Baumbank-Training

Mateusz Jozef Dworaczek

04.06.2007

Inhalt

Baumbank-Training

Baumbank

ein nicht leeres und endliches Korpus von Bäumen

Training

das; -s, -s: planmäßige Durchführung eines Programms von vielfältigen Übungen zur Ausbildung von Können, Stärkung der Kondition u. Steigerung der Leistungsfähigkeit.
Duden - Das Fremdwörterbuch, 8. Aufl. Mannheim 2005 [CD-ROM]

Beispiel: Baumbank

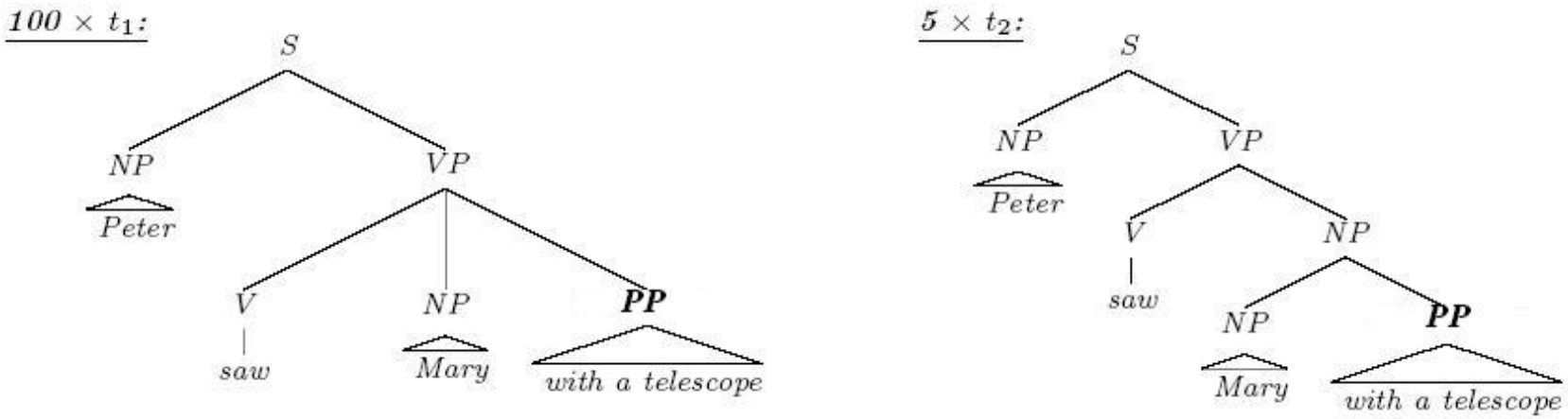


Abbildung: Beispiel:Baumbank

Trainig aus computerlinguistischer Sicht

1. eine große Beispielmenge sammeln;(Korpus)
2. Die Beispielmenge in zwei separate Mengen unterteilen: **Trainingsmenge** und **Testmenge**;

Trainig aus computerlinguistischer Sicht

1. eine große Beispielmenge sammeln;(Korpus)
2. Die Beispielmenge in zwei separate Mengen unterteilen: **Trainingsmenge** und **Testmenge**;
3. Den Lernalgorithmus(z.B.Viterbi) auf die Trainingsmenge anwenden und eine Hypothese h erzeugen;

Trainig aus computerlinguistischer Sicht

1. eine große Beispielmenge sammeln;(Korpus)
2. Die Beispielmenge in zwei separate Mengen unterteilen: **Trainingsmenge** und **Testmenge**;
3. Den Lernalgorithmus(z.B.Viterbi) auf die Trainingsmenge anwenden und eine Hypothese h erzeugen;
4. Den Prozentsatz der Beispiele in der Testmenge messen, die durch h korrekt kassifiziert wurden;

