

Sprachverarbeitung: Übung 18

Sprachsynthese: Akzentuierung und Phrasierung

In dieser Übung werden Sie nun die Gelegenheit haben, mit einem einfachen Synthesesystem in seiner Gesamtheit zu experimentieren. Die dafür benötigten Teile wurden fast alle in vergangenen Übungen bereits implementiert. Einzig die Akzentuierung und Phrasierung von Sätzen fehlt noch. Diese Teile sind im Rahmen dieser Übung zu realisieren.

Aufgabe 1: Synthesesystem ohne Akzentuierung und Phrasierung

Das komplette Synthesesystem können Sie mit dem Matlab-Befehl `tts_synthesis(inp)`; starten. Zum einfacheren Testen sind bereits zehn Sätze vorgegeben, die Sie der Synthese übergeben können, indem Sie für `inp` die entsprechenden Dateinamen `'satz01.txt' ... 'satz10.txt'` angeben. Sie können jedoch auch andere Sätze direkt synthetisieren lassen, also beispielsweise durch `tts_synthesis('die kinder besuchen den hausmeister des museums')`.

Die Synthese erfolgt in vier Schritten:

1. **Morphologische und syntaktische Analyse:** Erstellt einen Syntaxbaum analog zu den Übungen 16 und 17, mit dem Unterschied, dass hier nicht die Graphik benutzt wird, sondern der ausgegebene Syntaxbaum in Klammernotation (siehe Matlab-Funktion `parse-text`). Um Ihre eigene Grammatik aus der Übung 18 zu nutzen, kopieren Sie diese als `ueb18.dat` in Ihr Übungsverzeichnis.
2. **Akkentuierung und Phrasierung:** Die Betonung innerhalb des Satzes und Einteilung der Phrasen werden Sie in dieser Übung implementieren.
3. **Prosodiebestimmung:** Die Erzeugung der Grundfrequenzwerte und der Lautdauerwerte beruht auf den linearen Modellen aus den Übungen 7 und 8. Um Ihre eigenen Lösungen zu benutzen, kopieren Sie die Dateien `get_dur_rule_conditions.m` und `DurModel1.mat` aus Übung 7 und `get_F0_rule_conditions.m` und `F0Model1.mat` aus Übung 8 in Ihr Übungsverzeichnis.
4. **Signalgenerierung:** Das Sprachsignal wird mittels Diphon-Verkettung erzeugt, wobei die Dauer und die Grundfrequenz der Diphone angepasst werden. Dafür werden die Funktionen LPC-Analyse und LPC-Synthese aus den Übungen 5 und 6 eingesetzt. Auch hier können Sie Ihre eigenen Lösungen verwenden, indem Sie `LPC_analysis.m` und `LPC_synthesis.m` aus Übung 5 kopieren.

Für diese Übung wird standardmässig eine gegenüber der Übung 17 erweiterte Grammatik verwendet, die unter `Uebung18/Gegebenes/ueb18.dat` zu finden ist. Schauen Sie sich diese Grammatik an und versuchen Sie zu verstehen, welche zusätzlichen Satzformen und Konstituenten (Satzteile) akzeptiert werden. Testen Sie die Sprachsynthese mit Beispielsätzen und schauen Sie sich die Ausgabe an. Hören Sie sich auch das generierte Signal an, dem vorerst noch die Satzbetonung und Phraseneinteilung fehlen.

Aufgabe 2: Akzentuierung: Implementierung der “Nuclear Stress Rule”

Für die Akzentuierung eines Satzes aufgrund seines Syntaxbaumes ist die Matlab-Funktion `tree = accentuation(tree)` zu verwirklichen. Von dieser Funktion ist das meiste bereits in der Datei `accentuation_frame.m` vorgegeben. Dort wird, wie im Abschnitt 8.4.6.2 des Buches beschrieben, bereits die Initialisierung der unbetonbaren Wörter vorgenommen und auch die Iterationsschleife (also das Zusammenfassen der Teilkonstituenten) der “Nuclear Stress Rule” (NSR), sowie die Rhythmisierung. Ihre Aufgabe ist es, die eigentliche Satzakkzentuierung zu implementieren, also das Abschwächen der Wortakzente in zusammengefassten Teilkonstituenten.

