Exercise 2.1

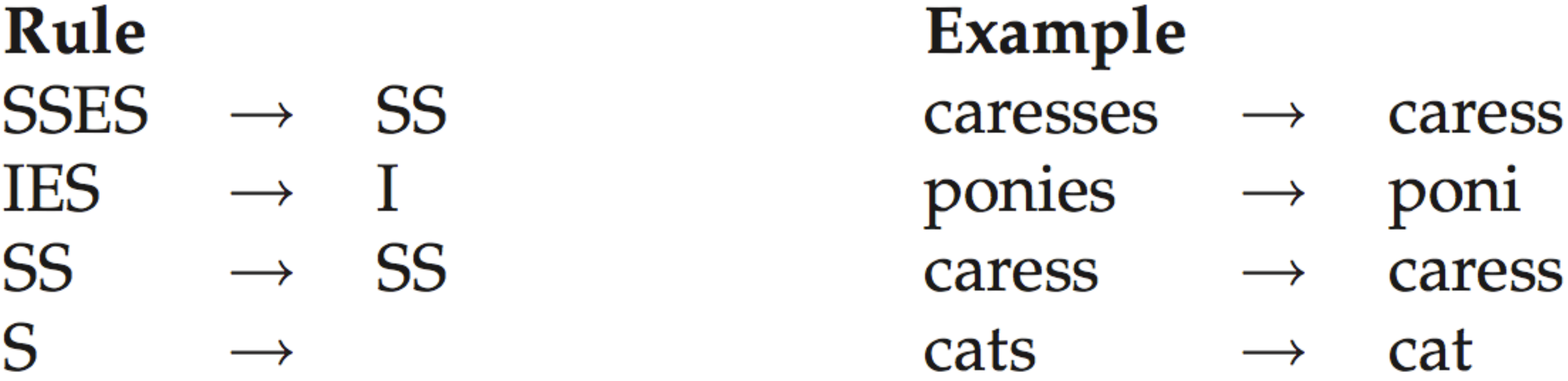
- Are the following statements true or false?
  - a. In a Boolean retrieval system, stemming never lowers precision.
  - b. In a Boolean retrieval system, stemming never lowers recall.
  - c. Stemming increases the size of the vocabulary.
  - d. Stemming should be invoked at indexing time but not while processing a query.

Exercise 2.3

- The following pairs of words are stemmed to the same form by the Porter stemmer.
- Which pairs would you argue shouldn’t be conflated. Give your reasoning.
  - a. abandon/abandonment
  - b. absorbency/absorbent
  - c. marketing/markets
  - d. university/universe
  - e. volume/volumes

Exercise 2.4

- Read section 2.2.2 in IIR – about stemming
- For the Porter stemmer rule group shown in formula (2.1) in the book:
  - a. What is the purpose of including an identity rule such as SS →SS?



Rule			Example		
SSES	→	SS	caresses	→	caress
IES	→	I	ponies	→	poni
SS	→	SS	caress	→	caress
S	→		cats	→	cat

Exercise 2.4 cont.

- b. Applying just this rule group, what will the following words be stemmed to?
  - circus canaries boss
- c. What rule should be added to correctly stem pony?
- d. The stemming for ponies and pony might seem strange. Does it have a deleterious effect on retrieval? Why or why not?

Exercise 2.8

- Assume a biword index. Give an example of a document which will be returned for a query of 'New York University' but is actually a false positive which should not be returned.