Trainig aus computerlinguistischer Sicht

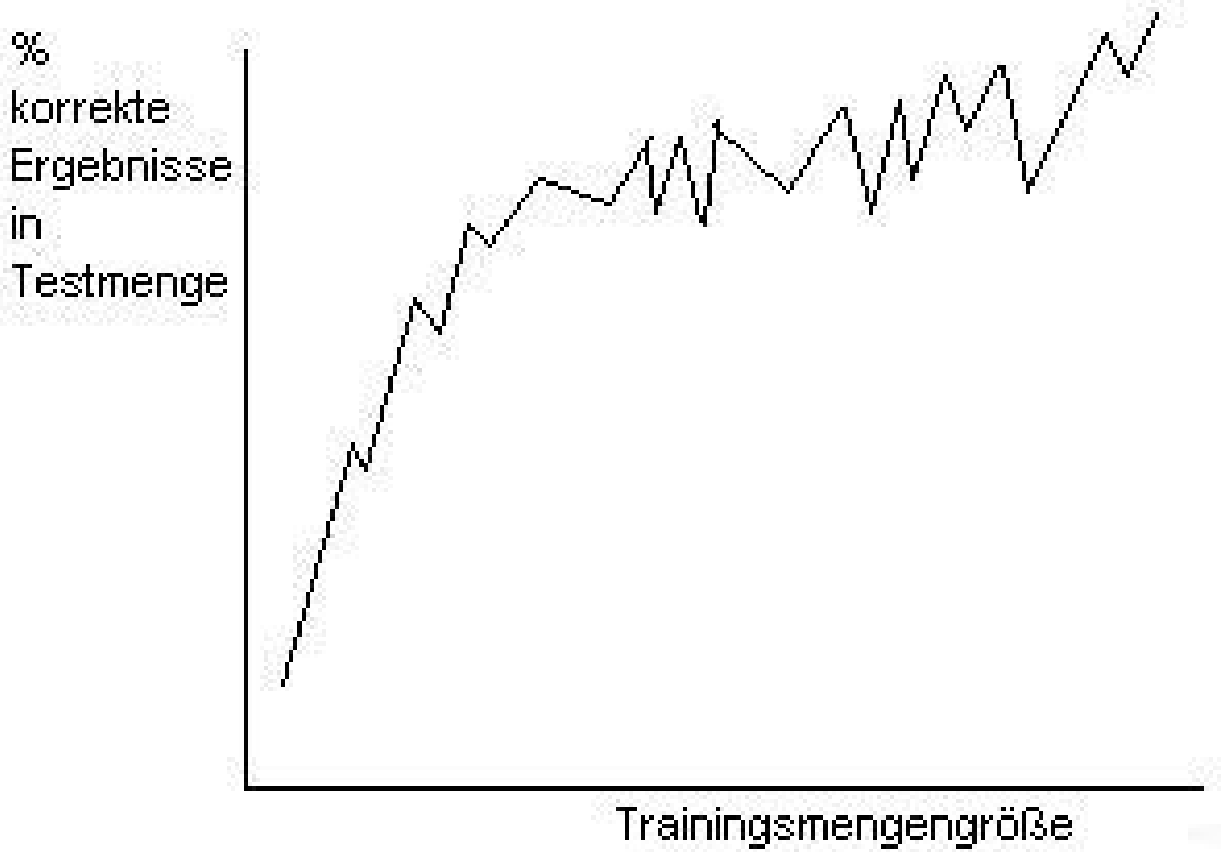
1. eine große Beispielmenge sammeln;(Korpus)
2. Die Beispielmenge in zwei separate Mengen unterteilen: **Trainingsmenge** und **Testmenge**;
3. Den Lernalgorithmus(z.B.Viterbi) auf die Trainingsmenge anwenden und eine Hypothese h erzeugen;
4. Den Prozentsatz der Beispiele in der Testmenge messen, die durch h korrekt kassifiziert wurden;
5. Die Schritte 1 bis 4 für verschiedene Größen von Trainingsmengen und verschiedene zufällig ausgewählte Trainingsmengen beliebiger Göße wiederholen.

Trainig aus computerlinguistischer Sicht

1. eine große Beispielmenge sammeln;(Korpus)
2. Die Beispielmenge in zwei separate Mengen unterteilen: **Trainingsmenge** und **Testmenge**;
3. Den Lernalgorithmus(z.B.Viterbi) auf die Trainingsmenge anwenden und eine Hypothese h erzeugen;
4. Den Prozentsatz der Beispiele in der Testmenge messen, die durch h korrekt kassifiziert wurden;
5. Die Schritte 1 bis 4 für verschiedene Größen von Trainingsmengen und verschiedene zufällig ausgewählte Trainingsmengen beliebiger Göße wiederholen.

„Übung macht den Meister“

Hier ist es auch nicht anders, mit wachsender Trainingsmenge steigt Anteil der korrekten Ergebnisse in Testmenge



Motivation

1. *Penn-Baumbank* - 1,6 Millionen Wörtern(eine CD)
   - nur für Sätze der durchschnittlichen Länge(22Wörter) Millionen von Analysen
2. *British National Corpus* (Leech et al.,2001) erweiterung der Penn-Baumbank auf 100Mio.
3. *World Wide Web* - enthält über eine Trillion Wörter(auf 10Mio. Servern gespeichert)

Common wisdom

- Warum erst 1996?
- Wo wir die Mitteln (CFGs,Bäume,...) schon lange zur Verfgung hatten.
- Es **herrschte** allgemeine **Überzeugung**, dass solche Baumbank-grammatiken(CFGs) uneffizient sind.
  Mehr dazu bei der Fehlerdiskussion.

