2.5 Baumbank-Trainingsverfahren

Das sogenannte *Baumbank-Trainingsverfahren* ist ein einfaches, nicht-iteratives Trainingsverfahren für eine probabilistische kontextfreie Grammatik. Das Baumbank-Trainingsverfahren ist eine Maximum-Likelihood-Estimierung mit annotierten Daten, wobei die Regelwahrscheinlichkeiten der gegebenen kontextfreien Grammatik als Parameter aufgefasst und estimiert werden. Das Baumbank-Trainingsverfahren setzt die Existenz einer sogenannten *Baumbank* voraus, d.h. eines grossen Korpus von manuell annotierten Sätzen. Gemäss der Maximum-Likelihood-Estimierung werden die Regelwahrscheinlichkeiten so gewählt, dass die Wahrscheinlichkeit der Baumbank maximiert wird.

Das Baumbank-Trainingsverfahren hat einige sehr gute Eigenschaften. Der eine Grund ist, dass das Baumbank-Trainingsverfahren als Maximum-Likelihood-Estimierung deren gute stochastische Eigenschaften erbt. Der andere Grund ist, dass die mit dem Baumbank-Trainingsverfahren trainierten Grammatiken in der Regel hervorragende linguistische Eigenschaften besitzen. Die Ursache hierfür ist, dass die Annotationen aus den manuell disambiguierten Syntaxbäumen aller Sätze eines grossen Textkorpus bestehen. Dies hat zwei Vorteile. Erstens ist in den Annotationen ein umfangreiches linguistisches Expertenwissen kodiert, welches direkt in die maschinell von der Baumbank abgelesenen Grammatikregeln zurückfliessen kann. Zweitens können die Vorkommenshäufigkeiten der Grammatikregeln direkt in der gegebenen Baumbank gezählt werden, was besonders gute, an diese Baumbank angepasste, Regelwahrscheinlichkeiten produziert. Beides führt dazu, dass eine linguistische Evaluierung einer so hergestellten probabilistischen Grammatik sehr gute Ergebnisse zeigt, vorausgesetzt, dass der Testkorpus der Baumbank ähnlich ist.

In der Abbildung 2.7 ist das Baumbank-Trainingsverfahren schematisch dargestellt. Ein- und Ausgabe-Daten wurden mit dem Symbol dargestellt, Algorithmen hingegen mit dem Symbol . Die Abbildung orientiert sich an der, in Charniak (1996) vorgestellten Vorgehensweise. Interessant daran ist, dass Charniak eine gegebene Baumbank nicht allein zum Training einer gegebenen kontextfreien Grammatik benutzt, sondern in einem vorangehenden Schritt die kontextfreie Grammatik unter Benutzung der gegebenen Baumbank herstellt. Der Vorteil dieser Vorgehensweise liegt klar auf der Hand: eine kontextfreie Grammatik muss nicht erst in langwieriger, manueller und deshalb teurer Entwicklungsarbeit hergestellt werden, sondern kann fast ohne Kosten maschinell direkt von der Baumbank abgelesen werden. Der Aufwand der Annotionsarbeit zur Herstellung der Baumbank wird daher teilweise durch die eingesparte Arbeit für die Grammatikentwicklung kompensiert. Ein zweiter Vorteil ist, dass die so abgeleitete Grammatik besonders gut an die gegebene Baumbank angepasst ist. Im folgenden werden die fünf Algorithmen kurz besprochen:

