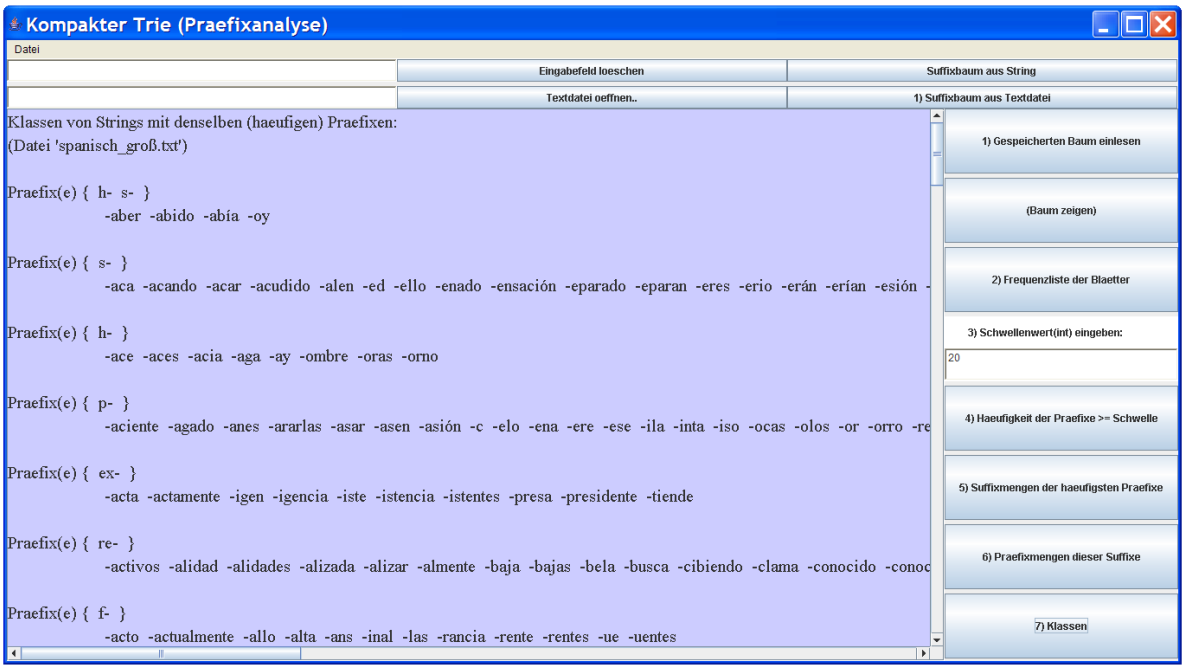

Bild 9: Präfixklassen

Beipiele zu Präfix-Klassen

1 Praefix(e) { ex- } -acta -actamente -igen -igencia -iste -istencia -istentes -presa -presidente -tiende

2 Praefix(e) { re- } -activos -alidad -alidades -alizada -alizar -almente -baja -bajas -bela -busca -cibiendo -clama -conocido -conocimiento -constituyen -construcción -creado -ducción -ferencias -ferente -fuerza -gimiento -ich -ificación -ificada -ina -mitir -ndir -orientación -presentación -produce -producen -sidentes -solver -suma -tira -tirar -torno -unir -unión -vista -vuelta

3 Praefix(e) { con- de- } -cisión -creto -solación -strucción -struir

4 Praefix(e) { a- re- } -cuerdo -cuerdos -nuncian -specto

5 ∗ Praefix(e) { h- s- } -aber -abido -abía -oy

Präfixe sind für die Wortart nicht ausschlaggebend, dasselbe Präfix kann mit Nomen, Verben, Adjektiven usw. auftreten (Nomen: *con-strucción*, Verb: *con-struir*, Adjektiv: *con-creto*). Durch Erkennung der Präfixe lassen sich also keine Mengen von Wörtern derselben Wortart erschließen. Die Untersuchung der Präfixe dient lediglich dazu, eine allgemeine Zusammenstellung der spanischen Präfixe und der Wörter, in denen sie vorkommen, zu erhalten.