Baumbank-Training als Beispiel für überwachtes Traning

Das überwachte Training in unserem Fall bedeutet, dass die Grammatik von einer vorhandenen Baumbank(z.B.Penn-Treebank) abgelesen und traininert wird.

Inhalt

Inhalt

Aufteilen in Trainings- und Testkorpus

1. Baumbank: Penn geparstes *Wall Street Journal*
2. Diese Baumbank wurde in zwei Korpora geteilt:

Aufteilen in Trainings- und Testkorpus

1. Baumbank: Penn geparstes *Wall Street Journal*
2. Diese Baumbank wurde in zwei Korpora geteilt:
    1. das erste Korpus (ca. 30000 Wörter) dient zum Testen

Aufteilen in Trainings- und Testkorpus

1. Baumbank: Penn geparstes *Wall Street Journal*
2. Diese Baumbank wurde in zwei Korpora geteilt:
   1. das erste Korpus (ca. 30000 Wörter) dient zum Testen
   2. das zweite (ca. 10-fache des ersten Korpus) für das Training

Aufteilen in Trainings- und Testkorpus

1. Baumbank: Penn geparstes *Wall Street Journal*
2. Diese Baumbank wurde in zwei Korpora geteilt:
   1. das erste Korpus (ca. 30000 Wörter) dient zum Testen
   2. das zweite (ca. 10-fache des ersten Korpus) für das Training
3. Von jedem Satz des Trainingskorpus werden die Regeln einer Grammatik abgelesen, dabei ignoriert man die Sätze, die länger als 40 Wörter sind. Die durchschnittliche Länge eines Satzes ist dabei 22 + Interpunktion.

Aufteilen in Trainings- und Testkorpus

1. Baumbank: Penn geparstes *Wall Street Journal*
2. Diese Baumbank wurde in zwei Korpora geteilt:
   1. das erste Korpus (ca. 30000 Wörter) dient zum Testen
   2. das zweite (ca. 10-fache des ersten Korpus) für das Training
3. Von jedem Satz des Trainingskorpus werden die Regeln einer Grammatik abgelesen, dabei ignoriert man die Sätze, die länger als 40 Wörter sind. Die durchschnittliche Länge eines Satzes ist dabei 22 + Interpunktion.

Beispiel für einen geparsten Satz des Trainingskorpus:

((S (NP The dog) (VP chewed (NP the bone))) .)

die abgeleitete Regeln (CFG, Baumbankgrammatik):

$$\begin{aligned} S &\rightarrow NP\ VP \\ NP &\rightarrow dt\ nn \\ VP &\rightarrow vb\ NP \end{aligned}$$

Die resultierende Grammatik hat 10605 Regeln, davon kommen 3943 Regeln mehrmals vor

Inhalt

Grammatik Transformationen

1. Es war nötig ein zusätzliches Startsymbol **S1(TOP)** hinzufügen, weil viele Parsen in der Baumbank folgende Form hatten:

   ((S (NP The dog) (VP chewed (NP the bone))) .)

   Beachte: die äußere nicht gelabelte Klammer in diesem Fall zwei Konstituenten S und „.“haben. Nach der Korrektur:

   (S1 (S (NP The dog) (VP chewed (NP the bone))) .)

2. Außerdem wurden die terminalen Penn-Baumbank-POS-Tags um zwei neue, nämlich **aux** und **auxg**, ergänzt.

Grammatik Transformationen

1. Es war nötig ein zusätzliches Startsymbol **S1(TOP)** hinzufügen, weil viele Parsen in der Baumbank folgende Form hatten:

   ((S (NP The dog) (VP chewed (NP the bone))) .)

   Beachte: die äußere nicht gelabelte Klammer in diesem Fall zwei Konstituenten S und „.“haben. Nach der Korrektur:

   (S1 (S (NP The dog) (VP chewed (NP the bone))) .)

