ETH Zürich
Dept. Informatik

Prof. Markus Gross
WS 06/07

Informationstheorie

Übung 5

Ausgabe: 27. Nov. 2006
Abgabe: 11. Dez. 2006

Inhalt dieser praktischen Übung ist es, Texte zu analysieren und als Informationsquelle zu modellieren. Das so gewonnene Modell einer Informationsquelle wird dann zum Synthetisieren von Text benutzt.

Programmiersprache und Programmierumgebung sind frei wählbar. Die Übung ist in Gruppen von 2 oder 3 Personen zu bearbeiten. Abgegeben werden müssen

- ausführbare Programme, welche die Spezifikationen unten erfüllen,
- dokumentierter Quelltext,
- und die zusätzlich in den Aufgaben geforderten Abgaben.

Die Abgabe erfolgt per Email an den jeweiligen Übungsgruppenleiter bis spätestens 11. Dezember 2006, 08:00 morgens. Bitte schicken Sie pro Übungsgruppe nur eine Email.

Es muss nur Aufgabe 1.1 gelöst werden. Aufgaben 1.2 und 1.3 sind optional.

5.1 Textanalyse, zeichenweise

Zuerst soll ein Text analysiert werden und ein Markov-Modell erstellt werden. Dazu wird eine Textdatei als Informationsquelle betrachtet. Der Einfachheit halber sollen Satzzeichen sowie Gross- und Kleinschreibung ignoriert werden.

Eine grosse Auswahl an Texten zum Testen findet sich auf den Webseiten des Gutenberg-Projekts: www.gutenberg.org

Alle Programme in dieser Aufgabe sollen mindestens zwei Parameter entgegennehmen. Der erste Parameter bezeichnet eine Eingabedatei, die den zu analysierenden Text beinhaltet. Der zweite Parameter bezeichnet eine Ausgabedatei, in die die Ergebnisse der Analyse geschrieben werden. Die Ausgabe soll in einer für Menschen lesbaren Form (nicht binär) sein.

Analysieren Sie in jeder Teilaufgabe (mindestens) den Text "The Autobiography of Charles Darwin". Der Text ist auf der Vorlesungshomepage verfügbar.

a) Markov-Quelle 0. Ordnung, zeichenweise

Der Text soll zunächst als Markov-Quelle 0. Ordnung modelliert werden. Das Alphabet dieser Quelle ist die Menge aller Grossbuchstaben und das Leerzeichen: {*A*–Z, `<space>`}. Interpretieren Sie mehrere aufeinanderfolgende Leerzeichen als nur ein Leerzeichen.

Lesen Sie die Eingabedatei ein und berechnen Sie wärend des Einlesens die Wahrscheinlichkeiten für das Auftreten der Elemente des Alphabets.

Schreiben Sie diese Wahrscheinlichkeiten in die Ausgabedatei.

Analysieren Sie mit Ihrem Programm den Text "The Autobiography of Charles Darwin". Die Wahrscheinlichkeitstabelle für diesen Text (die Ausgabedatei) soll mit abgegeben werden.

Stimmen die berechneten Wahrscheinlichkeiten mit den in der Vorlesung vorgestellten überein? Wie sind Abweichungen zu erklären?

Tip: Mit einer geschickten Implementierung ist Teilaufgabe a) nur ein Spezialfall von Teilaufgabe c) und muss nicht getrennt implementiert werden.

b) Entropieberechnung

Berechnen Sie mittels der Wahrscheinlichkeiten aus Teil a) die Entropie des analysierten Texts. Erweitern Sie Ihr Programm so, dass es die Entropie des Texts auf dem Bildschirm ausgibt, sobald die Analyse beendet wurde. Wie gross ist die Entropie von Darwin's Autobiographie? Generieren sie mittels eines gleichverteilten Pseudo-Zufallsgenerators einen zufälligen Text (aus dem Alphabet {A–Z, `<space>`}). Wie gross ist die Entropie des Zufallstextes? Vergleichen sie mit dem theoretischen Maximum.

c) Markov-Quellen höherer Ordnung, zeichenweise

Wie in Abschnitt a) soll wieder eine Datei eingelesen werden. Der enthaltene Text wird nun mit einer Markov-Quelle höherer Ordnung modelliert. Die Ordnung der Markov-Quelle wird als Parameter übergeben. Berechnen Sie während des Einlesens die bedingten Wahrscheinlichkeiten der Quelle. Speichern Sie diese Wahrscheinlichkeiten in der Ausgabedatei.

