Vorlesung "Computerlinguistische Techniken"

1. Übung (24.10.2023)

Wintersemester 2023/2024 – Prof. Dr. David Schlangen

1 Mehrdeutigkeiten

Der folgende Beispielsatz (von Hans Uszkoreit) ist massiv mehrdeutig.

> Früher stellten die Frauen der Inseln am Wochenende Kopftücher mit Blumenmotiven her, die ihre Männer an den folgenden Montagen auf dem Markt im Zentrum der Hauptinsel verkauften.

Finden Sie möglichst viele Mehrdeutigkeiten, zeigen Sie, wie sie sich auf die Bedeutung des Satzes auswirken, und berechnen Sie, wie viele Lesarten der Satz insgesamt hat. Es ist dabei nicht wichtig, dass die einzelnen Lesarten in der wirklichen Welt möglich sind; diese Unterscheidung kann eine Grammatik auch nicht ohne weiteres treffen. Denken Sie besonders an syntaktische, lexikalische und referentielle Ambiguitäten.

Ich komme auf etwas über 250.000 Lesarten mit verschiedenen Strukturen und/oder Bedeutungen.

2 Kontextfreies Parsing

Die folgenden Regeln definieren eine kontextfreie Grammatik G:

S	→	NP VP	VP	→	V NP	VP	→	VP PP
NP	→	PN	NP	→	Det N	NP	→	NP PP
PP	→	P NP	PN	→	Hans	V	→	beobachtet
Det	→	den	Det	→	dem	N	→	Mann
N	→	Fernrohr	P	→	mit			

Bestimmen Sie alle Parsebäume, mit denen G den Satz "Hans beobachtet den Mann mit dem Fernrohr" erzeugen kann. Sie müssen nicht einen der Parsing-Algorithmen verwenden, die wir in der Vorlesung besprochen haben. Beschreiben Sie aber auf jeden Fall, wie Sie auf diese Parsebäume gekommen sind und warum Sie überzeugt sind, dass es keine weiteren gibt.

(Sie können die Bäume per Hand in Ihre maschinengeschriebene Erklärung einfügen und das Ergebnis dann scannen. Sie könnten dies aber auch als Gelegenheit nehmen, sich einmal LaTeX anzuschauen und eines der vielen Pakete, mit denen man in LaTeX Syntaxbäume erzeugen kann, z.B. `tikz-qtree`.[1])

[1] `https://home.uni-leipzig.de/murphy/handouts/tikz-qtree-%20tutorial.pdf`

3 Der Recursive-Descent-Algorithmus

Implementieren Sie einen Recursive-Descent-Erkenner in Python. Parsen Sie die Sätze "Hans isst ein Kaesebrot" und "Kaesebrot isst Hans" mit der Käsebrot-Grammatik aus der Vorlesung (Folie 6). Ihr Erkenner soll den ersten Satz als grammatisch und den zweiten als ungrammatisch erkennen und dazu abschließend True oder False ausgeben.

Ihre Implementierung darf auf die konkrete Grammatik angepasst sein. Sie müssen also keine Grammatiken einlesen oder im Speicher darstellen, sondern Ihr Programm soll aus einer Reihe von Funktionen mit den Namen parse_S, parse_NP, parse_VP usw. bestehen, die sich gegenseitig aufrufen. Ein Code-Gerüst ist auf Moodle bereitgestellt unter dem Namen `rd.py`. Bitte verändern Sie die Funktionsnamen nicht und fügen Sie keine weiteren Funktionen hinzu. Außerdem stehen Ihnen in der Datei `student_test_suite.py` einige Tests zur Verfügung, mit denen Sie Ihren Code selbst überprüfen können. Die Tests können über die Kommandozeile mit dem Befehl `pytest student_test_suite.py` aufgerufen werden. Verändern Sie hierzu nicht den Namen von `rd.py`, da die Tests sonst nicht ausgeführt werden können.

Zusätzlich zur abschließenden True/False-Ausgabe soll Ihr Programm ausgeben, welche möglichen Zerlegungen der RD-Erkenner ausprobiert und welche von ihnen erfolgreich sind, z.B. so:

```
>>> ww = ["Hans", "isst"]
>>> S(ww, 0, len(ww))
Call: S von 0 bis 2?
Call: NP von 0 bis 1?
Wort 'Hans' von 0 bis 1? -> ja
NP von 0 bis 1? -> ja
Call: VP von 1 bis 2?
VP von 1 bis 2? -> nein
S von 0 bis 2? -> nein
False
```

Verfolgen Sie damit nach, wie der RD-Erkenner die Beispielsätze akzeptiert oder ablehnt. Bitte fügen Sie den Output des Parsers für die beiden Sätze in Ihre PDF ein.

Bitte reichen Sie eine PDF-Datei, benannt nach dem Schema `ue01-NACHNAME-MATRIKELNR.pdf`, und eine Python-Datei (Dateiname rd.py) mit dem Code über Moodle ein.

Abgabe bis 7.11.2023, 13:00 Uhr, via Moodle