Schriftliche Wiederholungsprüfung zur Übung
Statistische Methoden in der maschinellen Sprachverarbeitung
WS 2020/21
Dozent: Helmut Schmid

Sie haben für die Bearbeitung 60 Minuten Zeit.

Wenn Sie einen Fehler in den Aufgaben entdecken sollten, dann melden Sie sich bitte per Zoom (nicht Zoom-Chat). Notfalls können Sie mich auch unter der Nummer 07121 44240 anrufen.

Thema der Prüfung ist die Implementierung eines (vereinfachten) **Earley-Parsers**. Der Earley-Parser ist ein Chart-Parser, der sich vom Left-Corner-Parser vor allem in der Predict-Funktion unterscheidet. Die Complete-Operation ist gleich.

Als **Lösungsvorlage** können Sie die Musterlösung zur Übungsaufgabe "Statistischer Parser" auf der Kursseite nehmen. Der Parser muss nur die Wahrscheinlichkeit der besten Analyse berechnen, aber keinen Parsebaum erstellen.

Die **Predict**-Funktion des Earley-Parsers wird aufgerufen, wenn eine Punktregel $A \rightarrow \alpha \cdot B\beta$ in die Chart eingetragen wird, deren Punkt noch nicht das rechte Ende der Regel erreicht hat. Die Funktion erhält das Nichtterminal B nach dem Punkt und die Endposition *pos* des Spans der Punktregel als Argumente und schlägt alle Grammatikregeln mit B auf der linken Seite in der Grammatik nach. Jede Regel wird mit Punktposition 0 sowie Start- und Endposition *pos* in die Chart eingetragen. (Von der rechten Seite der eingetragenen Regel wurde also noch nichts gefunden. Der Span der Regel ist noch leer.)

Der Parser speichert Punktregeln als Tupel tup=(lhs, rhs, dotpos, startpos, endpos) in der Chart. Die Chart ist eine Liste von Dictionaries. Der Zugriff darauf erfolgt mit: **self.vitprob[endpos][tup] = prob**

Bedeutung der Variablennamen:

lhs = linke Seite der Regel
rhs = Liste mit den Elementen auf der rechten Seite der Regel
dotpos = Position des Punktes in der rechten Seite der Regel
startpos = Startposition der Punktregel (Index des ersten abgedeckten Wortes)
endpos = Endposition der Punktregel (Index des letzten abgedeckten Wortes + 1)
prob = Wahrscheinlichkeit der Punktregel

Schreiben Sie eine Python-Klasse **Parser** mit folgenden Methoden:

Aufgabe 1) Die Konstruktor-Methode **__init__** bekommt zwei Dateinamen *grammarfile* und *lexfile* als Argumente und ruft die Methoden *read_grammar* und *read_lexicon* auf, um die Dateien jeweils einzulesen. (1 Punkt)

Aufgabe 2) Die Methode **read_grammar** bekommt eine Grammatikdatei als Argument, liest die Regeln ein und speichert sie so in einer Datenstruktur *self.ruleprobs*, dass man für ein gegebenes Nichtterminal A direkt alle Regeln $A \rightarrow \alpha$ mit diesem Nichtterminal auf der linken Seite und die Regel-Wahrscheinlichkeit erhält. Das Symbol auf der linken Seite der ersten Grammatikregel wird in *self.start_symbol* gespeichert.

Jede Zeile der Grammatikdatei enthält (i) eine Wahrscheinlichkeit, (ii) die linke Seite einer Grammatikregel und (iii) die Symbole der rechten Seite der Regel, bspw.

```
1.0 S NP VP
0.5 VP VP PP
0.5 VP v NP
0.4 NP d N1
...
```

(4 Punkte)

Aufgabe 3) Die Methode **read_lexicon** liest die Lexikondatei ein und speichert die Grammatikregeln $A \rightarrow w$ so in einer Datenstruktur *self.lexprobs*, dass man für ein gegebenes Wort w direkt alle Wortarten A und ihre Wahrscheinlichkeiten erhält. Jede Zeile der Lexikondatei enthält eine Wahrscheinlichkeit, eine Wortart und ein Wort bspw.

```
0.1 DT the
0.002 N man
...
```

(2 Punkte)

Aufgabe 4) Die Methode **scan** erhält 2 Argumente: ein Wort und seine Position. Sie schlägt das Wort in *lexprobs* nach und trägt für jede erhaltene Wortart A die Punktregel $A \rightarrow w \cdot$ mit ihrer Wahrscheinlichkeit in die Chart ein. (1 Punkt)

Aufgabe 5) Die Methode **predict** erhält 2 Argumente: ein Nichtterminal und eine Position. Sie trägt alle Regeln mit diesem Nichtterminal auf der linken Seite wie oben beschreiben in die Chart ein. (8 Punkte)

Aufgabe 6) Die Methode **complete** erhält 2 Argumente: ein Punktregel-Tupel und seine Wahrscheinlichkeit. Sie führt die Complete-Operation aus. (1 Punkt)

Aufgabe 7) Die Methode **add** erhält 2 Argumente: ein Punktregel-Tupel und seine Wahrscheinlichkeit. Sie trägt die Punktregel in die Chart ein, falls sie noch nicht darin enthalten war, oder falls ihre Wahrscheinlichkeit größer als die der bereits eingetragenen Regel ist. Außerdem wird die Methode *complete* aufgerufen, falls der Punkt auf der rechten Seite der Punktregel am Ende angekommen ist. Die Methode *predict* wird aufgerufen, falls der Punkt auf der rechten Seite der Punktregel noch nicht am Ende angekommen ist. (8 Punkte)

Aufgabe 8) Schreiben Sie zum Schluss noch eine Methode **parse**, welche einen Satz (als Liste von Wörtern) als Argument erhält. Sie ruft zunächst die Methode *predict*

mit dem Startsymbol *self.start_symbol* und der Startposition 0 auf. Dann ruft Sie die Funktion *scan* mit jedem Wort des Satzes und seiner Position auf. Zum Schluss prüft die Methode, ob eine **vollständige Analyse** des Satzes gefunden wurde und gibt seine **Wahrscheinlichkeit** aus. (5 Punkte)

Bitte halten Sie sich bei den Aufgaben genau an die Anleitungen und prüfen Sie, ob die Wahrscheinlichkeiten der Punktregeln korrekt berechnet werden.

(30 Punkte insgesamt)

Viel Erfolg!