Die Beipiele eins bis vier zeigen Präfixe, die Morpheme des Spanischen sind. Die Wörter, in denen sie auftreten, sind Nomen, Verben und Adjektive. Die Präfixe in Beispiel fünf sind keine Morpheme des Spanischen. Es ist Zufall, dass die Suffix-Strings, mit denen sie auftreten, sowohl mit dem Präfix-String *h-* auch als mit *s-* verbunden ein Wort ergeben. Da es eine Vielzahl von Wörtern gibt, die sich nur im Anfangsbuchstaben unterscheiden, ohne dass dieser ein Morphem ist, ist der Anteil dieser "schlechten" Klassen sehr hoch. Mit demselben Text als Grundlage wie bei der Suffixanalyse und einem Schwellenwert von 20 ergeben sich 59 Präfixklassen, von denen nur 20 Klassen Präfixmorpheme enthalten und 39 ähnlich der Klasse aus Beispiel fünf sind.

Es ist schwierig zu bestimmen, ob ein gefundener Präfix-String tatsächlich ein Morphem ist. Bei Wörtern, die auch ohne das Präfix eine Bedeutung haben, könnte versuchen werden, dieses Wort im Korpus zu finden. Dann wäre die Wahrscheinlichkeit hoch, dass ein richtiges Präfix erkannt worden ist.

Beispiel *Re-baja* bedeutet 'Ermäßigung', *baja* bedeutet 'klein'.

Umgekehrt bedeutet es aber nicht, dass ein Präfix-String kein Morphem ist, nur weil das Wort ohne das Präfix keine Bedeutung hat. Bei Wörtern mit lateinischem Ursprung kann es sein, dass nur das affigierte Wort ins Spanische übernommen wurde, nicht aber das Grundwort. Für viele der Präfix-Strings lässt sich diese Frage ohne linguistisches Wissen nicht klären.

Beispiel *Re-ferente* bedeutet 'betreffend', *ferente* oder *fere* (von lat. *ferre* für 'tragen') hat keine Bedeutung im Spanischen.

4.3.8 Suffixanalyse mit dem Ukkonen-Suffixbaum

Zur Erstellung des Suffixbaums nach dem Ukkonen-Algorithmus verwendet das Programm die Klasse *UkkonenSuffixTree* aus der *BioJava*-Bibliothek, Version 1.4 (Meloan, 2004, [6]). Es wur-

den geringfügige Änderungen an der Klasse vorgenommen, die im Quellcode vermerkt sind. Bei dieser Biliothek handelt es sich um ein open-source Projekt für biotechnologische Anwendungen. Es wurde 1998 von Matthew Pocock und Thomas Down vom Sanger Institute für Genforschung in Cambridge initiiert und umfasst mittlerweile über 1200 Klassen, mit deren Hilfe das Problem der ständig wachsenden Datenmenge in der Genforschung bewältigt werden soll.

Die Erzeugung des Suffixbaums ist mehr als dreimal so schnell wie die des Tries mit den selbstgeschriebenen Klassen *Tree_praefixe.java* und *Tree_suffixe.java*. Allerdings bewirken die zusätzlichen Informationen im Suffixbaum gegenüber dem Trie, dass bei der Erkennung der Suffixe leichte Abweichungen entstehen. Grund dafür ist, dass durch zusätzliche Eintragungen in den Baum - alle Suffixe eines einzufügenden Wortes - mehr Verzweigungen entstehen. Die Unterschiede lassen sich am besten an der Frequenzliste erkennen und je nach Wahl des Schwellenwertes können die Endergebnisse, also die Klassen, ebenso abweichen.

Die Verwendung des Suffixbaums anstelle des kompakten Tries erfordert einige Modifikationen bei der Suffixanalyse. Zunächst muss eine Liste aller im Text vorkommenden graphischen Wörter erstellt werden. Durch den Abgleich mit dieser Liste können bei der Kontextuntersuchung der gefundenen Suffixe die Ergebnisse gefiltert werden. Es werden dann nur diejenigen Vorgängerstrings der Suffixe betrachtet, die mit dem Suffix konkateniert ein Wort aus der Liste ergeben. So wird vermieden, dass Strings weiterverarbeitet werden, die im Spanischen keine Bedeutung haben, sondern nur aufgrund der Struktur des Baums vorhanden sind.

Der Abgleich mit dieser Wortliste muss auch bei der Erzeugung der Frequenzliste erfolgen. Würde jedes im Baum vorkommende Suffix gezählt werden, würden Suffixe von langen Wörtern bezogen auf die Häufigkeit bevorzugt, da sie automatisch häufiger vorkommen. Liegt ein Suffix mit drei Buchstaben vor, das in einem Wort mit insgesamt zehn Buchstaben auftritt, würde das Suffix an acht Stellen im Baum vorkommen, bei einem Wort mit insgesamt sechs Buchstaben käme es nur viermal vor. Um dieses Ungleichgewicht zu verhindern, müssen die Suffixe bei der Zählung ebenfalls mit der Wortliste abgeglichen werden, damit nur Suffixe eines vollständigen Wortes in die Zählung mit eingehen.

