

Übung zur Vorlesung “Einführung in die Computerlinguistik und Sprachtechnologie”

Wintersemester 2017/2018, Prof. Dr. Udo Hahn, Sven Büchel

Übungsblatt 6 vom 01.12.2017

Abgabe bis 05.12.2017, 23.59 Uhr; per Email (PDF-Format) an sven.buechel@uni-jena.de

Aufgabe 1 Quantoren-Skopus

2

Suchen Sie ein neues Beispiel für die Ambiguität bezüglich des Quantoren-Skopus und diskutieren Sie die möglichen Interpretationsvarianten dieses Beispiels.

Aufgabe 2 Ambiguität

2

Welche Formen von Ambiguität (lexikalisch/syntaktisch) liegen bei den folgenden Äußerungen vor? Geben Sie alle möglichen Lesarten an.

1. Die Kinder haben den Kuchen mit dem Löffel gegessen.
2. Die LIEGEN im Gras ...
3. breast cancer therapy and prevention

Eine besondere Form syntaktischer Ambiguität sind sogenannte *garden-path* Sätze wie der folgende (bereits bekannte?) Satz. Warum? Welche andere Form von Ambiguität verursacht hier diese syntaktische?

- The horse raced past the barn fell.

Aufgabe 3 Algorithmus zur Satzsegmentierung

6

Eine der grundlegendsten Aufgabe in der Computerlinguistik ist es, in einem Fließtext Sätze zu erkennen. Ein Programm zur Satzerkennung bekommt als Eingabe einen Text, beispielsweise “Peter lief nach draußen. Seine Jacke wurde vom Regen durchweicht.”. Die Ausgabe des Programms wären dann die Sätze “Peter lief nach draußen.” und “Seine Jacke wurde vom Regen durchweicht.”

a) 1

Eine naive Herangehensweise zur Satzerkennung ergibt sich, wenn jeder Punkt im Text als Satzgrenze aufgefasst wird. Welche Probleme sehen Sie bei diesem Vorgehen? Nennen Sie Beispiele.

b) 3

Geben Sie einen einfachen Algorithmus an, der einen Text entgegen nimmt und diesen in einzelne Sätze zerlegt. Die entsprechende Funktion soll eine Liste von Sätzen zurückgeben. Hinweis:

- Als Hilfestellung sei die Funktion `getWords(text)` gegeben. Diese Funktion bekommt einen Text, trennt ihn an Leerzeichen und gibt die so erhaltenen Wörter (ggf. mit Interpunktion!) in einer Liste zurück. Die Leerzeichen selbst gehen dabei verloren. Beispiel: `words ← getWords("Peter lief nach draußen.")` zerlegt den eingegeben Satz in die Elemente "Peter", "lief", "nach" und "draußen." (man beachte, dass das letzte Element auch den Punkt enthält, da er nicht durch ein Leerzeichen abgetrennt ist). Die Variable `words` enthält nun alle diese Wörter. Wie gehabt kann mit `words[0]` auf das 1te Wort, mit `words[1]` auf das 2te Wort etc. zugegriffen werden.

c) 2

Verfeinern Sie ihren Algorithmus, indem sie versuchen die Probleme, die Sie in Teilaufgabe a) identifiziert haben, zu lösen. Nehmen Sie dafür die benötigten Listen mit bestimmten Wörtern oder Funktionen zur Identifikation bestimmter

problematischer Strukturen als gegeben an. Falls Sie solche Listen oder Funktionen verwenden, geben Sie bitte deren Inhalt bzw. Zweck an.