2. Außerdem wurden die terminalen Penn-Baumbank-POS-Tags um zwei neue, nämlich **aux** und **auxg**, ergänzt.

Inhalt

CFG

1. Man lese die Regeln von allen Sätzen im Trainingskorpus ab(dabei ignoriert man alle Trace-Elemente[1])
   - und bekommt als Ergebnis eine Baumbankgrammatik (**CFG**)
2. Man beachte wie oft jede Regel im Trainingkorpus benutzt wurde
   und weist jeder Regel nach folgender Formeln die Wahrscheinlichkeit zu. (Von **CFG** zu **PCFG**)

[1] trace(t)-bezeichnet Spur des Elementes, das durch Topikalisierungstransformation wegbewegt worden ist:
Sandy, we want to succeed **t**(wir wollen Sandy nachfolgen)
Sandy, we want **t** to succeed(wir wollen, dass Sandy nachfolgt)

Inhalt

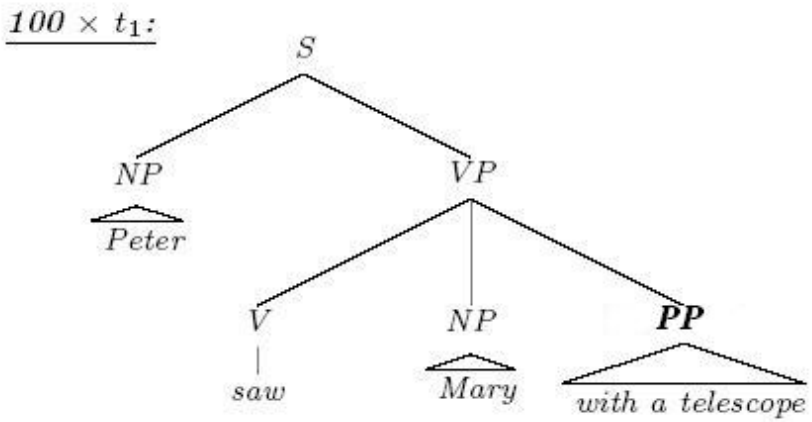
Regelwahrscheinlichkeit

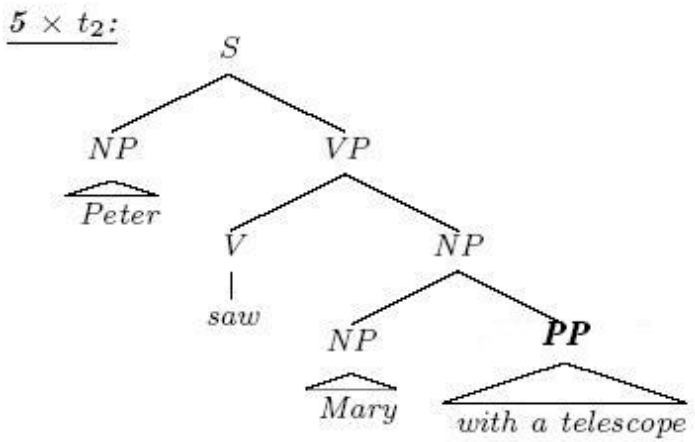
$$p(r) = \frac{|r|}{\sum_{r' \in \{r' | \lambda(r') = \lambda(r)\}} |r'|}$$

r-Regel
$|r|$-Vorkommenshäufigkeit der Regel r in der Baumbank
$\lambda(r)$-lhs(r),linke Seite von der Regel.
Interpretation: Vorkommenshäufigkeit der Regel r geteilt durch Vorkommenshäufigkeit aller Regel mit der selben linken Seite.

Baumbank mit der Wahrscheinlichkeitsverteilung

Weisen wir unserer Baumbank nun die Wahrscheinlichkeitsverteilung zu.





Inhalt

Wahrscheinlichkeitsverteilung

r	$\lvert r \rvert$	$p(r)$
$S \rightarrow NP\ VP$	$100 + 5$	$\frac{105}{105} = 1.000$
$VP \rightarrow V\ NP\ PP$	100	$\frac{100}{105} \approx 0.952$
$VP \rightarrow V\ NP$	5	$\frac{5}{105} \approx 0.048$
$NP \rightarrow Peter$	$100 + 5$	$\frac{105}{215} \approx 0.488$
$NP \rightarrow Mary$	$100 + 5$	$\frac{105}{215} \approx 0.488$
$NP \rightarrow NP\ PP$	5	$\frac{5}{215} \approx 0.024$
$PP \rightarrow with\ a\ telescope$	$100 + 5$	$\frac{105}{105} = 1.000$
$V \rightarrow saw$	$100 + 5$	$\frac{105}{105} = 1.000$

Die Summe der Wahrscheinlichkeiten über aller Regeln mit der selben linken Seite ist 1.