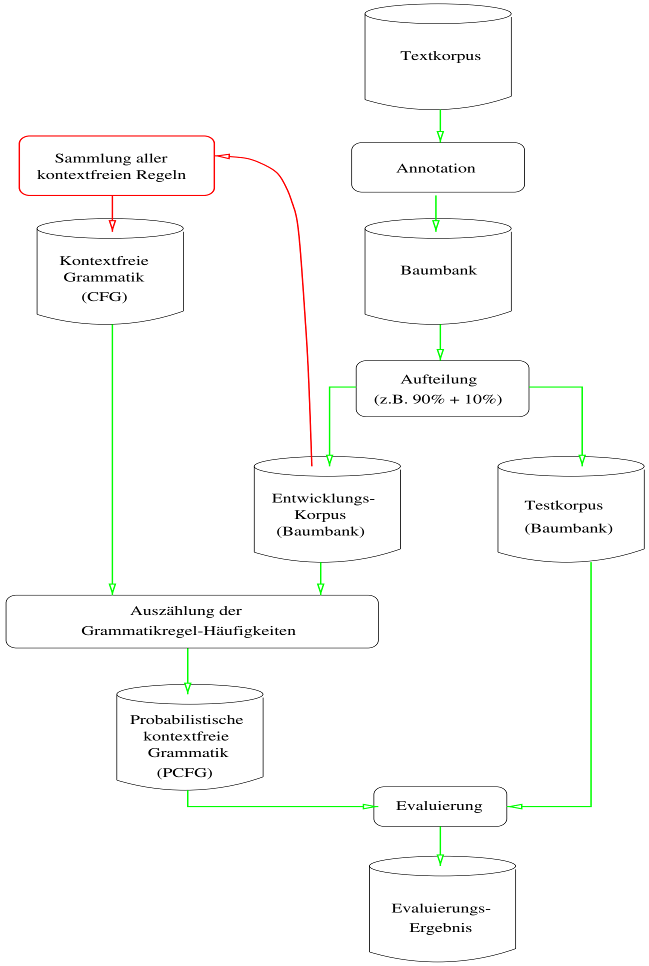

Abbildung 2.7: Herstellung und Training einer kontextfreien Grammatik mit einer Baumbank

- *Annotation* des Textkorpus; das Resultat ist eine *Baumbank*,
- *Aufteilung* der Baumbank in einen *Entwicklungskorpus* und einen *Testkorpus*,
- *Sammlung* aller kontextfreien Regeln des Entwicklungskorpus; das Resultat ist eine *kontextfreie Grammatik*,
- *Auszählung* der Vorkommenshäufigkeiten der Grammatikregeln im Entwicklungskorpus; das Resultat ist eine *probabilistische kontextfreie Grammatik*,
- *Evaluierung* der trainierten probabilistischen kontextfreien Grammatik mit Hilfe des Testkorpus; das Resultat ist ein *Evaluierungsergebnis*.

Die sogenannte *Annotation* eines gegebenen Textkorpus besteht in der Aufgabe, alle Sätze dieses Korpus linguistisch zu analysieren, sodass jeder Satz exakt eine Analyse aufweist. Da diese Aufgabe maschinell nicht in ausreichender Güte vorgenommen werden kann, wird sie in der Regel wenigstens teilweise manuell vorgenommen: Ein menschlicher Annotator hat die Aufgabe, die möglichen Syntaxbäume eines jeden Satzes aus dem Textkorpus (gegebenenfalls mit maschineller Hilfe) zu generieren und zu disambiguieren. Leider ist dies für Sätze mit vielen möglichen Analysen schwer und oft nicht eindeutig lösbar. Daher sind die Annotationen generell nur unter grossem Zeitaufwand zu realisieren. Aus diesem Grund wird der zu annotierende Korpus in der Regel in mehrere Sektionen eingeteilt, sodass die Annotationsarbeit mit parallel zueinander arbeitenden Annotierern geschehen kann. Leider hat dies zur Folge, dass die Qualität der Annotation von den Sektionen stark abhängig ist, sodass Konsistenz oder Homogenität der Baumbank zu einem Problem wird. Zusammenfassend muss man leider konstatieren, dass Baumbanken in der Regel nicht zugleich umfangreich, konsistent und billig sein können. Die wohl am meisten benutzte Baumbank ist die *Penn Wall Street Journal Treebank* (Marcus et al. 1993). Sie ist für Mitglieder des Linguistic Data Consortium (LDC 1993) erhältlich und ist eine Sammlung diverser englisch-sprachiger Texte (ca. 1 Million Wörter).

