TECHNISCHE UNIVERSITÄT DRESDEN

Dresden Database Systems Group

Exercise 2

Information Retrieval

3. Term Vocabulary and Normalization

General Understanding

Exercise 3.1

- Are the following statements true or false? Give reasons for your answer.

a) The tokenization of a document is a trivial task. t

b) In a Boolean retrieval system, stemming never lowers precision. f

c) In a Boolean retrieval system, stemming never lowers recall. f increased or unchanged recall

d) Stemming increases the size of the vocabulary. f decreases the size of the vocabulary.

e) Stemming should be applied to the documents, but not to the queries. f The same
processing should be applied

f) The postings list of a stop word is usually longer than the postings list of a non-stop word.

?- T

Tokens and Terms

Exercise 3.2

不变的

- How many tokens and terms do the following documents contain with and without normalization (make an educated guess at the results of the normalization)?

10 8
10 7

  - the black cat jumps over the other two black cats ok

  - the reader is reading the most informative book on information retrieval

11 10
11 8

see 照片

Porter Algorithm

Remember…

- Based on a set of context-sensitive rewriting rules
- The measure *m* of a stem is based on its alternate vowel-consonant sequences $[C](VC)^m[V]$
- Five phases of reduction rules, applied sequentially
- Within a phase, in the case of ambiguity, the longest suffix match is preferred

Exercise 3.3

a) Find the stems of the following words using the given rules (which are sufficient for this task*) ok

- `organizations`
- `organizer`
- `organ`
- `realness`
- `relativity`

b) What class of stemmers does the Porter Algorithm belong to, and what other approaches to stemming exist?

ppt? ?? ok

Condition on stem	Condition on suffix	Replacement
Step 1a		
	SS	SS
	S	
Step 1b (not required in this example)		
Step 1c		
(*V*)	Y	I
Step 2		
(m>0)	IZATION	IZE
(m>0)	IVITI	IVE
(m>0)	IZER	IZE
Step 3		
(m>0)	ATIVE	
(m>0)	NESS	
Step 4		
(m>1)	IZE	
(m>1)	AL	
Step 5 (not required in this example)		

* find the whole algorithm and set of rules at http://snowball.tartarus.org/algorithms/porter/stemmer.html

4. Dictionaries and Tolerant Retrieval

Permuterm Index for Wildcard Queries

Exercise 4.1

a) Which entries does the term `test` generate in the *permuterm* index? ok

b) Draw the *permuterm* index for for the terms `test`, `toast`, and `west`. ok

c) How can the following wildcard queries be answered using the *permuterm* index and/or the inverted index?

Q1	`t*`
Q2	`t*st`

Bigram Index for Wildcard Queries

Exercise 4.2

a) Which entries does the term `test` generate in the *bigram* index? ok

b) Draw the *bigram* index for for the terms `test`, `toast`, and `testament`. ok

c) How can the following wildcard queries be answered using the *bigram* index and/or the inverted index?

Q1	`t*`

Q2	`t*st`

Levenshtein Distance

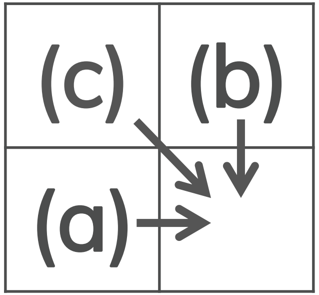
Exercise 4.3

a) Compute the Levenshtein distance matrix for the words `OSLO` and `SNOW` ok

b) What are the respective Levenshtein editing operations? ok

Remember…

- Coming from (a)
  - add 1 to cost in (a) = **insertion**
- Coming from (b)
  - add 1 to cost in (b) = **deletion**
- Coming from (c)
  - If characters in row and column are equal, **copy** costs from (c)
  - If they are not equal, add 1 to cost in (c) = **replacement**
- Take the **minimum** of the costs



Levenshtein Distance

	“”	S	N	O	W
“”	0	1 1	2 2	3 3	4 4
O	1 1				
S	2 2				
L	3 3				
O	4 4				

ok

Soundex

Exercise 4.4

- Compute the Soundex codes of the words `SMITH`, `MILLER`, and `MUEHLHERR` ok
- Find two phonetically similar proper nouns whose Soundex codes are different ok

Remember…

1. Retain the first letter of the term
2. Change all occurrences of the following letters to ’0’ (zero): A, E, I, O, U, H, W, Y
3. Change letters to digits as follows
   - B, F, P, V to 1
   - C, G, J, K, Q, S, X, Z to 2
   - D, T to 3
   - L to 4
   - M, N to 5
   - R to 6
4. Repeatedly remove one out of each pair of consecutive identical digits
5. Remove all zeros from the resulting string; pad the resulting string with trailing zeros and return the first four positions, which will consist of a letter followed by three digits