Diskussion

Auflösung der Ambiguität
Nach dem Prinzip: kommt häufiger vor, muss richtiger sein

$$\begin{array}{ccccccc} p(t_1) & >^2 & p(t_2) & & & & \\ p(VP \rightarrow V\ NP\ PP) & > & p(VP \rightarrow V\ NP) & * & p(NP \rightarrow NP\ PP) & & \\ 0.952 & > & 0.048 & * & 0.024 & & \end{array}$$

---

[2] Beweis:

$$\begin{array}{lcll} p(t_1) & = & 1.000 * 0.488 * 0.952 * 1.000 * 0.488 * 1.000 & \approx 0.227 \\ p(t_2) & = & 1.000 * 0.488 * 0.048 * 1.000 * 0.024 * 0.488 * 1.000 & \approx 0.00027 \end{array}$$

Probleme mit PCFGs

- **die Wahrscheinlichkeitsabschtzung basiert nur auf der Information über die Struktur eines Satzes und nicht auf der lexikalischen Information über Kollokationen**
  PCFGs sind kontextfrei.Das bedeutet,der Unterschied zwischen p(Tree saw ...) and p(Peter saw ...) ist nur von p(Peter) im vergleich zu p(Tree) abhängig und nicht von der Relation zwischen „saw “und den betreffenden Subjekten, um diese Art der Beziehung zu erhalten brauchen wir eine Art kontextabhängiges Modell, wie beispielweise eine lexikalisierte PCFG(kombinieren kontexfreie und wortbasierte Statistiken).Dazu mehr in den nächsten Vorträgen.

Probleme mit PCFGs

- **die Wahrscheinlichkeitsabschtzung basiert nur auf der Information über die Struktur eines Satzes und nicht auf der lexikalischen Information über Kollokationen**
  PCFGs sind kontextfrei.Das bedeutet,der Unterschied zwischen p(Tree saw ...) and p(Peter saw ...) ist nur von p(Peter) im vergleich zu p(Tree) abhängig und nicht von der Relation zwischen „saw “und den betreffenden Subjekten, um diese Art der Beziehung zu erhalten brauchen wir eine Art kontextabhängiges Modell, wie beispielweise eine lexikalisierte PCFG(kombinieren kontexfreie und wortbasierte Statistiken).Dazu mehr in den nächsten Vorträgen.
- **Die kleineren Syntaxbäume haben mit PCFGs eine grössere Wahrscheinlichkeit als grössere Syntaxbäume**

Probleme mit PCFGs

- **die Wahrscheinlichkeitsabschtzung basiert nur auf der Information über die Struktur eines Satzes und nicht auf der lexikalischen Information über Kollokationen**
  PCFGs sind kontextfrei.Das bedeutet,der Unterschied zwischen p(Tree saw ...) and p(Peter saw ...) ist nur von p(Peter) im vergleich zu p(Tree) abhängig und nicht von der Relation zwischen „saw “und den betreffenden Subjekten, um diese Art der Beziehung zu erhalten brauchen wir eine Art kontextabhängiges Modell, wie beispielweise eine lexikalisierte PCFG(kombinieren kontexfreie und wortbasierte Statistiken).Dazu mehr in den nächsten Vorträgen.
- **Die kleineren Syntaxbäume haben mit PCFGs eine grössere Wahrscheinlichkeit als grössere Syntaxbäume**

Inhalt

Resultat

1. Mit der abgeleiteten Grammatik wird das Testkorpus geparst.
2. **Viterbi-Parse:** Jeder Satz des Testkorpus hat dank dem Viterbi-Algorithmus für PCFGs einen Parsebaum mit der größten Wahrscheinlichkeit.

Resultat

1. Mit der abgeleiteten Grammatik wird das Testkorpus geparst.
2. **Viterbi-Parse:** Jeder Satz des Testkorpus hat dank dem Viterbi-Algorithmus für PCFGs einen Parsebaum mit der größten Wahrscheinlichkeit.

Anschließend wird der Viterbi-Parse mit dem Baumbank-Parse(Testkorpus) vergliechen.

Resultat

1. Mit der abgeleiteten Grammatik wird das Testkorpus geparst.
2. **Viterbi-Parse:** Jeder Satz des Testkorpus hat dank dem Viterbi-Algorithmus für PCFGs einen Parsebaum mit der größten Wahrscheinlichkeit.

Anschließend wird der Viterbi-Parse mit dem Baumbank-Parse(Testkorpus) vergliechen.