Die *Aufteilung* der Baumbank in einen *Entwicklungskorpus* und einen *Testkorpus* bereitet den Test vor, wie gut die trainierte Grammatik arbeitet. Damit der Test fair ist, sollten die Testdaten sogenannte *ungesehene Daten* sein. Diese Terminologie bringt zum Ausdruck, dass die Trainingsdaten *gesehene Daten* für ein maschinelles Lernverfahren sind. Natürlich ist es nicht uninteressant, wie ein maschinelles Lernverfahren auf gesehenen Daten arbeitet, d.h. die gesehenen Daten gelernt hat. Die wichtigere Eigenschaft ist aber seine Fähigkeit zur *Generalisierung*, d.h. die Fähigkeit mit Hilfe der gesehenen Daten, neue ungesehene Daten richtig beurteilen zu können. Ein Phänomen, das an dieser Stelle zu nennen ist, ist das sogenannte *Übertraining*. Dies bedeutet, dass ein System auf den gesehenen Daten sehr gute Evaluierungsergebnisse aufweist, jedoch auf den ungesehenen Daten sehr

schlechte. Übertraining erklärt man damit, dass das System ein sehr feines Wissen über die gesehenen Daten gelernt hat, durch eine Überspezialisierung aber leider nicht mehr fähig ist, neue Daten richtig beurteilen zu können[15]. In der Regel wird die Aufteilung der Baumbank in Trainings- und Testdaten so vorgenommen, dass man 90% Trainingsdaten und 10% Testdaten erhält. Natürlich sollten die Testdaten zwar unabhängig von den Trainingsdaten sein, aber diesen auch nicht zu unähnlich sein. Eine übliche Prozedur, um dies zu erreichen, ist es, jeden zehnten Satz der Baumbank als Testdatum zu nehmen. Aus der Penn Wall Street Journal Treebank werden in der Regel die Sektionen 02-21 (ca. 1 Million Wörter) zum Training benutzt und die Sektion 23 (ungefähr 50000 Wörter) zum Testen. Bemerkenswert ist, dass die Testdaten im Gegensatz zu den Trainingsdaten niemals manuell gesichtet werden sollten. Der Grund ist, dass eine manuelle Sichtung der Testdaten dazu führen kann, dass das System während der Entwicklungsphase, unter impliziter Benutzung der Testdaten, dahingehend getunt wird, die Testdaten richtig zu bearbeiten. Eine sicherlich unerwünschte Folge dieses Tunings ist aber, dass das System nicht mehr generalisieren kann. Ein wissenschaftlicher Betrug liegt dann vor, wenn nur getunte Ergebnisse reportiert werden. Daher wird während der Systementwicklung oft ein sogenannter *Entwicklungs-Testkorpus* zur Evaluierung benutzt. Nach vollendeter Entwicklung wird das System abschliessend auf den Testdaten evaluiert. Der Entwicklungs-Testkorpus fehlt in Abbildung 2.7, da in diesem Abschnitt das Baumbank-Trainingsverfahren in seiner einfachsten Form vorgestellt wird, welches von Natur aus kein Tuning zulässt. In späteren Arbeiten weicht Charniak aber von diesem einfachen Baumbank-Trainingsverfahren ab und zieht sehr hilfreiche Wortstatisken (Charniak 1997) hinzu, oder benutzt komplexere Parameter-Estimierungsverfahren (Charniak 2000). Aus der Penn Wall Street Journal Treebank wird in der Regel die Sektion 24 als Entwicklungs-Testkorpus benutzt (Charniak (1997), Collins (1996), Magerman (1995)).

Die Evaluierung einer Grammatik ist in der Regel eine sehr schwierige Aufgabe. Es wäre schön, wenn es für diese Aufgabe eine standardisierte Vorgehensweise gäbe. Dies ist zum Beispiel in der Sprachmodellierung der Fall, in der sich die Korpusperplexität, gemessen auf einem Testkorpus (siehe Abschnitt 2.3), als Standard-Evaluierungsmetrik durchgesetzt hat. Obwohl bekannt ist, dass ein Sprachmodell stets in einer Anwendung getestet werden muss, kursieren "Pi-mal-Daumen"-Regeln der folgenden Art: eine Verbesserung von 5% Korpusperplexität ist für die Anwendung "uninteressant", 10-20% ist "bemerkenswert", und 30% ist "signifikant" (Rosenfeld 2000). Obwohl es alte Ansätze gibt, probabilistische kontextfreie Grammatiken mit Sprachmodellen zu mischen und die Güte dieser hybriden Modelle per Korpusperplexität zu evaluieren (Carroll 1995), werden probabilistische