Exercise 2.9

- Shown below is a portion of a positional index in the format: **term**: **doc1**:[position1, position2, . . .]; **doc2**:[position1, position2, . . .]; etc.

  angels: **2**:[36,174,252,651]; **4**:[12,22,102,432]; **7**:[17];
  fools: **2**:[1,17,74,222]; **4**:[8,78,108,458]; **7**:[3,13,23,193];
  fear: **2**:[87,704,722,901]; **4**:[13,43,113,433]; **7**:[18,328,528];
  in: **2**:[3,37,76,444,851]; **4**:[10,20,110,470,500]; **7**:[5,15,25,195];
  rush: **2**:[2,66,194,321,702]; **4**:[9,69,149,429,569]; **7**: [4,14,404];
  to: **2**:[47,86,234,999]; **4**:[14,24,774,944]; **7**:[199,319,599,709];
  tread: **2**:[57,94,333]; **4**:[15,35,155]; **7**:[20,320];
  where: **2**:[67,124,393,1001]; **4**:[11,41,101,421,431]; **7**[16,36,736];

- Which document(s) if any match each of the following queries, where each expression within quotes is a phrase query?
  - a. “fools rush in”
  - b. “fools rush in” AND “angels fear to tread”

Exercise 2.10

- Consider the following fragment of a positional index with the format:
- word: document: [position, position, . . .]; document: [position, . . .] . . .

  Gates: 1: [3]; 2: [6]; 3: [2,17]; 4: [1];

  IBM: 4: [3]; 7: [14];

  Microsoft: 1: [1]; 2: [1,21]; 3: [3]; 5: [16,22,51];
- The /*k* operator, word1 /*k* word2 finds occurrences of word1 within *k* words of word2 (on either side), where *k* is a positive integer argument. Thus *k* = 1 demands that word1 be adjacent to word2.
  - a. Describe the set of documents that satisfy the query Gates /2 Microsoft.
  - b. Describe each set of values for *k* for which the query Gates /*k* Microsoft returns a different set of documents as the answer.

Exercise 2.11

- Consider the general procedure for merging two positional postings lists for a given document, to determine the document positions where a document satisfies a /*k* clause (in general there can be multiple positions at which each term occurs in a single document).
- We begin with a pointer to the position of occurrence of each term and move each pointer along the list of occurrences in the document, checking as we do so whether we have a hit for /*k*. Each move of either pointer counts as a step. Let *L* denote the total number of occurrences of the two terms in the document.
- What is the big-O complexity of the merge procedure, if we wish to have postings including positions in the result?

Exercise 2.11 cont.

- The simple idea of checking the /k condition by moving each pointer along the list of occurrences in the document, checking as we do so whether we have a hit for /k, will miss some solutions
- Try with this example: pp1 = <1,5,8>, pp2 = <2,3,4>, and k=2 - (pp1 and pp2 are the positions of the two terms in a document)
- Find what hits you miss if you move:
  - a) first the pointer to the smallest position
  - b) first the pointer with the smallest next position
- c) Run in this example the Positional Intersect algorithm presented in part 03 of the slides (or fig. 2.12 of IIR) and find the right solution.

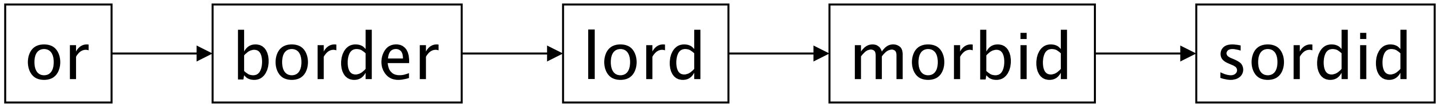
Exercise 3.2, 3.3, 3.5

- Write down the entries in the permuterm index dictionary that are generated by the term mama.
- If you wanted to search for s*ng in a permuterm wildcard index, what key(s) would one do the lookup on?
- Consider again the query fi*mo*er from Section 3.2.1. What Boolean query on a bigram index would be generated for this query? Can you think of a term that matches the permuterm query in Section 3.2.1 – and it is a false positive - but does not satisfy this Boolean query?

Exercise 3.7, 3.10

- If |*si*| denotes the length of string *si*, show that the edit distance between *s*1 and *s*2 is never more than max{|*s*1|, |*s*2|}.

- Compute the Jaccard coefficients between the query **bord** and each of the terms that contain the bigram 'or', as below:



or → border → lord → morbid → sordid

- Based on that computation what is the spell corrected word of **bord?**