Inhalt

Inhalt

Parameter[a] zur Messung der Güte des Parsens

[a] Diese Größen stammen aus Information-Retrieval

1. „Precision“:

$$\frac{|\{NTSymbole\ im\ ViterbiParse\} \cap \{NTSymbole\ im\ BaumbankParse\}|}{|\{NTSymbole\ im\ ViterbiParse\}|}$$

Wie viele nichtterminale Symbole vom Viterbi-Parse kommen auch im Baumbank-Parse vor

2. „Recall“:

$$\frac{|\{NTSymbole\ im\ ViterbiParse\} \cap \{NTSymbole\ im\ BaumbankParse\}|}{|\{NTSymbole\ im\ BaumbankParse\}|}$$

Wie viele nichtterminale Symbole vom Baumbank-Parse kommen auch im Viterbi-Parse vor

Parameter[a] zur Messung der Güte des Parsens

[a] Diese Größen stammen aus Information-Retrieval

1. „Precision“:

$$\frac{|\{NTSymbole\ im\ ViterbiParse\} \bigcap \{NTSymbole\ im\ BaumbankParse\}|}{|\{NTSymbole\ im\ ViterbiParse\}|}$$

Wie viele nichtterminale Symbole vom Viterbi-Parse kommen auch im Baumbank-Parse vor

2. „Recall“:

$$\frac{|\{NTSymbole\ im\ ViterbiParse\} \bigcap \{NTSymbole\ im\ BaumbankParse\}|}{|\{NTSymbole\ im\ BaumbankParse\}|}$$

Wie viele nichtterminale Symbole vom Baumbank-Parse kommen auch im Viterbi-Parse vor

Inhalt

Ergebnisse

Satzlänge	Satzlänge im Durchschnitt	Precision	Recall
$2-12$	8.7	88.6	91.7
$2-16$	11.4	85.0	87.7
$2-20$	13.8	83.5	86.2
$2-25$	16.3	82.0	84.0
$2-30$	18.7	80.6	82.5
$2-40$	21.9	78.8	80.4

Achtung: Charniak hat in diesem Experiment Wörter durch POS-Tags ersetzt(d.h.terminale Symbole=POS-Tags).Evaluierungswerte sind deswegen etwa zu optimistisch(siehe nächste Vorträge).

Ergebnisse

Satzlänge	Satzlänge im Durchschnitt	Precision	Recall
$2-12$	8.7	88.6	91.7
$2-16$	11.4	85.0	87.7
$2-20$	13.8	83.5	86.2
$2-25$	16.3	82.0	84.0
$2-30$	18.7	80.6	82.5
$2-40$	21.9	78.8	80.4

Achtung: Charniak hat in diesem Experiment Wörter durch POS-Tags ersetzt(d.h.terminale Symbole=POS-Tags).Evaluierungswerte sind deswegen etwa zu optimistisch(siehe nächste Vorträge).

Inhalt

Inhalt

Ursachen für falches Parsen

1. die nötige Regeln sind nicht in der Grammatik
2. die Regeln sind da, aber ihre Wahrscheinlichkeiten sind falsch

Ursachen für falches Parsen

1. die nötige Regeln sind nicht in der Grammatik
2. die Regeln sind da, aber ihre Wahrscheinlichkeiten sind falsch
3. Pruning könnte den korrekten Parse eliminiert haben

Ursachen für falches Parsen

1. die nötige Regeln sind nicht in der Grammatik
2. die Regeln sind da, aber ihre Wahrscheinlichkeiten sind falsch
3. Pruning könnte den korrekten Parse eliminiert haben
4. der richtige Parse wurde gefunden, aber Baumbank „gold standard“ war falsch (oder der richtige Parse ist nicht sauber)

Ursachen für falches Parsen

1. die nötige Regeln sind nicht in der Grammatik
2. die Regeln sind da, aber ihre Wahrscheinlichkeiten sind falsch
3. Pruning könnte den korrekten Parse eliminiert haben
4. der richtige Parse wurde gefunden, aber Baumbank „gold standard“ war falsch (oder der richtige Parse ist nicht sauber)

Inhalt

Beispiel zum Fehler 1

Charniak vermutet, dass Baumbank-Grammatiken untergeschätzt wurden, weil man Angst vor unvollständiger Grammatik hatte(Fehler 1).

Beispiel zum Fehler 1:

NP „the $ 200 hat“$\longrightarrow$ $NP \longrightarrow dt\ \$\ cd\ nn$

Beispiel zum Fehler 1

Charniak vermutet, dass Baumbank-Grammatiken untergeschätzt wurden, weil man Angst vor unvollständiger Grammatik hatte(Fehler 1).