[15] Das folgende Sprichwort beschreibt Übertraining gut: *Er sieht den Wald vor lauter Bäumen nicht!*

kontextfreie Grammatiken, anders als in der Sprachmodellierung, nicht mittels der Korpusperplexität evaluiert. Die Standard-Vorgehensweise ist, den Testkorpus mit Hilfe der trainierten Grammatik zu parsen und den jeweils wahrscheinlichsten Syntaxbaum (den Viterbi-Parse) mit dem im Testkorpus annotierten Syntaxbaum zu vergleichen. In Abschnitt 2.6 wird gezeigt, wie eine probabilistische kontextfreie Grammatik benutzt werden kann, um Syntaxbäume zu erzeugen und zu disambiguieren. An dieser Stelle soll auf die Vielzahl der Methoden verwiesen werden, mit der zwei Syntaxbäume miteinander verglichen werden. Dem Autor sind die folgenden Evaluierungsmetriken bekannt:

- Die PARSEVAL-Metriken:
  (Labelled) Precision, *(Labelled) Recall*, und *Crossing Brackets*,
- Akkuratheit für *Subkategorisierungsrahmen*,
- Akkuratheit für *Nomen-, Verb-, und Präpositional-Chunks*,
- Akkuratheit für *Exact Match Task*,

Die PARSEVAL-Metriken sind ein Ergebnis der PARSEVAL-Initiative (Black 1992): *Precision* ist der Prozentanteil der *Klammern*[16] im Parse, die auch im annotierten Parse zu finden sind. *Recall* ist der Prozentanteil der Klammern im annotierten Parse, die auch im Parse zu finden sind. *Labelled* heisst, dass hierbei die Grammatikkategorien mit berücksichtigt werden. *Crossing Brackets* ist der Prozentanteil der Klammern, die sich in beiden Parses überschneiden. Diese Masse sind vor allem im Verlauf des noch offenen Wettbewerbs, den besten stochastischen Parser für die Penn Wallstreet Journal Treebank zu bauen, weiter verfeinert worden. Auf der Homepage von Collins und Sekine (2001) kann Software für eine Evaluierung mit den verfeinerten PARSEVAL-Metriken heruntergeladen werden. Die besten stochastischen Parser liefern für alle Metriken weit über 90% Precision/Recall. Akkuratheit für *Subkategorisierungsrahmen* und für *Nomen-, Verb-, und Präpositional-Chunks* misst den Prozentanteil der korrekt erkannten Subkategorisierungsrahmen und Chunks (Abney 1996) im Parse. Evaluierungsergebnisse guter stochastischer Parser liegen typischerweise im Bereich von 80-90% (Beil et al. 1999). Der *Exact Match Task* ist ein sehr hartes Mass, weil es die völlige Übereinstimmung beider Analysen verlangt (Johnson et al. 1999). Soweit dem Autor bekannt ist, wurde es bisher nicht für kontextfreie Grammatiken mit im Durchschnitt vielen tausend Analysen, sondern nur für Sätze mit extrem wenigen, durchschnittlich ca. zehn Analysen angewandt. Die besten Ergebnisse liegen bei ca. 30% (Bod 2000), bzw. im Bereich von 55-75% (Johnson et al. (1999), Riezler et al. (2000)).

[16] Ein Parse wird hierbei in der gewöhnlichen Klammerschreibweise der Linguisten dargestellt.

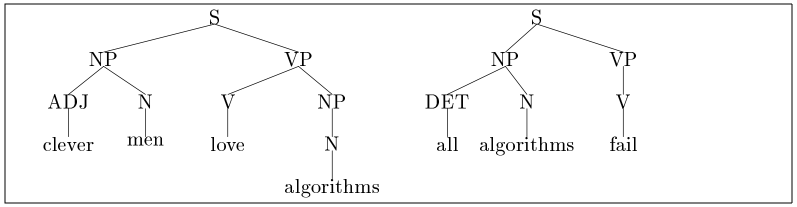

Abbildung 2.8: Beispiel-Baumbank.

$$
\begin{array}{rcl}
S & \rightarrow & NP\ VP \\
NP & \rightarrow & ADJ\ N \\
NP & \rightarrow & DET\ N \\
NP & \rightarrow & N \\
VP & \rightarrow & V\ NP \\
VP & \rightarrow & V \\
DET & \rightarrow & all \\
ADJ & \rightarrow & clever \\
N & \rightarrow & men \\
N & \rightarrow & algorithms \\
V & \rightarrow & love \\
V & \rightarrow & fail
\end{array}
$$

Abbildung 2.9: Aus der Spielzeug-Baumbank in Abbildung 2.8 abgelesene kontextfreie Grammatik.

