4. Übung zur Vorlesung "Internet-Suchmaschinen" im Wintersemester 2007/2008
– mit Musterlösungen –

Dr. Andreas Hotho, Prof. Dr. Gerd Stumme, M.Sc. Wi.-Inf. Beate Krause

13. Dezember 2007

1 Levenstein-Metrik

Berechnen Sie die Levenstein-Metrik *lev*("carsten","christina").
Geben Sie das Berechnungsschema an, und nennen Sie die einzelnen Umformungsschritte einer kürzesten Umformung in der Art: test → tost → toast

		C	H	R	I	S	T	I	N	A
	0	1	2	3	4	5	6	7	8	9
C	1	**0**	1	2	3	4	5	6	7	8
A	2	1	**1**	2	3	4	5	6	7	7
R	3	2	2	**1**	**2**	3	4	5	6	7
S	4	3	3	2	2	**2**	3	4	5	6
T	5	4	4	3	3	3	**2**	3	4	5
E	6	5	5	4	4	4	3	**3**	4	5
N	7	6	6	5	5	5	4	4	**3**	**4**

Eine kürzeste Umformung wäre also
carsten → chrsten → christen → christin → christina

2 Soundex

1. Vergleichen Sie die folgenden Wörter mit Soundex!
   - through, thru, trough T62, T6, T62

- Mr, Mayer, Meier M6, M6, M6
- Smith, Schmidt, Schmitz S53, S253, S253
- data, date, dito D3, D3, D3

2. Sehen Sie Probleme bei der Verwendung des Algorithmus'?

   - Soundex macht öfters den Fehler, nicht phonetisch ähnliche Worte auf den gleichen Code zu reduzieren (etwa bei Mr/Mayer), sowie ähnliche Worte auf unterschiedliche Codes (through, thru) abzubilden.
   - Einige der Regeln sind sprachabhängig. So ergibt es im Englischen einen Sinn, *h* und *y* wie Vokale zu behandeln, im Deutschen weniger.
   - Auch macht es nicht immer Sinn, den ersten Buchstaben zu behalten. Kamper und Camper werden dadurch auf unterschiedliche Codes abgebildet.
   - Gleiche Codes können nicht gerankt werden.

3. Schlagen Sie vor, wie Sie die Probleme in Soundex beheben können.

   - Eine größere Menge von Regeln und Ausnahmen helfen, solche sprachlichen Feinheiten zu berücksichtigen. Phonix z. B. beinhaltet über 100 Regeln, die Fälle wie *ough* $\rightarrow$ *ow, kn* $\rightarrow$ *n, chr* $\rightarrow$ *kr* abdecken (Gadd T., PHONIX: The Algorithm, Program, 24(4), p381-402, 1990).
   - "Combination of evidence": Eine Kombination aus Editierdistanzen und phonetischen Distanzen kann die Wahrscheinlichkeit, dass zwei Namen ähnlich sind, verbessern. (z.B. Editex in Justin Zobel, Philip Dart, Phonetic String Matching: Lessons from Information Retrieval, SIGIR, 1996)

4. Welche neuen Nachteile bringen Ihre Verbesserungen mit sich?

   Neben der größeren Komplexität erhöht solch ein umfangreiches Regelwerk vor allem die Sprachabhängigkeit. Phonix in der vorliegenden Form ist z. B. nur für die englische Sprache zu gebrauchen.

3 XML/XPath

1. Betrachten Sie das XML-Dokument auf Seite 27 des Kapitels "Strukturelle Anfragen" (`<library ... >`).

   Geben Sie XPath-Ausdrücke an, die folgende Teile des Dokuments auswählen:

   a) alle Bücher
   /library/author/book
   //book

b) alle Bücher von William Smart
/library/author[@name="William Smart"]/book
//author[@name="William Smart"]/book

2. Funktionieren die folgenden Ausdrücke auf dem Dokument auf Seite 19 genau so?

a) alle Personen
//Person geht, nicht aber /db/Person , da hier Personen an verschiedenen Stellen im Baum vorkommen.

b) alle Personen, die Robert heißen. *Tip:* Den Textinhalt von Kindelementen kann man prüfen, indem man einen relativen Pfad statt `@attribut` in die eckige Klammer schreibt!
//Person[Name/Vorname = 'Robert']

4 Eigenschaften von Texten/Power Laws

1. Begründen Sie die Aussage von Luhn auf Seite 6 des Kapitels: Warum sind besonders häufige und besonders seltene Wörter nicht sehr nützlich?

Luhn hat den charakteristischen Kurvenverlauf, den Zipf für die Häufigkeit von Wörtern herausgefunden hat, weiter untersucht. Zipf hatte gezeigt, dass die Häufigkeit von Wortvorkommen (n) multipliziert mit dem Rang des Wortvorkommens (r) in etwa konstant bleiben: $n * r = k$. Diese Funktion spiegelt das Potenzgesetz wider: $n = k * r^{(-1)}$. Der Exponent der Wortverteilungen (oder die Steigung im log-log skalierten Diagramm) ist also -1.

Luhn beschäftigte sich damit, wie oft ein Wort in einem Dokument vorkommen sollte, um ein gutes Schlüsselwort zu sein. Seine These war, dass ein Wort, welches zu oft vorkommt, eben so wenig aussagekräftig ist wie eines, das zu selten vorkommt. Er legte eine Kurve über Zipf's Häufigkeitsverteilung, mit der die Bedeutung eines Wortes als Schlüsselwort dargestellt werden sollte.

Besonders häufige Wörter (Stoppwörter) kommen in fast jedem Dokument vor und sind somit nicht geeignet, relevante von nicht relevanten Dokumenten zu unterscheiden.

Besonders seltene Wörter sind unter Umständen so selten, dass sie auch praktisch nie angefragt werden. Dadurch ergeben sie ebenfalls keinen Nutzen beim Information Retrieval, auch wenn sie für sehr spezifische Anfragen perfekte Resultate liefern können (100% Precision, 100% Recall).

Als charakteristisch ("resolving power") für einen Text können dagegen solche Wörter gelten, die informativ und zugleich redundant genug sind, um die Sätze herauszufiltern, die für diesen Text typisch sind.

2. Auch die Grade von Webseiten sind nach einem Potenzgesetz (*power law*) verteilt. Die folgende Tabelle gibt einige In-grade einer Menge von Webseiten zu einem Zeitpunkt im Jahr 1999 wieder:

Grad	Anzahl Seiten
1	63100000
10	501200
100	4000
1000	32
5000	1

Bestimmen Sie den Exponenten c im Potenzgesetz!

Grad	Anzahl Seiten	Log Grad	Log Anz	Diff
1	63100000	0	7,8	
10	501200	1	5,7	-2,1
100	4000	2	3,6	-2,1
1000	32	3	1,51	-2,1
5000	1	3,7	0	-2,15

Die Steigung im Log-Log-Plot – also der Exponent c im Potenzgesetz – ist hier etwa -2.1.

5 Praxisübung; Abgabe am 09.01.2008

Implementieren Sie die Phrasensuche wie auf Übungsblatt 3, Aufgabe 1, skizziert. Wenn nötig, erweitern Sie die entsprechenden Datenstrukturen.