Beispiel zum Fehler 1:

NP „the $ 200 hat“$\longrightarrow$ $NP \longrightarrow dt\ \$\ cd\ nn$

Unsere Baumbank-Grammatik hat aber keine entsprechende **NP** Regel und konnte keinen korrekten Parse zum Satz mit diesem **NP** zuweisen.

Beispiel zum Fehler 1

Charniak vermutet, dass Baumbank-Grammatiken untergeschätzt wurden, weil man Angst vor unvollständiger Grammatik hatte(Fehler 1).

Beispiel zum Fehler 1:

NP „the $ 200 hat“$\rightarrow$ $NP \rightarrow dt\ \$\ cd\ nn$

Unsere Baumbank-Grammatik hat aber keine entsprechende **NP** Regel und konnte keinen korrekten Parse zum Satz mit diesem **NP** zuweisen.

Inhalt

Pruning

Mit häutigen Rechnerkapazitäten ist es keine Lösung alle Parsen zu betrachten, man behilft sich dagegen mit:

Pruning

ist der englische Ausdruck für die Beschneidung von abgestorbenen, überreifen, oder aus anderen Gründen unerwünschten Teilen von Bäumen und Sträuchern
WIKIPEDIA

Pruning

Mit häutigen Rechnerkapazitäten ist es keine Lösung alle Parsen zu betrachten, man behilft sich dagegen mit:

Pruning

ist der englische Ausdruck für die Beschneidung von abgestorbenen, überreifen, oder aus anderen Gründen unerwünschten Teilen von Bäumen und Sträuchern
WIKIPEDIA

Pruning wird verwendet, um die unwahrscheinlichen Hypothesen(Parsen) zu eliminieren. Charniak hat in seinem Experiment die Strategie der Bestensuche(Best-First) verwendet.

Pruning

Mit häutigen Rechnerkapazitäten ist es keine Lösung alle Parsen zu betrachten, man behilft sich dagegen mit:

Pruning

ist der englische Ausdruck für die Beschneidung von abgestorbenen, überreifen, oder aus anderen Gründen unerwünschten Teilen von Bäumen und Sträuchern
WIKIPEDIA

Pruning wird verwendet, um die unwahrscheinlichen Hypothesen(Parsen) zu eliminieren. Charniak hat in seinem Experiment die Strategie der Bestensuche(Best-First) verwendet.

Best-First

Die Bestensuche ist ein Beispiel für den allgemeinen TREE-SEARCH- oder GRAPH-SEARCH-Algorithmus, wobei ein Knoten abhängig von eine Evaluierungsfunktion f(n) für die Expandierung ausgewählt wird.

Es gibt eine ganze Familie von BEST-FIRST-SEARCH-Algorithmen mit unterschiedlichen Evaluierungsfunktionen. Eine Schlüßelkomponente dieser Algorithmen ist eine **Heuristikfunktion**[3], hier als h(n) bezeichnet:

h(n)=geschätzte Kosten für den billigsten Pfad vom Knoten n zu einem Zielknoten.

[3]Eine Heuristikfunktion h(n) nimmt einen Knoten als Eingabe entgegen, aber sie ist nur vom *Zustand* an diesem Knoten abhängig.
Heuristikfunktionen sind die gebräuchlichste Form,dem Suchalgorithmus zusätzliches Wissen über das Problem mitzuteilen
Wenn n ein Zielknoten ist, dann gilt h(n)=0.

