# Übungsblatt 12

## Präsenzaufgaben

### Aufgabe 1 Herunterladen von Ressourcen

Laden Sie sich zunächst die Ressource `corpora/treebank` über den NLTK Download-Manager herunter.

```
1 import nltk
2 nltk.download()
```

### Aufgabe 2 Grammatikinduktion

In dieser Aufgabe soll vollautomatisch aus Daten (Syntaxbäumen) eine probabilistische, kontextfreie Grammatik erzeugt werden.

Veranschaulichen Sie sich das Vorgehen zunächst an einem Beispiel. Gegeben sei folgende kontextfreie Grammatik:

```
1  S → NP VP
2  VP → V NP PP
3  VP → V NP
4  NP → DET N
5  NP → NP PP
6  PP → P NP
7
8  DET → "the" | "a"
9  N → "boy" | "woman" | "telescope"
10 V → "saw"
11 P → "with"
```

Sie modelliert sehr einfache Sätze der Form `SBJ` *saw* `OBJ` mit optionaler Präpositionalphrase am Ende. Diese Präpositionalphrase kann entweder der näheren Bestimmung des Objekts oder der näheren Bestimmung der in der Verbalphrase ausgedrückten Handlung dienen.

**Für welche Regeln müssen wir die Wahrscheinlichkeiten berechnen, wenn wir mit statistischen Methoden untersuchen wollen, ob PPs häufiger Teil der VP oder Teil der NP sind?**

(a) Approximieren Sie mittels vergleichbarer Konstruktionen in der Penn Treebank die Wahrscheinlichkeiten für die ersten beiden dieser Regeln.

(b) Im Folgenden wollen wir vollautomatisch eine dem Penn Treebank Schema entsprechende PCFG erzeugen. Kopieren Sie den mit Lücken versehenen Code aus dem Notebook für diese Woche, ergänzen Sie den Code und versuchen Sie mithilfe Ihrer automatisch erstellten Grammatik die folgenden Sätze zu parsen:

(1) the men saw a car .

(2) the woman gave the man a book .

(3) she gave a book to the man .

(4) yesterday , all my trouble seemed so far away .

### Aufgabe 3 Neuronale Netze

In dieser Aufgabe sollen Sie die grundlegenden Eigenschaften von Feedforward-Netzen vertiefen. Falls noch nicht geschehen, installieren Sie dafür die pytorch-Bibliothek. Befehle dafür finden Sie unter `https://pytorch.org/get-started/locally/`.

Wenn Sie den Code der heutigen Übung auf Ihrem eigenen Rechner ausführen wollen, sollten Sie anschließend auch ignite installieren, das einige zusätzliche Hilfsfunktionen bereitstellt:

Mit pip: `pip install pytorch-ignite`

ODER mit conda: `conda install ignite -c pytorch`

Folgen Sie nun den Anweisungen auf dem Jupyter Notebook dieser Woche bzw. kopieren Sie relevanten Code in eine Entwicklungsumgebung Ihrer Wahl.

## Hausaufgaben

### Aufgabe 4 PCFGs und Viterbi

Beantworten Sie die folgenden grundlegenden Fragen über die heute verwendeten Technologien.

(a) Welche der folgenden Bedingungen wird an eine PCFG gestellt?

- ◯ Die Summe aller Regelwahrscheinlichkeiten für jede LHS ist jeweils 1.
- ◯ Die Summe aller Regelwahrscheinlichkeiten für jede RHS ist jeweils 1.
- ◯ Die Summe aller Regelwahrscheinlichkeiten innerhalb einer Grammatik ist 1.

(b) Was ist die Aufgabe des Viterbi-Algorithmus?

- ◯ Bestimmung der Köpfe und Dependenzrelationen
- ◯ Bestimmung des wahrscheinlichsten Syntaxbaums
- ◯ Finden aller Konstituenten

(c) Warum muss zwischen zwei Schichten eines Feedforward-Netzwerks eine nicht-lineare Aktivierungsfunktion eingefügt werden?

- ◯ Weil Vektoren normalisiert werden müssen, um bestimmte Funktionen zu lernen.
- ◯ Weil die zwei Schichten sonst nicht mehr lernen können als nur eine Schicht.
- ◯ Weil eine valide Wahrscheinlichkeitsverteilung nur mit Normalisierung gelernt werden kann.
- ◯ Weil nicht-lineare Funktionen die Effizienz des Netzwerks verbessern.

## *Aufgabe 5 NLTK-Kapitel zu PCFGs

In folgenden NLTK-Kapiteln wird das Parsing mit Probabilistischen kontextfreien Grammatiken behandelt:

- **Teilkapitel 8.6** ('Grammar Development'):
  `http://www.nltk.org/book/ch08.html`
- **Teilkapitel 2.12 und 2.13** ('Grammar Induction' und 'Normal Forms') des Zusatzkapitels zu Kapitel 8:
  `http://www.nltk.org/book/ch08-extras.html`
- (Teilkapitel 2.9-2.11 des Zusatzkapitels zu Kapitel 8 behandelt Probabilistische 'Chart Parsing'-Algorithmen:
  `http://www.nltk.org/book/ch08-extras.html`)

Beantworten Sie folgende Fragen zu Teilkapitel 8.6.2 ('Pernicious Ambiguity'):

(a) Welche zwei Faktoren führen bei der syntaktischen Analyse natürlicher Sprache mittels formaler Grammatiken zu mehr Ambiguität (Anzahl an Ableitungen)?

(b) Welche zwei Arten von Ambiguität unterscheidet man hier?