Die *Sammlung* aller kontextfreien Regeln des Entwicklungskorpus dient in erster Linie dazu, eine kontextfreie Grammatik maschinell zu konstruieren und damit teure manuelle Entwicklungskosten einzusparen. Weitere Vorteile dieser Vorgehensweise sind, dass die Annotionskosten für die Herstellung der Baumbank teilweise kompensiert werden und die so abgeleitete Grammatik gut an die gegebene Baumbank angepasst ist. Der Sammelvorgang ist sehr einfach. Abbildung 2.8 zeigt eine Spielzeug-Baumbank mit zwei Syntaxbäumen. In Abbildung 2.9 werden alle kontextfreien Grammatikregeln gezeigt, die in einem der beiden Syntaxbäume vorkommen. Die aufgesammelten Regeln bilden eine kontextfreie Grammatik, die sogenannten *Baumbank-Grammatik*, mit welcher neue Sätze, zum Beispiel:

all men fail

abgeleitet werden können. Durch das Ablesen/Aufsammeln der Grammatikregeln von der Baumbank wurde also etwas gelernt. Das Aufsammeln der Grammatikregeln ist daher ein eigenständiges symbolisches Lernverfahren, das als Teil des im folgenden zu erläuternden stochastischen Lernverfahrens anzusehen ist. Der hier vorgestellte einfache Sammelvorgang wurde zuerst von Charniak (1996) auf ca. 300000 Wörter (grob geschätzt: 15000 Sätze) der Penn Wallstreet Journal Treebank angewandt, und ergab eine Baumbank-Grammatik mit ca. 10000 Grammatikregeln, von denen ca. 4000 Grammatikregeln mehr als einmal vorkamen[17].

Die *Auszählung* der Vorkommenshäufigkeiten der Grammatikregeln im Entwicklungskorpus ergibt eine *probabilistische kontextfreie Grammatik*. Der Schlüsselschritt ist überaus einfach: In Abschnitt 2.2 wurden probabilistische kontextfreie Grammatiken eingeführt. Die Standard-Nebenbedingung von Booth und Thompson (1973) für diese Grammatiken ist, dass die Grammatikregel-Wahrscheinlichkeiten auf allen Äquivalenzklassen G_A einer bestimmten Grammatikregelrelation eine Wahrscheinlichkeitsverteilung induzieren sollen:

$$\sum_{r \in G_A} p(r) = 1 \ .$$

Dabei gilt (siehe Abschnitt 2.2):

$$r \sim r' \quad :\Leftrightarrow \quad \mathrm{lhs}(r) = \mathrm{lhs}(r') \ ,$$

sowie (für geeignete Grammatikkategorien A):

$$G_A = \{ r \in G \mid \mathrm{lhs}(r) = A \} \ .$$

[17] Müller (2001a) stellt eine sehr interessante Technik vor, in der die Regeln nicht nur aus einer gegebenen Baumbank aufgesammelt werden, sondern durch zusätzliche linguistische Informationen angereichert werden, mit dem Ziel, in der Anwendung die Disambiguierung alternativer Analysen zu unterstützen. Es wird experimentell gezeigt, dass die besten angereicherten Grammatiken gegenüber der "reinen Baumbank-Grammatik" eine verbesserte linguistische Performanz aufweisen und weniger Trainingsmaterial benötigen.

