

Name: Laura Melton

## Assignment 1: Part-1: Tokenization and Stemming

LIS-544: Information Retrieval Systems

For lab on Friday 1/23/2003; **DUE 1/30/04**

Value: 15% of grade

### Objectives:

Become familiar with:

- the process of document analysis for text search, emphasizing the following stages: parsing, indexing, matching
- the statistical properties of terms in a document collection.
- the operation of weighting algorithms, such as *tf*, *idf*, *tf\*idf*, *OKAPI* ☺
- the way weighting algorithms influence the ranking of retrieved documents

For an in depth treatment of these topics please read Belew chapters 2-3.

### Exercise 1: Term lists

“There was nothing so VERY remarkable in that; nor did Alice think it so VERY much out of the way to hear the Rabbit say to itself, `Oh dear! Oh dear! I shall be late!’ (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural); but when the Rabbit actually TOOK A WATCH OUT OF ITS WAISTCOAT-POCKET, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket, or a watch to take out of it, and burning with curiosity, she ran across the field after it, and fortunately was just in time to see it pop down a large rabbit-hole under the hedge.”

**Directions:** Parse the above quotation once for each of the following sets of rules. Write the list of terms in the box below each rule set. *If you run out of room to write the entire list in the box, note that somewhere and move on to the next rule set.*

#### Rule Set 1

- A term is defined as any contiguous set of non-whitespace characters.
- Non-whitespace characters include letters, numbers, and punctuation.

|             |       |          |        |        |            |         |         |       |      |
|-------------|-------|----------|--------|--------|------------|---------|---------|-------|------|
| “There      | was   | nothing  | so     | VERY   | remarkable | in      | that;   | nor   | did  |
| Alice       | think | it       | so     | VERY   | much       | out     | of      | the   | way  |
| to          | hear  | the      | Rabbit | say    | to         | itself, | `Oh     | dear! | Oh   |
| dear!       | I     | shall    | be     | late!’ | (when      | she     | thought | it    | over |
| afterwards, | it    | occurred | to     | her    | that       | she     | ought   | to    | have |
| wondered    | at    | this,    | but    | at     | the        | time    | it      | all   |      |

[no room for the rest]

## Rule Set 2

- All text is folded into lower case.<sup>1</sup>
- A term is defined as any contiguous set of letters and numbers.
- Any character that is not a letter or number is considered a term delimiter.

|            |             |           |        |           |            |        |           |       |      |
|------------|-------------|-----------|--------|-----------|------------|--------|-----------|-------|------|
| there      | was         | nothing   | so     | very      | remarkable | in     | that      | nor   | did  |
| alice      | think       | it        | so     | very      | much       | out    | of        | the   | way  |
| to         | hear        | the       | rabbit | say       | to         | itself | oh        | dear  | oh   |
| dear       | i           | shall     | be     | late      | when       | she    | thought   | it    | over |
| afterwards | it          | occurred  | to     | her       | that       | she    | ought     | to    | have |
| wondered   | at          | this      | but    | at        | the        | time   | it        | all   |      |
| seemed     | quite       | natural   | but    | when      | the        | rabbit | actually  | took  | a    |
| watch      | out         | of        | its    | waistcoat | pocket     | and    | looked    | at    | it   |
| and        | then        | hurried   | on     | alice     | started    | to     | her       | feet  | for  |
| it         | flashed     | across    | her    | mind      | that       | she    | had       | never |      |
| before     | seen        | a         | rabbit | with      | either     | a      | waistcoat |       |      |
| pocket     | or          | a         | watch  | to        | take       | out    | of        | it    | and  |
| burning    | with        | curiosity | she    | ran       | across     | the    | field     | after | it   |
| and        | fortunately |           | was    | just      | in         | time   | to        | see   | it   |
| pop        | down        | a         | large  | rabbit    | hole       | under  | the       | hedge |      |

## Rule Set 3

- All rules in Rule Set 2 apply.
- Any terms in this stop list should be ignored: a, is, was, it, but, had, or, in, the, of, and.<sup>2</sup>
- the following stemming rules should be applied:

- drop trailing:    -ing   -ed                   ○ change plurals to singular  
                          -ity   -able

|         |         |        |            |           |        |       |        |             |        |
|---------|---------|--------|------------|-----------|--------|-------|--------|-------------|--------|
| there   | noth    | so     | very       | remark    | that   | nor   | did    | alice       | think  |
| so      | very    | much   | out        | way       | to     | hear  | rabbit | say         | to     |
| itself  | oh      | dear   | oh         | dear      | i      | shall | be     | late        | when   |
| she     | thought | over   | afterwards | occurr    | to     | her   | that   | she         | ought  |
| to      | have    | wonder | at         | this      | at     | time  | all    | seem        | quite  |
| natural | when    | rabbit | actually   | took      | watch  | out   | its    | waistcoat   |        |
| pocket  | look    | at     | then       | hurri     | on     | alice | start  | to          | her    |
| feet    | for     | flash  | across     | her       | mind   | that  | she    | never       | before |
| seen    | rabbit  | with   | either     | waistcoat | pocket | watch | to     | take        | out    |
| burning | with    | curios | she        | ran       | across | field | after  | fortunately |        |
| just    | time    | to     | see        | pop       | down   | large | rabbit | hole        | under  |
| hedge   |         |        |            |           |        |       |        |             |        |

<sup>1</sup> See page 44 of Belew for a brief explanation

<sup>2</sup> See page 47 of Belew for a discussion of noise words. Noise words are called stop words when they are filtered out of a stream of terms in a document.

## Exercise 2: Use IR toolbox to examine term lists

In this exercise you will have an opportunity to get accustomed to the IR toolbox. You will create two different types of document analyzers (equivalent to the rule sets in the first exercise). the first analyzer will be very simple. the second will incorporate a greater number of rules to parse the documents. You will then compare the term lists.

**Directions:** Login to the IR toolbox at <http://ir.ischool.washington.edu>. For security's sake, the username is 'ischool' and the password is 'ir=fun'. this is the class-wide login. At the IR toolbox login screen, type in your UWNNetID and click Login to create an account for yourself. Always use your UWNNetID when login in the IR toolbox to be able to review your prior work. Follow each of the steps below to build an index.

1. Click 'Create a new index'.
2. type in a name for the new index. Index names can contain only word characters (letters, digits, and the underscore). No spaces are allowed.
3. Select the data source: LA010189.
4. Click 'Assign'.
5. On the Define Options page, change the tokenizing option to whitespace. Leave all other options at their defaults.
6. On the Defining Stop Words page, leave as is and click 'Assign'.
7. On the Define Document Fields page, type:
  - a. HEADLINE in the title field, *[you need to type the field names in UPPERCASE because the parser is case sensitive]*
  - b. tEXt in the Main Body field, and
  - c. DOCID in the Document Number field.
8. Click 'Assign' and review your indexing setup. type a short description of the index you are about to create.
9. Click 'Build' when you are ready to build the index. **DO NOT CLICK build MORE tHAN ONCE!**
10. After a **short wait**, the screen should update and you should see your index listed in the table something like below. Congratulations.

| <a href="#">Searching</a> |               | <a href="#">Indexing</a> |                       | <a href="#">Login</a> |                                       |
|---------------------------|---------------|--------------------------|-----------------------|-----------------------|---------------------------------------|
| Name                      | Source Data   | Analysis Example         | View Term Frequencies | Description           | Delete?                               |
| testLA                    | LA010189.sgml | <a href="#">Example</a>  | <a href="#">Terms</a> |                       | <input type="checkbox"/>              |
|                           |               |                          |                       |                       | <input type="button" value="Delete"/> |

11. Repeat the above steps but use these options where applicable:
  - a. Convert(fold) all text to lower case,
  - b. Use the 'Standard rule-based tokenizer',
  - c. Apply the Porter Stemmer, and
  - d. Choose a stop word list to apply.

**For each of your new document databases**, click on the ‘Example’ link. this link shows how the text from “Alice in Wonderland” has been processed based on the rules you have chosen. Do you see any differences between the original text and the processed (or analyzed) text? Why or why not? What is the nature of the differences? In what ways will these differences help our IR system? *to assist you in the discussion “cut-n-paste” the processed texts here, then comment.*

**Simple:**

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, `and what is the use of a book,' thought Alice `without pictures or conversation?'

**Complex:**

alic wa begin get veri tire sit her sister on bank have noth do onc or twice she had peep into book her sister wa read but it had no pictur or convers in it what is us a book thought alic without pictur or convers

There isn't any difference at all between the original text and the simple analysis. On the other hand, the complex analysis makes the text barely recognizable.

