

Prof. Dr. Marcus Spies
Christian Leibold

Übungen zur Vorlesung
Automatische Spracherkennung und integrierte Dialoganwendungen
Sommersemester 2005

Blatt 3

Im letzten Übungsblatt haben Sie bereits den LiveDecoder verwendet um die Auswirkungen von Einstellungen am Sphinx-Frontend zu messen. In diesem Übungsblatt werden Sie mit Hilfe des Tools Ihre eigene Grammatik in Sphinx einfügen und verwenden können.

Aufgabe 3-1 Erstellung einer eigenen Decoderinstanz:

1. Kopieren Sie sich für die Bearbeitung der Aufgabe die Dateien des Decoders „Player“ in ein neues Verzeichnis unterhalb von <sphinx-home>/tests/live und benennen die Dateien entsprechend um.
2. Tragen Sie den Decoder in der Datei <sphinx-home>/tests/live/decoders.list ein.
3. Passen Sie die Einstellung in der .config.xml an:
 - a. Verwendete Grammatik (bsp: `<property name="grammarName" value="tutoriumTest.tutoriumTest"/>`)
 - b. Hinzufügen des linguistischen Modells in der „linguist configuration“ (`<property name="phoneLoopAcousticModel" value="wsj"/>`)
 - c. Deaktivieren der strikten Grammatikbindung in der „linguist configuration“ (`<property name="addOutOfGrammarBranch" value="false"/>`)
4. Starten Sie nun den LiveDecoder. Bei korrekter Eingabe der Informationen in der Decoderliste können Sie nun auf Ihre Decoderinstanz zugreifen. Diese sollte nun die selben Wörter erkennen wie die Ursprungsgrammatik.

Aufgabe 3-2 Erstellung einer eigenen Grammatik: (Abgabe erforderlich)

Verändern Sie die neue Grammatikdatei .gram um eine Grammatik zu erstellen, die die Sätze aus der Datei tutorialTest.test erkennt. (Download auf der Übungsseite)

Anmerkungen und Tipps:

- Das von „Player“ verwendete linguistische Modell finden Sie unter <sphinx4>\models\acoustic\wsj\
- Die Datei „cmudict.0.6d“ (<sphinx4>\models\acoustic\wsj\dict) enthält die komplette Liste erkennbarer Worte mit den dazugehörigen Baseformen.
- Vergleichen Sie verschiedene .gram Dateien um eine sinnvolle, möglichst kleine Grammatik abzuleiten.

Um sicherzustellen, dass eine minimale Grammatik bzgl. der Aussagen über Orte angegeben zu haben, sollen folgende Sätze nicht erkannt werden:

„we are german“

„she is from germany“

“she lives in five two eight rome italy”

(Setzen Sie bei Bedarf in der Linguist configuration das Property `<property name="addOutOfGrammarBranch" value="true"/>`. Bei den Sätzen sollten Sie nun als Erkennungsergebnis <unk>, für „unknown“ erhalten.)

Aufgabe 3-3 Bewertung der Ausgabe auf der Shell: (Abgabe erforderlich)

Beim Test Ihrer Grammatik haben Sie in der Standardausgabe neben den bekannten Statistiken und erkannten Wörtern auch noch die erkannten Baseformen gesehen.

- a) Wie korrelieren diese Baseformen mit den HiddenMarkovModellen (HMMs)?
- b) Wie werden die HMMs in Sphinx aufgebaut, welche Rolle spielen hierbei die eingesetzten Komponenten (Grammatik, Dictionary, Decoder, Linguistisches Modell)? Nehmen Sie hierbei Bezug auf die verwendeten Klassen und den Datenfluß.

Aufgabe 3-4 Ausgabe der aktuellen Scores aller Wörter: (freiwillig)

Wie wir beim gemeinsamen Erarbeiten der Sphinx4 – Architekturübersicht gesehen haben, stehen dem Recognizer während dem Erkennungsprozess mehrere mögliche Ergebnisse zur Verfügung. Die Decoder aus den Beispielanwendungen verwenden aber immer nur das wahrscheinlichste Wort aus dem „result“. In diesem Result ist aber auch die aktuelle „ActiveList“ enthalten. Diese enthält alle „möglichen“ Wörter mit der aktuellen Anzahl der Tokens (gesehen zur Gesamtsumme der Tokens also die relative Wahrscheinlichkeit).

Erweitern Sie eine beliebige Demo der Art, dass nun nicht das erkannte Wort ausgegeben wird, sondern eine Liste der n wahrscheinlichsten Wörter.

Zur Bearbeitung der Aufgaben können Sie in Teams zu ZWEI Personen arbeiten. Abgabe der Lösungen zu den MARKIERTEN Aufgaben bis Dienstag, 05.07.2005, 12 Uhr an Christian.Leibold@stud.ifi.lmu.de. Namen und Matrikelnummer nicht vergessen!