Übungsblatt 12

16.01.19

Präsenzaufgaben

Aufgabe 1 Grammatikinduktion

In dieser Aufgabe soll vollautomatisch aus Daten (Syntaxbäumen) eine probabilistische, kontextfreie Grammatik erzeugt werden. 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 2 Informationsextraktion per Syntaxanalyse

Gegenstand dieser Aufgabe ist eine anwendungsnahe Möglichkeit, Ergebnisse einer Syntaxanalyse weiterzuverarbeiten. Aus den syntaktischen Abhängigkeiten eines Textes soll (unter Zuhilfenahme einiger Normalisierungsschritte) eine semantische Repräsentation der im Text enthaltenen Informationen gewonnen werden.

Für die syntaktische Analyse soll der `DependencyParser` der Stanford CoreNLP Suite verwendet werden. Die semantische Repräsentation eines Satzes sei ein zweistelliges, logisches Prädikat, dessen Argumente durch Subjekt und Objekt gefüllt sind. (Bei Fehlen eines der beiden Elemente soll `None` geschrieben werden.)

Die nötigen Normalisierungsschritte umfassen insbesondere Passivkonstruktionen und Relativsätze. Folgendes Beispiel illustriert das gewünschte Ergebnis:

Eingabe:

> I shot an elephant in my pajamas. The elephant was seen by a giraffe in the desert. The bird I need is a raven. The man who saw the raven laughed out loud.

Ausgabe:

- shot(I, elephant)

- seen(giraffe, elephant)
- need(I, bird)
- raven(bird, `None`)
- saw(man, raven)
- laughed(man, `None`)

Anmerkung: Sie können sich insbesondere für die Benutzung des Stanford Parsers an dem im Jupyter Notebook gegebenen Code orientieren.

Hausaufgaben

Aufgabe 3 Parent Annotation

Parent Annotation kann die Performanz einer CFG wesentlich verbessern. Schreiben Sie eine Funktion, die einen gegebenen Syntaxbaum dieser Optimierung unterzieht. Auf diese Art und Weise transformierte Bäume können dann wiederum zur Grammatikinduktion verwendet werden.

Ihre Funktion sollte neben dem Syntaxbaum selbst noch folgende Parameter erwarten:

- `parentHistory` soll die Anzahl der Vorgänger sein, die zusätzlich zum direkten Elternknoten berücksichtigt werden. (Kann bei der Lösung der Aufgabe auch ignoriert werden.)
- `parentChar` soll ein Trennzeichen sein, das bei den neuen Knotenlabels zwischen dem ursprünglichen Knotenlabel und der Liste von Vorgängern eingefügt wird.

Aufgabe 4 Mehr Semantik für IE

Zusätzlich zu den in Aufgabe 2 behandelten Konstruktionen sollen jetzt auch negierte und komplexe Sätze mit Konjunktionen sinnvoll verarbeitet werden.

Eingabe:

> I see an elephant. You didn't see the elephant. Peter saw the elephant and drank wine.

Ausgabe:

- see(I, elephant)
- not_see(You, elephant)
- saw(Peter, elephant)
- drank(Peter, wine)

*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?