

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

Implementieren Sie dann eine Funktion **parse**, welche zunächst 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 dann weiteren Code, um die Abbildungstabellen einzulesen (mit `data = Data(filename)`), das trainierte Parser-Netzwerk einzulesen, die Sätze einzulesen (mit `data.sentences(filename)`), zu parsen und die Parsebäume auf Stdout auszugeben. Erstellen Sie eine Beispieldatei mit einem tokenisierten Satz pro Zeile und testen Sie damit ihren Parser.

Schicken Sie mir bitte Ihren Code mit trainierten Parameterdateien zum Testen.

### Vorüberlegungen

- Wie implementieren Sie die Funktion *parse*?
- Wie implementieren Sie die Funktion *build\_parse*?