

## 11. Übungsblatt - Abgabe: 27.01.2015

### Aufgabe 11.1 - Sprachsynthese

Lesen Sie aus dem CL-Handbuch (K.-U. Carstensen et al., Computerlinguistik und Sprachtechnologie, 3. Auflage 2010) den Artikel 3.2.2 Sprachsynthese. Eine Kopie steht im Vorlesungsordner in der Bibliothek.

- (a) Der Artikel zeigt an einem Beispiel, dass für eine korrekte Aussprache die Kenntnis über die interne morphologische Struktur des Wortes relevant ist. Geben Sie (mindestens) zwei weitere (möglichst vom Typ her unterschiedliche) Beispiele, und kommentieren Sie sie kurz.
- (b) Der Artikel führt (im Abschnitt „Perspektiven“ über „concept-to-speech“) aus, dass gute Sprachsynthese ein gewisses Semantik-, Pragmatik- und Kontextwissen voraussetzt.

Betrachten Sie den folgenden Satz: *Die Baustelle bitte großräumig umfahren.* Erklären Sie, worin die Schwierigkeit für eine Sprachsynthese besteht, welches Wort insbesondere Probleme bereitet und welches Wissen nötig ist, um die richtige Aussprache dieses Wortes zu finden.

- (c) Der Satz *Peter hat den Kuchen gegessen.* kann unter anderem Antwort auf die folgenden Fragen sein:
- Wer hat den Kuchen gegessen?
  - Was hat Peter gegessen?
  - Isst Peter gerade den Kuchen?

Beschreiben Sie, wie sich die Aussprache des Satzes als Antwort auf die verschiedenen Fragen jeweils verändert. Versuchen Sie, eine Regel abzuleiten, die eine Aussage darüber trifft, welcher Satzteil betont wird (in Abhängigkeit von neuer/gegebener Information).

- (d) Die Firma Nuance, zurzeit fast Monopolist für Sprachsoftware, bietet ein Online-Interface für TTS an:

<http://www.nuance.com/for-business/text-to-speech/vocalizer/index.htm#demo>  
Wörter/Sätze werden eingetippt und mit einer vorher ausgewählten Stimme akustisch realisiert. Probieren Sie das Demo-System aus, indem Sie zunächst eine deutsche Stimme wählen und dann eine Reihe von Beispielen eintippen (Beispiele aus dem Artikel oder frei gewählte). Beschreiben Sie stichwortartig, welche Aspekte beim System gut/weniger gut behandelt werden bzw. zu Fehlern führen.

- (e) Geben Sie die Ausdrücke, die Sie zu Frage (a) gefunden haben, sowie den Satz aus (b) und die Frage-Antwortpaare aus (c) ein, und beschreiben Sie den Effekt. Wenn das System in allen Fällen korrekt verbalisiert, suchen Sie nach ähnlichen Alternativen, bei denen das System Fehler macht.

### Aufgabe 11.2 - n-Gramme

- (a) Welcher der folgenden Sätze hat die höhere Bigramm-Wahrscheinlichkeit? Schätzen Sie die benötigten Bigrammwahrscheinlichkeiten (also  $P(w_n|w_{n-1})$ ), indem Sie die Begriffe entsprechend googlen und aus den Trefferzahlen die Wahrscheinlichkeiten abschätzen. Nehmen Sie dazu an, dass google auf 20 Milliarden Webseiten arbeitet. Konfigurieren Sie google so, dass nur deutsche Webseiten gefunden werden und setzen Sie die Bigrammbegriffe gemeinsam in Anführungszeichen.
- Saarbrücken liegt im Saarland.
  - Saarbrücken liegt im Garten.
- (b) Entspricht das Ergebnis Ihrer Intuition? Wie kommt es zustande? Was kann durch Bigramme also nicht modelliert/abgedeckt werden?
- (c) Geben Sie zwei weitere Beispielpaare von Satzpaaren, bei denen der Satz mit der geringeren Bigrammwahrscheinlichkeit der linguistisch plausiblere ist. Finden Sie darunter auch ein Beispiel, bei dem der wahrscheinlichere Satz syntaktisch/morphologisch falsch ist und schätzen Sie die Wahrscheinlichkeiten wieder mit Google-Counts.
- (d) Finden Sie einen deutschen Satz mit 5 Wörtern, der eine möglichst hohe Bigrammwahrscheinlichkeit hat. Beschreiben Sie Ihr Vorgehen.
- (e) Finden Sie eine beliebige Folge von 5 Wörtern mit möglichst hoher Bigrammwahrscheinlichkeit.

Hinweis: Es reicht, wenn Sie in Aufgabe (a) und (c) in jedem Satzpaar diejenigen Bigrammwahrscheinlichkeiten bestimmen, die sich für beide Sätze unterscheiden.

### Aufgabe 11.3 Unix-Tools

In dieser Aufgabe sollen Sie die Evaluation aus Aufgabe 11.2 auf dem Tiger-Corpus nachvollziehen.

- a) Bauen Sie sich eine Tiger-Version zusammen, die auf jeder Zeile jeweils ein tab-separiertes Bigramm von Wörtern enthält. Benutzen Sie dazu das Tiger-Corpus aus Woche 3 und einige der bisher bekannten Linux-Tools. (Hinweis: Am einfachsten ist es, wenn Sie sich zwei Versionen der tiger-Datei erstellen, bei der eine am Anfang eine extra Leerzeile enthält und die dann zeilenweise aneinanderhängen.)
- b) Ermitteln Sie dann mit Hilfe von *grep* auf dem Tiger-Corpus die Bigrammwahrscheinlichkeiten und beschreiben Sie ihr vorgehen. Auf welches konzeptuelle Problem stoßen Sie? Wie kommt es dazu?

- c) Berechnen Sie jetzt unter dem Tiger-Corpus die Wahrscheinlichkeit für die beiden Sätze mit dem Bigramm-Sprachmodell. Lassen Sie dabei die Wahrscheinlichkeit für *Saarbrücken liegt* komplett weg. Sie sehen hier ein anderes Ergebnis als in der google-Analyse. Wie lässt sich dieses Ergebnis im Hinblick auf die verwendeten textsorten erklären? Was bedeutet das für die Auswahl der Sprachmodell-Daten in einer echten Spracherkennungsaufgabe?

---

Abgabe in Gruppen von bis zu drei Studierenden am **27.01.2015** vor der Vorlesung.