

Maschinelles Übersetzen natürlicher Sprachen

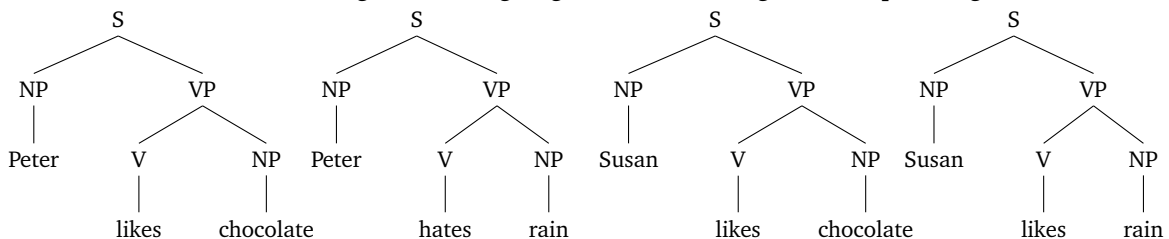
Diese Übung findet wieder im **Rechnerkabinett E042** statt!

Wir werden uns mit dem XTT-Toolkit *Tiburon* beschäftigen, runterzuladen unter <http://www.isi.edu/licensed-sw/tiburon/>.

Aufgabe 1 (Reguläre Baumgrammatiken)

Machen Sie sich anhand der Beispiele in den Ordnern `samples/` und `tutorial/` mit der Tiburon-Syntax für (P)RTGs vertraut. Wie unterscheiden sich diese von den in der Vorlesung definierten?

- Erstellen Sie eine RTG mit möglichst wenig Regeln, deren erzeugte Baumsprache genau die Bäume



umfasst. Testen Sie diese RTG mit Tiburon mittels der Optionen `-k` oder `-g`.

Der Befehl `./tiburon -h` gibt eine Beschreibung der verschiedenen verwendbaren Optionen aus.

- Wir betrachten die Datei `rtg1`:

```
q
q -> s(p q) # 0.6
q -> a      # 0.4
p -> g(p)   # 0.7
p -> b      # 0.3
```

Welche PRTG (\mathcal{G}, p) ist durch diesen Tiburon-Code gegeben? Bestimmen Sie die drei wahrscheinlichsten Bäume unter \mathcal{G} und überprüfen Sie Ihre Antwort mittels der k -best-Funktionalität von Tiburon:

```
./tiburon -k <anzahl> rtg1
```

- Fügen Sie `rtg1` die Regel `q -> s(p a) # 0.8` hinzu und führen Sie wieder k -best aus. Was fällt Ihnen auf? Geben Sie eine Erklärung für das auftretende Phänomen.

Aufgabe 2 (Extended Tree Transducers)

Machen Sie sich nun anhand der mitgelieferten Beispiele mit der Tiburon-Syntax für Extended Tree Transducers vertraut.

- Implementieren Sie für $\Sigma = \{\sigma, \gamma, \alpha, \beta\}$ den XTT $\mathcal{M} = (\{q, p\}, \Sigma, \Sigma, q, R)$ mit Regelmenge R gegeben durch

$$\begin{aligned} q(\sigma(x_1, x_2)) &\rightarrow \sigma(q(x_2), p(x_1)) & q(\alpha) &\rightarrow \alpha \\ p(\gamma(x_1)) &\rightarrow \sigma(p(x_1), \beta) & p(\alpha) &\rightarrow \beta \end{aligned}$$

in Tiburon. Nutzen Sie die Syntax

```
echo "<Eingabebaum>" | ./tiburon [optionen] - <xtt>
```

um verschiedene Eingabebäume mittels \mathcal{M} zu übersetzen.

- **Zusatzaufgabe:** Implementieren Sie den Tree Transducer aus Aufgabe 2 der 8. Übung und übersetzen Sie ausgewählte Beispielsätze. Erweitern Sie den Transducer um weitere sinnvolle Regeln.

Aufgabe 3 (Training)

Im Ordner `samples/` befinden sich der untrainierte Transducer `y1.ts` sowie der bilinguale Korpus `y1.train`. Nutzen Sie letzteren, um Tiburon Regelgewichte für `y1.ts` trainieren zu lassen. Der entsprechende Aufruf ist von der Form

```
./tiburon -t <EM-Iterationen> -o <ausgabedatei> <transducer> <korpus>.
```

Nutzen Sie den erstellten PXTT, um die fünf besten Übersetzungen des Satzes “`sus clientes estan enfadados`” zu bestimmen.

Experimentieren Sie mit Tiburon, indem Sie den Korpus bzw. den zu trainierenden Transducer erweitern oder erstellen Sie eigene Trainingsdaten.

Aufgabe 4 (Forward/Backward Application)

Mit Tiburon kann man auch ganze reguläre Baumsprachen durch Tree Transducer übersetzen lassen. Setzen Sie die aus Aufgabe 3 der letzten Übung bekannten Grammatiken \mathcal{G}_1 , \mathcal{G}_2 und \mathcal{G}'_1 sowie den Transducer \mathcal{M} in Tiburon um und nutzen Sie die Aufruf

```
./tiburon <G1> <M>
```

bzw.

```
./tiburon <M> <G2>
```

um Tiburon die entsprechenden Forward/Backward Applications ausgeben zu lassen und vollziehen Sie die Konstruktion nach. Was bewirken die Befehle

```
./tiburon -l <G1> <M> <G2>
```

bzw.

```
./tiburon -r <G1> <M> <G2> ?
```