S	$\rightarrow$	$NP\ VP$	(1.0)
NP	$\rightarrow$	$ADJ\ N$	(0.33333)
NP	$\rightarrow$	$DET\ N$	(0.33333)
NP	$\rightarrow$	N	(0.33333)
VP	$\rightarrow$	$V\ NP$	(0.5)
VP	$\rightarrow$	V	(0.5)
DET	$\rightarrow$	*all*	(1.0)
ADJ	$\rightarrow$	*clever*	(1.0)
N	$\rightarrow$	*men*	(0.33333)
N	$\rightarrow$	*algorithms*	(0.66666)
V	$\rightarrow$	*love*	(0.5)
V	$\rightarrow$	*fail*	(0.5)

Abbildung 2.10: Aus der Spielzeug-Baumbank in Abbildung 2.8 abgelesene und trainierte probabilistische kontextfreie Grammatik.

In Abschnitt 2.2 wurde darauf hingewiesen, dass die Standard-Nebenbedingung weder notwendig noch hinreichend dafür ist, dass eine probabilistische kontextfreie Grammatik konsistent ist, d.h. eine Wahrscheinlichkeitsverteilung auf der Menge ihrer Syntaxbäume induziert. Befriedigend ist allerdings, dass Booth und Thompson (1973) zeigen konnten, dass eine zusätzliche Nebenbedingung die Konsistenz der Grammatik erzwingt. Ist $f(r)$ die Häufigkeit einer Grammatikregel r in der Baumbank, so kann die von der Baumbank abgelesene kontextfreie Grammatik mit der folgenden Definition zu einer probabilistischen kontextfreien Grammatik gemacht werden:

$$p(r) := \frac{f(r)}{\sum_{r \in G_A} f(r)}, \quad (r \in G_A) \ .$$

Natürlich folgt aus der Definition sehr einfach, dass diese stochastische Grammatik die Standard-Nebenbedingung für kontextfreie Grammatiken erfüllt. Dies ist eine sehr starke Begründung für die Wahl gerade dieser Wahrscheinlichkeitswerte. Bevor weitere gute Eigenschaften dieser Grammatik angeführt werden, soll die Beispiel-Grammatik aus Abbildung 2.9 "trainiert" werden. Da die Baumbank aus lediglich zwei Syntaxbäumen besteht, kann aus ihr wenig stochastische Information gezogen werden. Abbildung 2.10 zeigt, die mit dem Baumbank-Trainingsverfahren trainierte kontextfreie Grammatik. Interessant ist lediglich, dass die Grammatikregel $N \rightarrow$ *algorithms* in der Baumbank zweimal häufiger vorkommt, als die Grammatikregel $N \rightarrow$ *men*. Hieraus folgt zum Beispiel, dass der Satz

clever men fail (0.01388)

genau zweimal unwahrscheinlicher ist, als der Satz

clever algorithms fail (0.02777) .

Die Satzwahrscheinlichkeiten sind in den runden Klammern angegeben und können leicht nachgerechnet werden. Bessere Grammatiken sollten berücksichtigen, wie oft Nomen als Subjekte oder als Objekte in einer Analyse vorkommen: Als Subjekt kommt *algorithms* in der Beispiel-Baumbank genauso oft vor wie *men*. Das sinnvolle Ergebnis dieser differenzierteren Vorgehensweise wäre, dass dann beide Beispielsätze gleich wahrscheinlich sind. Lexikalisierte Grammatiken berücksichtigen in der Regel die Subkategorisierungsrahmen von Verben und bewerten stochastisch, ob Nomen gute oder schlechte Argumente von Verben sind (*men love* versus *algorithms love*). Tatsächlich bringt diese verfeinerte Vorgehensweise ausgezeichnete stochastische Grammatiken hervor ((Carroll und Rooth 1998), (Beil et al. 1999)). Für echte Baumbanken (mit mehr als zwei Analysen) sind die stochastischen Implikationen natürlich wesentlich schlechter zu überblicken, als für die Beispiel-Baumbank. Umso interessanter ist, dass auch für grosse Baumbanken relativ einfach beweisbar ist, dass gerade Baumbank-Grammatiken unter den vielen möglichen probabilistischen Standard-Grammatiken stochastisch optimale Grammatiken sind. Dies wird im folgenden begründet. In der Definition der Grammatikregel-Wahrscheinlichkeiten für die Baumbank-Grammatik kommt auf der linken Seite eine empirische Wahrscheinlichkeit (siehe Abschnitt 2.1) vor:

