Parsing mit neuronalen Netzen: Anwendung

In dieser Aufgabe wird die Parser-Implementierung durch ein Programm für die Anwendung des Parsers abgeschlossen. Das Programm soll das trainierte neuronale Netzwerk einlesen und damit Sätze aus einer Datei parsen. Jede Zeile der Datei enthält einen tokenisierten Satz (keinen Parsebaum!)

Korrigieren Sie zunächst die Fehler in Ihrem Code aus Aufgabe 10.

Erstellen Sie eine Datei **parser-analyze.py** für das Parsen neuer Sätze, in der Sie die Module *Parser.py* und *Data.py* importieren.

Implementieren Sie eine Funktion **parse**, welche für jeden Span das wahrscheinlichste Label und seinen Score berechnet. Dann wird mit dem Viterbi-Algorithmus der beste Parsebaum extrahiert. Der beste Parsebaum maximiert die Summe der Label-Scores.

Schreiben Sie ferner eine Funktion **build_parse**, welche anhand der Ausgabe der Methode *parse* den Parsebaum als String generiert und zurückgibt. Sie können mit dem Befehl *data.ID2label[labelID]* den Index einer syntaktischen Kategorie auf ihren Namen abbilden.

Schreiben Sie nun das Hauptprogramm. Dieses liest erst die Abbildungstabellen ein (mit `data = Data(filename)`), dann das trainierte Parser-Netzwerk, und schließlich die Sätze (mit `data.sentences(filename)`). Dann werden die Sätze geparst und die Parsebäume auf Stdout ausgegeben. Erstellen Sie zum Testen eine Textdatei mit einem tokenisierten Satz pro Zeile und parsen Sie die Sätze mit ihrem Parser.

Aufruf: python parser-analyse.py parfile sentences

Schicken Sie mir bitte den kompletten ausführbaren Code mit den trainierten Parameterdateien, so dass ich das Programm direkt testen kann.

Vorüberlegungen

- Wie implementieren Sie die Funktion *parse*?
- Wie implementieren Sie die Funktion *build_parse*?