

Sprachmodelle

Hauptseminar: Dialogsysteme

Dozenten: Prof. Dr. Klaus Schulz,

Christoph Ringlstetter, Florian Schiel

Referentinnen: Galina Trefilova,

Susanne O'Shaughnessy

12.01.2006

Wozu braucht man ein Sprachmodell?

I recognize speech.

Zwei mögliche Hypothesen eines Spracherkenners:

- * I wreck a nice beach.
- * I recognize speech.

Sprachmodelle: Aufgabe

Die Aufgabe des Sprachmodells besteht darin:

- * Die Wortfolgen in Texten linguistisch zu bewerten;
- * Die Wahrscheinlichkeit einer Sequenz von Wörtern zu berechnen (Satzvervollständigung);
- * Den Suchraum für mögliche folgende Wörter einzuschränken;
- * In Kombination mit dem akustischen Modell die Wahrscheinlichkeit der Hypothesen zu berechnen.

→ Die Struktur der Sprache oberhalb der Wortebene als linguistisches Wissen für die Spracherkennung auszunutzen.

Sprachmodelle: Realisierung

- Sprachmodelle: Regelsysteme, die die Struktur der Sprache beschreiben
- Lernen von Sprachmodellen
- Trainingskorpus, Trainingsdaten
- Sprachmodellvorhersagen

Sprachmodelle: Realisierung

Implementierungsmöglichkeiten:

- ◆ Deterministische Grammatiken
- ◆ N-Gramm-Modelle (Markov-Modelle)
- ◆ Probabilistische kontextfreie Grammatiken

N-Gramme

Ziel: die Vorhersage der nächsten Wortform aufgrund ihrer Wahrscheinlichkeit

- Die Wahrscheinlichkeit des i -ten Wortes wird in Abhängigkeit vom Kontext aus den $i-1$ vorhergehenden Wörtern berechnet.

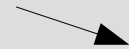
$$P(w_i | w_1, \dots, w_{i-1})$$

- $N=2$ (Bigramme): $P(w_i | w_1, \dots, w_{i-1}) = P(w_i | w_{i-1})$
- $N=3$ (Trigramme): $P(w_i | w_1, \dots, w_{i-1}) = P(w_i | w_{i-2}, w_{i-1})$

N-Gramm-Modelle trainieren !

Einfache N-Gramm-Modelle

$$P(\text{N-Gramm}) = C_{\text{N-Gramm}}/M$$



Die Gesamtzahl aller N-Gramme

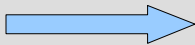
$$P(w_n | w_1, \dots, w_{n-1}) = \frac{P(w_1, \dots, w_n)}{P(w_1, \dots, w_{n-1})} = \frac{C(w_1, \dots, w_n)}{C(w_1, \dots, w_{n-1})}$$




Die Anzahl der N-Gramme

Die bedingte Wahrscheinlichkeit für w_n gegeben den vorhergehenden Kontext w_1, \dots, w_{n-1}

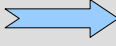
Einfache N-Gramm-Modelle

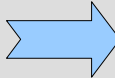
argmax berechnen  Das häufigste N-Gramm gibt die beste Vorhersage für das letzte Wort des N-Gramms, wenn die Wörter zuvor bereits bekannt sind.

Beispiel: Markov-Annahme N=3

'Heute ist es zunächst sonnig, von Süden her'  'wolkiger'

=

'Süden her'  'wolkiger'

'Heute ist es zunächst heiter, von Süden her'  'wolkiger'
Dieselbe Wahrscheinlichkeit wird verteilt.

Einfache N-Gramm-Modelle

Für jede mögliche Wortsequenz der Länge N über einem Vokabular W wird die Wahrscheinlichkeit

$$P(w_i | w_{i-N+1}, \dots, w_{i-1})$$

mit einer ausgewählten Methode geschätzt.

z.B. *Maximum-Likelihood Estimation*

- * Wahrscheinlichkeit eines N-Gramms ist gleich der relativen Häufigkeit im Trainingskorpus

$$P(w_1, \dots, w_n) = \frac{C(w_1, \dots, w_n)}{N_{n\text{-Gramme}}}$$

Maximum-Likelihood Estimation

Beispiel:

* Unser Trainingskorpus enthält insgesamt 899000 Bigramm-Token.

* Das Bigramm „die schweiz“ kommt 1009 mal vor.

$$P_{\text{MLE}}(\text{die schweiz}) = \frac{C(\text{die schweiz})}{N_{\text{Bigramme}}} = \frac{1009}{899000} = 0,001121$$

Die Anzahl der möglichen Wortsequenzen $|W|^N$
($|W|$ -die Anzahl von Wörtern im Vokabular W)

Anzahl der möglichen Wortsequenzen

→ Die Anzahl der möglichen N-Gramm-Typen in einem Markov-Modell wächst exponentiell zur Ordnung des Modells.

Vokabular $W = 20000$ Wörter und $N = 1, \dots, 5$

N	Anzahl möglicher Wortsequenzen
1	$2 * 10^4$
2	$4 * 10^8$
3	$8 * 10^{12}$
4	$16 * 10^{16}$
5	$32 * 10^{20}$

→ Beschränkung auf Bigramm- und Trigramm-Modelle und ein möglichst kleines Vokabular!

Anzahl der möglichen Wortsequenzen

Beispiel:

Trainingsdaten: *Heute ist es zunächst sonnig, von Süden her wolkiger. Morgen vor allem im Osten noch einige Schauer, von Süden her Aufheiterungen.*

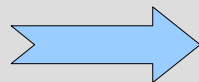
Die Vorhersage des nächsten Wortes entspricht:

* im 5-Gramm den Wörtern

Aufheiterungen ($P(w|, \text{ von Süden her})=0.5$) und
wolkiger ($P(w|, \text{ von Süden her})=0.5$),

* im 6-Gramm dem Wort *wolkiger* ($P(w|\text{sonnig, von Süden her})=1$).

➡ N möglichst gross



Datenspärlichkeitsproblem

Probleme

Prinzipiell mögliche, aber zufällig im Trainingskorpus nicht vorkommende N-Gramme



Den wohlgeformten Sätzen wird die Wahrscheinlichkeit 0 zugewiesen.

★ Auch immer größere Trainingskorpora helfen nicht gegen das Problem der „knappen“ Daten!