

ÜBUNGSKLAUSUR ZUR VORLESUNG

Teilweise aus Aufgaben der im Netz verfügbaren Übungsklausuren anderer CL Standorte in Deutschland übernommen.

Einführung in die Computerlinguistik

WS 2011/2012

CHRISTOPH RINGLSTETTER

Es dürfen keine Hilfsmittel verwendet werden. Aufgabe 8 ist eine Zusatzaufgabe. Zu erreichen sind also die 100 Punkte der Aufgaben 1-7. Etwa fehlende Punkte können mit Punkten aus Aufgabe 8 ausgeglichen werden.

Aufgabe 1 (10 Punkte)

Nennen und beschreiben Sie die verschiedenen Teilbereiche der Linguistik, die auch bei der automatischen Verarbeitung von Sprache involviert sind. Geben Sie jeweils einen Satz an, worum es in diesen Teilbereichen geht. Und einen weiteren Satz, in welchem Bereich der Computerlinguistik diese Teilbereiche besondere Relevanz haben.

Aufgabe 2 (15 Punkte)

Geben Sie die verschiedenen Teilsysteme eines automatischen Dialogsystems an, wie es etwa die telefonische Bahnauskunft darstellt. Sagen Sie zu jedem Teilsystem einen Satz, was es tut und einen Satz wo Sie erläutern, ob Sie das jeweilige Teilsystem schon als reif im Sinne der Ziele computerlinguistischer Anwendungen sehen oder ob Sie noch großen Grundlagenforschungsbedarf sehen. Sagen Sie in einem Satz worin Sie den Unterschied etwa zwischen dem Bahnsystem und dem Watson-System sehen.

Aufgabe 3 (15 Punkte)

- Erklären Sie die Begriffe Phon, Phonem, Allophon.
- Erklären Sie die Begriffe Morph, Morphem, Allomorph.
- Zerlegen Sie die folgenden Wörter morphologisch, weisen Sie den Teilen die jeweilige Funktion zu:
"Mäuse, ablegen, Verbraucherschutz, mütterliches"
- Geben Sie für die Wörter mögliche Einträge in einem Lexikon an. Überlegen Sie welche Informationen Sie im Lexikon ablegen würden, welche nicht.
- Geben Sie 3 mögliche Schwierigkeiten bei der Tokenisierung englischer Texte an.

Aufgabe 4 (20 Punkte)

(a) Schreiben Sie eine kontextfreie Grammatik für deutsche Nominalphrasen wie „der kleine Mann“, „ein kluges Kind“, „ein kleines kluges kleines Kind, ein kleiner kluger kluger kleiner ... Mann, usw. Beachten Sie hierbei, dass nur korrekte Formen generiert werden (also nicht eine kleiner Kind). Beschränken Sie sich auf den Nominativ. Vergessen Sie die Terminalsymbole nicht.

(b) Geben Sie den Strukturbaum für „ein kleiner kluger Mann „ an.

(c) Zeigen Sie Schritt für Schritt, wie die NP ein kleiner kluger kleiner Mann von einem Top-Down-Parser erkannt würde, der Tiefe-Zuerst vorgeht und immer das am weitesten links stehende Symbol zuerst ersetzt.

Aufgabe 5 (10 Punkte)

a) Was ist anaphorische Referenz, d.h., wann sagt man, dass eine Äußerung anaphorisch referiert? Welche Äußerungen im folgenden Beispiel sind als Anapher benutzt? (Markieren Sie sie.)

Anita hat ein Buch gekauft. Sie hat es heute in der Mensa Thomas gezeigt. Er fand das Buch interessant.

(b) Nennen Sie die konversationellen Maximen (maxims of conversation) von Grice und erläutern Sie kurz, was die jeweiligen Maximen besagen.

Geben Sie die Maxime für die konversationelle Implikatur an, auf die man aus der Antwort im folgenden Beispiel aufgrund der Maxime schließt.

Frage: Hat Hans Maria schon erzählt, dass er ihr Auto kaputt gemacht hat?

Antwort: Hans hat heute mit Maria gesprochen und sie hat sich sehr geärgert.

(c) Was ist ein Sprechakt (auch Illokutionärer Akt genannt)?

Woran erkennt man den Sprechakt einer Aussage?