4.4 Fazit und Ausblick

Das hier vorgestellte Programm ermöglicht es, anhand eines relativ kleinen spanischen Textes einige Informationen über die spanische Morphologie zu erhalten, genauer über Suffixe und Präfixe. Um die Analyse fortzuführen, könnte auch innerhalb des Baums nach signifikanten Knoten gesucht werden, um weitere Morpheme des Spanischen zu erkennen. Beispielsweise könnte es sinnvoll sein, nach Knoten zu suchen, die sich in auffällig viele Kanten verzweigen, da dies ebenfalls auf Morphemgrenzen hindeuten könnte. Zusammen mit den gefundenen Suffixen und Präfixen würde sich so die komplette morphologische Zerlegung eines Wortes ergeben.

Neben wichtigen Suffixen und Präfixen des Spanischen, die im Baum erkannt werden, werden aber auch solche Affixe ausgegeben, die mit den bisherigen morphologischen Regeln nicht konform sind. Dies ist prinzipiell nicht falsch, denn diese Regeln beschreiben lediglich die Phänomene, schreiben aber die Bildung von sprachlichen Einheiten nicht vor. Daher ist es auch möglich, neue Elemente in das Inventar der sprachlichen Einheiten aufzunehmen, solange sie die Phänomene nicht schlechter beschreiben als die bisherigen. Wie und ob die neugewonnenen Informationen Verwendung finden können, muss in einem nächsten Untersuchungsschritt geklärt werden.

Problematisch kann allerdings sein, dass die Ergebnisse des Programms stark vom Text abhängen, aus dem der Baum erzeugt wurde. Je nachdem, welche Ausprägungen eines Verbs im Text vorkamen, wird bei Verben mit der Endung *-ar* entweder *-r* oder *-ar* als Suffix erkannt.

Beispiel Suffix(e) { **-ar** -ó } analiz- apost- señal-
vs. Suffix(e) { -ndo **-r** } apoya- ataca- baila- busca- cambia- combina-

Außerdem können für dasselbe Verb unterschiedliche Zerlegungen erzeugt werden, wenn an zwei verschiedenen Stellen des Verbs im Baum ein Blatt abzweigt.

Beispiel Suffix(e) { -do -n -r } acaba- da- engendra- paga- **resulta-** señala-
vs. Suffix(e) { -a -ar -en } **result-** tom-

Die Ergebnisse können also widersprüchlich sein, wodurch sich die generelle Frage der Verlässlichkeit der Informationen stellt. Wenn die Ergebnisse trotzdem benutzt werden sollen, müssen Kriterien festgelegt werden, nach denen widersprüchliche Informationen gefiltert werden.

Generell ist der Ansatz der Analyse mit Bäumen aber sehr interessant, da die Wörter in eine sinnvolle und übersichtliche Ordnung gebracht werden, die die zugrundeliegende Struktur zugänglicher macht. Durch eine Verfeinerung des Verfahrens mit dem Ziel der Eindeutigkeit der Wortzerlegung ließe sich die Qualität der Ergebnisse noch verbessern.

Literatur

[1] *Trie.* 25. März 2006 <http://en.wikipedia.org/wiki/Trie>.

[2] Gusfield, Dan(1999) *Algorithms on Strings, Trees and Sequences,* Cambridge: Cambridge University Press.

[3] Nelson, Mark (1996), "Fast String Searching With Suffix Trees." *Dr. Dobb's Journal*, August 1996.

[4] "Strukturalismus." Eagleton, T. (1994) *Einführung in die Literaturtheorie,* Stuttgart <http://www.uni-essen.de/einladung/Vorlesungen/methoden/strukturalismus.htm>

[5] *Strukturalismus.* 28. März 2006 <http://de.wikipedia.org/wiki/Strukturalismus>.

[6] Meloan, Steven "BioJava – Java Technology Powers Toolkit for Deciphering Genomic Codes", Juni 2004 <http://java.sun.com/developer/technicalArticles/javaopensource/biojava/>.