The omission of stop words will help decrease the size of the data and reduce noise. Word stemming will increase recall, because other forms of the same word will be retrieved.

Term frequencies are an important characteristic of a document corpus. For each of the new LA times databases you have indexed, first click “explore” on the “explore index” column, then right-click on the ‘term frequencies’ link and ‘Save target as...’ (Only in Internet Explorer). Save the ‘termFreq.jsp’ file as \*.txt where \* is some unique name. (Choose “Save as type” – All files, and affix the .txt extension at the end of the filename.)

Using SPSS, Excel, Access or some other data analysis application you are comfortable with, import the data you just saved in the application, then process and answer the following questions:

1. Which term occurs in the greatest number of documents for each dataset?

| Simple, whitespace analysis | Complex, rule-based analysis |
|-----------------------------|------------------------------|
| the (183)                   | a (178)                      |

2. Which term has the highest overall frequency for each dataset?

| Simple, whitespace analysis | Complex, rule-based analysis |
|-----------------------------|------------------------------|
| the (5829)                  | a (2649)                     |

3. Note the number of terms listed for each dataset. Briefly, how is the number of terms significant in the implementation of an IR system? How will using one list of terms over another change performance?

Simple: 24086; Complex: 21626

The size of the list of terms for the complex analysis is just over 10% less than the other, which can be a significant decrease in searching time. Also, a lot of “meaningless” words are eliminated; those words could conceivably lead to false hits.

### **Exercise 3: Searching over a processed index**

You will now have an opportunity to see just how different indices (or term-document databases) impact the searching process.

1. Click the 'Searching' link.
2. Do the following queries. For each query, note (1) the number of retrieved documents out of the total number of documents and (2) the highest ranked document.<sup>3</sup>
  - a. 'county' on the simple, whitespaced index
  - b. 'county' on the complex, rule-based index
  - c. 'County' on the simple, whitespaced index
  - d. 'County' on the complex, rule-based index

In the box below, discuss why and how these queries differ or do not differ in their results. Why is the number of documents retrieved different? Why does the highest ranked document change? In what fashion does the capitalization of 'county' impact the result document hits?

- a. 10, "Trying time for new courthouse"
- b. 53, "1988 the year in review"
- c. 48, "1988 the year in review"
- d. 53, "1988 the year in review"

The capitalization of "county" affects hits using the simple index because case information is stored in the database. Evidently more records have the word "county" capitalized than lower-case, so the two queries return a different number of results.

In contrast, queries passed to the simple index go through the same capitalization-folding and stemming that the index does, so capitalization doesn't affect the number of hits.

The highest-ranked document changes because the word "county" doesn't appear as many times as the capitalized version. That doesn't matter with the complex index, but it does with the simple index, so the document that floats to the top with most of the queries gets shuffled to the bottom with the lower-case query and simple index. Maybe the word "county" appears once and "County" appears five times, but those aren't counted with this index.

<sup>3</sup> The search results page has a line at the top like this: Search retrieved XX out of YYY.

## **Exercise 4: Document Discrimination**

In this exercise, you will work with the 'Explore' page linked from the 'Indexing' page of the IR toolbox. Among a few other things, the web page enables the user to produce term weights lists and analyze the term weighting algorithms. You need only use Excel for this exercise.

### **Directions:**

1. Select a database to use (or create a new one if you so desire).
2. Click the 'Explore' link.
3. Run each of the weighting algorithms by searching using a single term (IDF, tF-IDF, OKAPI).
4. Save each set of data to file.
5. Open each file in Excel and plot the weight distributions in whichever you feel is most appropriate.

Given the plots that you have created, can you determine which algorithm is the most successful at discriminating among documents? Explain below.

IDF clearly does not discriminate at all; all the weights for the one term are exactly the same for all the documents.

TF-IDF is a little better, because it has a few variations; however, there were only seven possible values, with many documents sharing each value.

AKAPI was the best, since each document had a unique term weight.

In reference to the question below, "county" was the term that I used, on the complex index; it was not the top-ranked term by AKAPI or TF-IDF. It was actually ranked fairly low by AKAPI.

try the weighting algorithms again but this time use multiple terms. From the data, can you determine which document might be ranked the highest if this had been a traditional search query? Note the document number. Go to the 'Searching' page and do a search using the same terms. Was the document you noted ranked highly