Kopieren Sie dazu zuerst die Datei `Uebung18/Gegebenes/accentuation_frame.m` in Ihr Übungsverzeichnis und geben Sie ihr den Dateinamen `accentuation.m`. Fügen Sie nun an der markierten Stelle den Matlab-Code für die NSR ein. Um nicht stets die komplette Synthese ausführen zu müssen, können Sie Ihre Funktion während der Entwicklung mit `test_accentuation(sent_num)` an den gegebenen Beispielsätzen testen, wobei `sent_num` eine Zahl zwischen 1 und 10 ist.

Der Syntaxbaum des Satzes wird der Funktion als Array von Strukturen übergeben, wobei jedes Array-Element einen Knoten des Syntaxbaumes repräsentiert und die Struktur die folgenden Elemente aufweist:

id	Knoten ID (entspricht Array-Index des Knotens)
path	Pfad vom Wurzelknoten zu diesem Knoten (Array von IDs)
name	Name der Konstituente
level	Hierarchieebene im Syntaxbaum
term	Terminalelement im Syntaxbaum
graph	graphemischer String, falls Wortkonstituente
phon	phonetischer String, falls Wortkonstituente
sylnr	Anzahl der Silben
accnr	Anzahl der betonten Silben
acc	Akzentstärke des Worthauptakzents

Einen Syntaxbaum können Sie auch mit dem Befehl `tree = morphosyntactic_analysis(inp);` erzeugen, wobei `inp` wiederum entweder einen Satz oder den Namen einer Textdatei mit einem Beispielsatz erwartet. Danach können Sie den Baum mit der Funktion `disptree` graphisch oder textlich anzeigen lassen oder mittels `tree(i)` auf das i-te Element (also den i-ten Knoten) des Baumes zugreifen.

Aufgabe 3: Phrasierung

Die Phrasierung folgt dem im Abschnitt 8.4.7 des Buches beschriebenen Verfahren und ist in der Matlab-Funktion `phrasing(tree)` implementiert. Dabei ist das Bestimmen der initialen Phrasengrenzen zwischen den einzelnen Wörtern von entscheidender Bedeutung. Ihre Aufgabe ist es, diesen Teil zu implementieren.

Kopieren Sie dazu die Datei `Uebung18/Gegebenes/phrasing_frame.m` in Ihr Übungsverzeichnis und geben Sie ihr den Dateinamen `phrasing.m`. Implementieren Sie die Initialisierung an den kommentierten Stellen. Achten Sie dabei darauf, dass Satzzeichen gesondert behandelt werden müssen, da sie zwar keine Wörter sind, jedoch selbst eine Phrasengrenze bilden.

Zum Testen Ihrer Funktion während der Entwicklung steht Ihnen wiederum eine Testfunktion zur Verfügung, nämlich `test_phrasing(sent_num)`. In dieser Testfunktion wird auch die Akzentuierungs-Funktion gebraucht. Falls Sie die Aufgabe 2 noch nicht gelöst haben, dann können Sie anstelle der Funktion `accentuation.m` auch entweder `accentuation_frame.m` oder `accentuation_sol.m` für den Test einsetzen. Letztere ist im Unterverzeichnis *Lösung* zu finden.

Aufgabe 4: Synthesystem mit Akzentuierung und Phrasierung

Testen Sie nun die ganze Sprachsynthese mit der Funktion `tts_synthesis(inp,vers,fign)` und vergleichen Sie insbesondere die gegebenen Funktionen für Akzentuierung und Phrasierung mit Ihren eigenen und mit denen im Unterverzeichnis *Lösung*. Die Funktionen können Sie mit dem Argument `vers = 'frame' | '' | 'sol'` wählen. Mit dem Argument `fign` können Sie zudem bestimmen, in welchem Figurenfenster die graphische Ausgabe erscheinen soll. So sind z.B. auch die graphischen Darstellungen der Grundfrequenzverläufe, die mit den verschiedenen Akzentuierungs- und Phrasierungsfunktionen entstehen, miteinander vergleichbar.