Übung 04: Sequence Tagging

Programmiertechniken in der Computerlinguistik II, FS 16

Abgabetermin: 05.05.2016

Hinweise zur Abgabe

- Bitte gib jedes Python-Programm in einer eigenen Datei ab, die die Dateiendung `.py` hat und *ausführbaren* Python-Code enthält.
- Geize nicht mit Kommentaren direkt im Programm-Code, wo Erläuterungen angebracht sind. Umfangreiche Erklärungen werden hingegen besser in einer separaten README-Datei mitgeliefert (vorzugsweise Plain-Text oder PDF).
- Um das Hochladen der Abgabe auf OLAT zu erleichtern, kannst du die Dateien mit `zip` oder `tar` (oder einem anderen verbreiteten Format) archvieren / komprimieren.

1 POS-Tagging

In dieser Aufgabe geht es darum, selbst einen POS-Tagger zu implementieren. Der Tagger soll ein statistischer Bigramm-Tagger sein. Als Trainingskorpus kannst du getaggte Sätze aus einer Kategorie deiner Wahl des Brown Corpus verwenden.

a) Teile dein Korpus in zwei Teile zu je 90 und 10%. Der grössere Teil soll dein Trainingskorpus sein, der kleinere dein Testkorpus.

b) Implementiere deinen Tagger. Du musst deinen Tagger selber schreiben; es ist nicht erlaubt, die fertigen Tagger von NLTK zu verwenden. Du darfst externen Code benutzen, aber du solltest zeigen, dass du verstanden hast, wie der Tagger genau funktioniert. Wenn der Tagger ein Bigramm noch nicht gesehen hat, sollte er es als unbekannt taggen. Teste deinen Tagger auf dem Satz "This is a sentence that we want to tag." Welches Problem tritt auf?

c) Schreibe eine Funktion `evaluate(tagger, tagged_sents)`, die die Genauigkeit eines Taggers berechnet. Evaluiere deinen Tagger auf deinem Testkorpus. Wie genau ist er?

d) Erstelle eine Konfusionsmatrix für deinen Tagger. Bei welchen Tags macht er besonders viele Fehler?

2 Backoffs

In dieser Aufgabe sollst du den Tagger aus der letzten Aufgabe erweitern, sodass er unbekannte Kontexte mit Backoffs auflöst.

Schreibe dazu zwei weitere Tagger: einen Unigramm-Tagger und einen Default-Tagger. Ändere deinen Tagger so, dass er Wörter in unbekannten Kontexten mit dem Unigramm-Tagger taggt. Der Unigramm-Tagger sollte wiederum bei unbekannten Wörtern auf den Default-Tagger zurückgreifen.

a) Trainiere und evaluiere den neuen Tagger auf den gleichen Korpora wie in der letzten Aufgabe. Gibt es einen Unterschied in der Genauigkeit?

b) Evaluiere deine Tagger auf einer anderen Kategorie des Brown-Korpus. Wie verändert sich ihre Genauigkeit? Wie sieht es aus, wenn du sie über dem conll2000-Korpus evaluierst?

Reflexion/Feedback

a) Fasse deine Erkenntnisse und Lernfortschritte in zwei Sätzen zusammen.

b) Wie viel Zeit hast du in diese Übungen investiert?