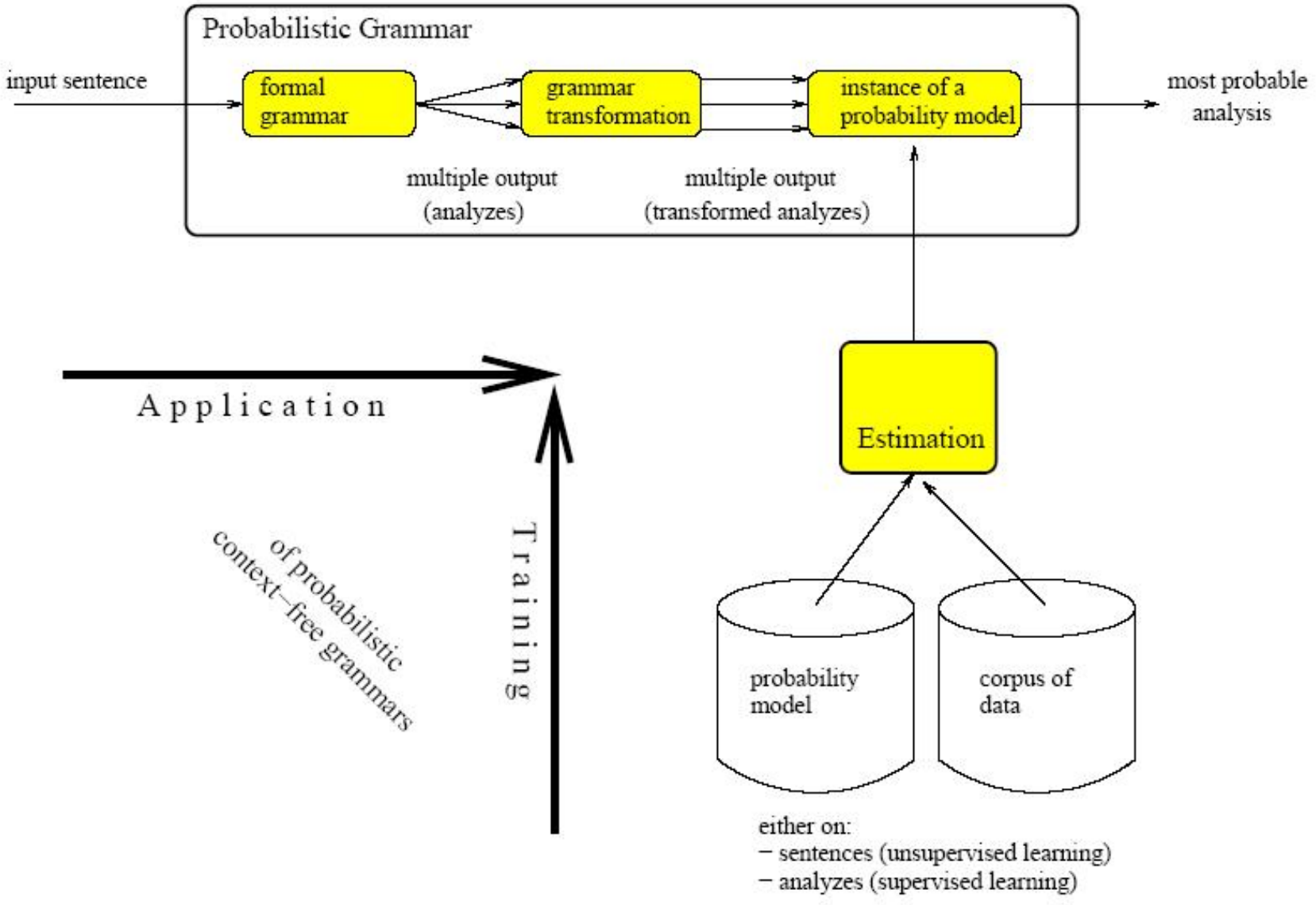
Best-First

Die Bestensuche ist ein Beispiel für den allgemeinen TREE-SEARCH- oder GRAPH-SEARCH-Algorithmus, wobei ein Knoten abhängig von eine Evaluierungsfunktion f(n) für die Expandierung ausgewählt wird.
Es gibt eine ganze Familie von BEST-FIRST-SEARCH-Algorithmen mit unterschiedlichen Evaluierungsfunktionen. Eine Schlüßelkomponente dieser Algorithmen ist eine **Heuristikfunktion**[3], hier als h(n) bezeichnet:

h(n)=geschätzte Kosten für den billigsten Pfad vom Knoten n zu einem Zielknoten.

[3]Eine Heuristikfunktion h(n) nimmt einen Knoten als Eingabe entgegen, aber sie ist nur vom *Zustand* an diesem Knoten abhängig.
Heuristikfunktionen sind die gebräuchlichste Form,dem Suchalgorithmus zusätzliches Wissen über das Problem mitzuteilen
Wenn n ein Zielknoten ist, dann gilt h(n)=0.

Baumbank-Training graphisch



Literatur


Eugene Charniak: Tree-bank Grammars, Technical Report CS-96-02, Department of Computer Science, Brown University, 1996


Detlef Prescher: A Tutorial on the Expectation-Maximization Algorithm Including Maximum-Likelihood Estimation and EM Training of Probabilistic Context-Free Grammars, ESSLLI 2003


Stuart Russell,Peter Norvig: Künstliche Intelligenz, Ein moderner Ansatz, 2.Aufl. *Pearson Studium Verlag*, 2004


Ruslan Mitkov: The Oxford Handbook of Computational Linguistics, *Oxford University Press*, 2004

Danke

Vielen Dank
für eure Aufmerksamkeit!