$$\tilde{p}_A(r) := \frac{f(r)}{\sum_{r \in G_A} f(r)}, \quad (r \in G_A) \ .$$

Für diese empirische Wahrscheinlichkeitsverteilung $\tilde{p}_A(.)$ fungiert die Äquivalenzklasse G_A als Menge der Datentypen. Die von der Baumbank abgelesenen Regeln r mit einer linken Seite $\mathrm{lhs}(r) = A$, zusammen mit ihren Vorkommenshäufigkeiten $f(r)$ in der Baumbank, fungieren als Korpus:

$$f_A : \ G_A \to \mathcal{N}, \ r \mapsto f(r) \ .$$

Die trainierte Baumbank-Grammatik besteht also aus den empirischen Wahrscheinlichkeitsverteilungen $\tilde{p}_A(.)$ auf den Grammatikregel-Teilkorpora $f_A(.)$. In Abschnitt 2.4 wurde darauf hingewiesen, dass die empirische Wahrscheinlichkeitsverteilung die hervorragende Eigenschaft besitzt, die Korpuswahrscheinlichkeit zu maximieren. Übertragen auf die vorliegende Situation, bedeutet dies, dass die Baumbank-Grammatik die Wahrscheinlichkeiten aller Teilkorpora $f_A(.)$ maximiert:

$$\tilde{p}_A = \mathrm{argmax}_p \quad p(f_A) = \prod_{r \in G_A} p(r)^{f_A(r)} \ .$$

Dies ist ein schönes Ergebnis. Wünschenswert wäre aber auch, dass die trainierte Grammatik sogar der ganzen Baumbank (und nicht nur den Teilkorpora f_A) die grösstmögliche

Wahrscheinlichkeit zuweist. Auch dies ist der Fall. Der nun vorgestellte Beweis ist neu und greift auf die zuletzt hergeleiteten Ergebnisse zurück. Eine Baumbank ist ein Korpus $f_{\mathcal{T}}$ von Syntaxbäumen. Die in der Baumbank vorkommenden Syntaxbäume können als Syntaxbäume der von der Baumbank abgelesenen Baumbank-Grammatik G aufgefasst werden. Mit $\mathcal{T} := \mathcal{T}(G)$ gilt somit:

$$f_{\mathcal{T}} : \ \mathcal{T} \to \mathcal{N} \ .$$

Die trainierte Baumbank-Grammatik weist jedem Syntaxbaum $x \in \mathcal{T}$ eine Wahrscheinlichkeit $p(x)$ zu (siehe Abschnitt 2.2). Daher ist die Baumbank-Wahrscheinlichkeit eine gewöhnliche Korpuswahrscheinlichkeit (Abschnitt 2.1) und es gilt:

$$\begin{aligned}
p(f_{\mathcal{T}}) &:= \prod_{x \in \mathcal{T}} p(x)^{f_{\mathcal{T}}(x)} \\
&= \prod_{x \in \mathcal{T}} \left(\prod_{r \in G} p(r)^{f_r(x)} \right)^{f_{\mathcal{T}}(x)} \\
&= \prod_{x \in \mathcal{T}} \prod_{r \in G} p(r)^{f_r(x) \cdot f_{\mathcal{T}}(x)} \\
&= \prod_{r \in G} \prod_{x \in \mathcal{T}} p(r)^{f_r(x) \cdot f_{\mathcal{T}}(x)} \\
&= \prod_{r \in G} p(r)^{\sum_{x \in \mathcal{T}} f_r(x) \cdot f_{\mathcal{T}}(x)} \\
&= \prod_{r \in G} p(r)^{f(r)} \\
&= p(f) \ .
\end{aligned}$$

