Vorlesung "Computerlinguistische Techniken"

6. Übung (08.12.2015)

Wintersemester 2015/16 – Prof. Dr. Alexander Koller

Verwenden Sie in dieser Übung geeignete Klassen und Methoden aus NLTK, um sich die Arbeit soweit wie möglich zu vereinfachen. Besonders relevante Module sind `nltk.grammar`, `nltk.tree` und `nltk.parse.viterbi`. Denken Sie an die Konvention, dass Methoden, deren Namen mit einem Unterstrich anfangen, private Methoden der Klasse sind und nicht zur Verwendung durch Benutzer (= Sie) gedacht sind.

1 Parsing mit der Penn Treebank

a) Mit `nltk.corpus.treebank.parsed_sents()` erhalten Sie die Liste aller Bäume aus einem Fragment der Penn Treebank, das bei NLTK mitgeliefert ist. Konvertieren Sie alle diese Bäume in Chomsky-Normalform (mit `collapse_unary(collapsePOS=False)` und `chomsky_normal_form`).

   Wir verwenden die ersten 90% der konvertierten Baumbank als Trainingskorpus. Schätzen Sie aus diesem Trainingskorpus mit dem Maximum-Likelihood-Verfahren eine PCFG-Grammatik. Sammeln Sie dazu die Produktionsregeln aus der Baumbank in einer Liste auf und verwenden Sie die Funktion `induce_pcfg` mit dem Startsymbol S. Die Produktionsregeln eines Parsebaumes bekommt man mit der `productions`-Methode.

b) Als Testkorpus verwenden wir diejenigen Sätze in den letzten 10% der konvertierten Baumbank, die höchstens zehn Tokens lang sind. Versuchen Sie, das Testkorpus mit dem Viterbi-Parser zu parsen. Was passiert? Warum?

c) Erweitern Sie Ihr System um Add-One-Smoothing für Terminalproduktionen. Das bedeutet, dass für jedes Wort w, das mindestens einmal im Trainings- oder Testkorpus vorkommt, und jedes POS-Tag A im Trainingskorpus die absolute Häufigkeit der Produktionsregel $A \to w$ um 1 erhöht wird. Erst danach schätzen Sie mit dem ML-Training die

Regelwahrscheinlichkeiten ab. Parsen Sie dann das Testkorpus nochmals.

d) Machen Sie für diejenigen besten Parsebäume, die der Viterbi-Parser gefunden hat, die CNF-Konvertierung rückgängig, um die Bäume direkt mit dem Goldstandard vergleichbar zu machen (`un_chomsky_normal_form`). Geben Sie labeled und unlabeled Precision, Recall und F_1-Score an. Sie können dazu die Funktionen aus `parseval.py` (siehe Moodle) verwenden.

e) Vergleichen Sie von Hand die 28 Goldstandard-Bäume für das Testkorpus mit den Viterbi-Parses. Erkennen Sie wiederkehrende Fehler, die der Parser typischerweise macht? Woran liegt es, dass für manche Sätze gar kein Parsebaum gefunden werden konnte?

Abgabe bis 08.01.2016, vor der Vorlesung, per Mail an `mgerdes@uni-potsdam.de`