Testen Sie die Implementation bis mindestens zur 3. Ordnung. Welche Probleme treten bei höheren Ordnungen auf? Wie kann man ihnen beikommen?

Analysieren Sie mit Ihrem Programm den Text "The Autobiography of Charles Darwin", 1., 2., und 3. Ordnung. Die Wahrscheinlichkeitstabellen (ohne Einträge für Ereignisse mit Wahrscheinlichkeit null) sollen mit abgegeben werden.

Tip: Mit einer geschickten Implementierung ist Aufgabe 1.1 nur ein Spezialfall von Aufgabe 1.2 und muss nicht getrennt implementiert werden.

5.2 Textanalyse, wortweise (fakultativ)

Der Text soll wieder als eine Markov-Quelle modelliert werden. Das Alphabet dieser Quelle besteht diesmal allerdings nicht aus einzelnen Buchstaben, sondern aus ganzen Wörtern, d.h. in unserem Fall aus der Menge aller englischen Wörter. Damit ist die Anzahl der Elemente des Alphabets der Quelle unbekannt. Weiterhin ist die Anzahl der Elemente sehr hoch, so dass einige Verfahren, die in Aufgabe 1.1 zum Erfolg führen können, hier versagen.

Die Quelle soll als Markov-Quelle n-ter Ordnung modelliert werden. Die Ordnung soll als Parameter eingelesen werden. Lesen Sie die Eingabedatei ein und berechnen Sie während des Einlesens die (bedingten) Übergangswahrscheinlichkeiten. Testen Sie Ihr Programm, bis zu welcher Ordnung funktioniert Ihre Implementation?

5.3 Textsynthese (fakultativ)

In dieser Aufgabe sollen die in Aufgabe 1.1 und 1.2 berechneten Wahrscheinlichkeiten benutzt werden, um Text zu synthetisieren. Schreiben Sie ein Programm, welches die Wahrscheinlich-

keiten aus einer Datei, wie Sie Ihre Programme aus den Aufgaben 1.1 bzw. 1.2 erstellen, liest, und eine Markov-Quelle der entsprechenden Ordnung simuliert. Das Programm sollte mindestens einen Parameter entgegennehmen, der die Eingabedatei bezeichnet (welche die Wahrscheinlichkeiten enthält).

Es ist hier hilfreich, die Markov-Quelle wie in der Vorlesung als eine Quelle mit einem Gedächtnis zu betrachten. Für eine Quelle n-ter Ordnung "erinnert" sich die Quelle an die zuletzt ausgegebenen n Symbole (Zeichen oder Worte). Diese n Symbole bestimmen, welche bedingten Wahrscheinlichkeiten zur Auswahl des nächsten Symbols verwendet werden.

Ein Zufallsgenerator trifft die Auswahl aus den möglichen Symbolen. Ein gleichverteilter Zufallsgenerator ist in der Standardbibliothek der meisten Programmiersprachen enthalten. Überlegen Sie sich, wie Sie aus einer Gleichverteilung eine beliebige diskrete Verteilung erzeugen können (die Sie zur Auswahl der Symbole benötigen).

Analysieren Sie einen Text, und generieren Sie aus dessen Übergangswahrscheinlichkeiten einen neuen Text. Wie hängen Qualität der Ausgabe und Ordnung der Quelle zusammen? Wie hängen Länge des Eingabetexts und Qualität der Ausgabe zusammen?

Können Sie anhand einen generierten Texts den Autor/die Epoche des Originaltexts raten?

Tip: Wenn die Programme aus Aufgabe 1.1 und 1.2 dasselbe Dateiformat schreiben, brauchen Sie auch nur ein Dateiformat zu lesen.