Aufgabe 6 (15 Punkte)

Erklären Sie kurz was der unten angegebene Algorithmus zur Kompositazerlegung macht. Kommentare sind wie in PERL mit # markiert. Eingabe ist ein String s der aus den Zeichen $s = s_0..s_n$ besteht. Spielen Sie den Algorithmus für das potentielle Kompositum Verbraucherschutz durch. Im Lexikon seien

FF: Verb, Verbraucher, Raucher, Schutz

KK: Verb, Verbraucher, Raucher, Schutz, angegeben.

findSplitting(s) {

for $i = 1..i = n$ do

$T_1 = s_0..s_{i-1}$

 #prüft ob T_1 lexikalisch ist und ein gültiges Erstglied ist (=Fugenform FF)

 if $L(T_1) == FF$ then

$T_2 = s_i..s_n$

 #prüft ob T_2 lexikalisch ist und eine gültige Kopfkongstituente ist (=KK)

 if $L(T_2) == KK$ then

 push splittings, $T_1_T_2$

 #zweiter Wortteil nicht-lexikalisch, aber möglicherweise weiter zerlegbar
 else

 splittings $T_2 = findSplitting(T_2)$ #Rekursiver Aufruf des Algorithmus

 for all $s \in splittingsT_2$ do

 #Aufgespaltenes T_2 mit dem gültigen Erstglied verbinden

$s = T_1_s$

 end for

 end if

 #schreiben auf die Datenstruktur

 push splittings, splittings T_2

 end if

end for

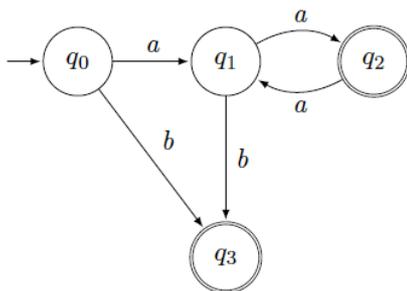
```
return splittings
}
```

Aufgabe 7 (15 Punkte)

- (a) Erläutern Sie drei der in Wordnet angegebenen semantischen Relationen.
- (b) Geben Sie in einem oder zwei Sätzen an, was nach Ihrer Meinung die Wortsemantik von der Satzsemantik unterscheidet.
- (c) Geben Sie folgende Sätze einmal in aussagenlogischer und zum anderen in prädikatenlogischer Schreibweise an (falls das überhaupt geht).
- c1) Rabe A und Rabe B sprechen.
 - c2) Rabe A spricht, falls Rabe B das auch tut.
 - c3) Alle Raben sprechen
 - c4) Einige Raben sprechen.

Zusatzaufgabe 8 (20 Punkte)

(a)



- (a) Geben Sie den Automaten in Tupelschreibweise an. D.h., als Tupel $\langle Q, \Sigma, \delta, q_0, F \rangle$.
- (b) Geben Sie den regulären Ausdruck der Sprache an, die von dem Automaten akzeptiert wird.

(b) Gegeben ist der folgende nicht-deterministische Automat

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

$\Sigma = \{a, b\}$

$(q_0, a) = \{q_1, q_2\}$

$(q_1, a) = \{q_3\}$

$(q_3, a) = \{q_1\}$

$(q_2, b) = \{q_2\}$ $(q_2, a) = \{q_4\}$

q_0 ist der Startzustand

$F = \{q_3, q_4\}$

(a) Woran erkennt man, dass dieser Automat nicht deterministisch ist? (deterministisch heisst, dass der Weg durch den Automaten für jede gültige Eingabe eindeutig ist)

(b) Zeichnen Sie den NFA.

(c) Welche Sprache wird vom Automaten akzeptiert

(c) Schauen Sie sich folgende Sequenz aus dem Brown Korpus an:

which/wdt are/ber genuine/jj protests/nns of/in our/pp\$ own/jj

(c1) geben sie an was die Wortartentags im Beispiel bedeuten.

(c2) Nehmen Sie an dass in der ersten Runde des Brillverfahrens

"protests" als Verbform 3.Pers.Sing. (vbz) statt als nominaler Plural getaggt wird.

Geben Sie ein Template und eine Regel an, die in der nächsten Runde den Fehler bereinigen kann.