Die vorletzte Identität folgt, weil die Häufigkeit $f(r)$ einer Grammatikregel r in der Baumbank sich additiv aus ihren Vorkommenshäufigkeiten $f_r(x) \cdot f_{\mathcal{T}}(x)$ in allen Syntaxbäumen x der Baumbank zusammensetzt. Bei typweiser Durchsicht der Baumbank, muss die Häufigkeit $f_r(x)$ der Grammatikregel r im Syntaxbaum x natürlich mit der Häufigkeit $f_{\mathcal{T}}(x)$ des Syntaxbaums multipliziert werden:

$$f(r) = \sum_{x \in \mathcal{T}} f_r(x) \cdot f_{\mathcal{T}}(x) \ .$$

Die letzte Identität der Ableitung folgt mit der Definition der Korpuswahrscheinlichkeit. Interessant ist lediglich die damit verbundene Interpretation: Die Baumbank kann einerseits als Korpus von Syntaxbäumen aufgefasst werden, andererseits aber auch als Korpus von Grammatikregeln. Ein erstes schönes Ergebnis ist, dass die Wahrscheinlichkeit der Baumbank von der gewählten Interpretation unabhängig ist:

$$p(f_{\mathcal{T}}) = p(f) \ .$$

Dieses Ergebnis kann weiter verfeinert werden:

$$\begin{aligned} p(f_{\mathcal{T}}) &= \prod_{r \in G} p(r)^{f(r)} \\ &= \prod_{A} \prod_{r \in G_A} p(r)^{f(r)} \\ &= \prod_{A} \prod_{r \in G_A} p(r)^{f_A(r)} \\ &= \prod_{A} p(f_A) \ . \end{aligned}$$

Das sehr befriedigende Ergebnis ist also, dass die Baumbank-Wahrscheinlichkeit das Produkt der Wahrscheinlichkeiten aller Teilkorpora f_A ist. Natürlich kann dieses Ergebnis (nach dem oben erzielten Teilergebnis) nicht mehr überraschen: Da alle beteiligten Wahrscheinlichkeiten (Baumbank- und Teilkorpora-Wahrscheinlichkeiten) Produkte von Grammatikregel-Wahrscheinlichkeiten sind, und da bei der Aufteilung der Grammatikregeln in Teilkorpora f_A keine Grammatikregel doppelt gezählt wird, muss die soeben bewiesene Faktorisierung gelten. Alles zusammenfassend:

$$p(f_{\mathcal{T}}) = p(f) = \prod_{A} p(f_A) \ .$$

Da ein Produkt maximiert wird, indem alle seine Faktoren maximiert werden, gilt:

$$\begin{aligned} \mathrm{argmax}_p \quad p(f_{\mathcal{T}}) &= \mathrm{argmax}_p \quad \prod_{A} p(f_A) \\ &= \Big\langle \mathrm{argmax}_p \quad p(f_A) \Big\rangle_A \\ &= \langle \tilde{p}_A \rangle_A \ . \end{aligned}$$

Der erste Schlüsselschritt dieses neuen Beweises eines bekannten Resultats ist, dass sich die Baumbank-Wahrscheinlichkeit als Produkt von Wahrscheinlichkeiten gewisser Teilkorpora darstellen lässt. Der zweite Schlüsselschritt ist, dass die empirische Wahrscheinlichkeitsverteilung $\tilde{p}_A(.)$ die Korpuswahrscheinlichkeit $p(f_A)$ unter allen Wahrscheinlichkeitsverteilungen $p(.)$ maximiert. Aus beiden Schritten ergibt sich dann leicht, dass die Baumbank-Wahrscheinlichkeit mit der Baumbank-Grammatik maximiert wird, solange für die Maximierung nur Grammatiken mit Standard-Nebenbedingung (Booth und Thompson 1973) betrachtet werden[18]. Dieses schöne Ergebnis soll diesen Abschnitt beschliessen.

[18]**Bemerkung**: Möglicherweise gibt es probabilistische kontextfreie Grammatiken, die der Baumbank eine höhere Wahrscheinlichkeit zuweisen, als die Baumbank-Grammatik. Offenbar verletzen solche Grammatiken die Standard-Nebenbedingung für probabilistische kontextfreie Grammatiken, welche aber für die Konsistenz von Grammatiken weder notwendig noch hinreichend ist (Booth und Thompson 1973). Daher gilt: ***Möglicherweise gibt es konsistente Grammatiken mit einer höheren Baumbank-Wahrscheinlichkeit, als die Baumbank-Grammatik.*** Es könnte daher eine lohnende Aufgabe sein, nach