CS1321 Homework 3 FAQ*

Jeffrey D. Oldham

2000 Feb 11

Contents

1 2000 Feb 11

1.1 Words with Three or Fewer Characters Not in Database

An undocumented feature of the database preparation program is that words with three or fewer characters are ignored. Also, the first occurrence of a punctuation mark marks the end of a word. For example, the apostrophe in "don't" causes the database preparation program to consider only "don," which is too short to be included in the database.

To summarize, search only for words with four or more characters.

1.2 Ending Spaces on Database Lines Important

Each line in the database file ends with a space. Without this space, the distributed code is not guaranteed to work correctly.

2 2000 Feb 09

2.1 Comparator Function for `sort`

The second form for sort requires a third parameter to compare elements in the container that are being sorted. The parameter can be a function that acts like less-than $<$. That is, the function should take two arguments and return a `bool` indicating whether the first argument is strictly less than the second argument. Using such a function with `sort` will cause the container to be sorted from smallest to largest. For an example, see the program sorting in a case-insensitive way.

3 2000 Feb 08

3.1 Bug Report and Fix

Reported by Daniel White that some vector components of the distributed database files had zero for all documents.

Problem Solution (JDO): Copy newly created databases: hello-goodbye.db, etext90.db, etext90.db and, if desired, prepareDatabase.cc.

Problem Description (JDO): During the two phases of database preparation, words were canonicalized during the "word collection phase" but not during the phase when documents' vectors were created.

3.2 Math Vectors and C++ Vectors

C++ vectors were named after mathematical vectors, but it becomes confusing.

C++ vector	mathematical vector
like an array	a sequence of numbers
length means the number of components	length is Euclidean length

The Euclidean length of a vector $(3, 4, 0, 5, 6, 0)$ is $\sqrt{3*3+4*4+0*0+5*5+6*6+0*0}$.

In this program, we use C++ vectors to store mathematical vectors.

3.3 How Do We Use the Mathematical Vectors?

The short answer is

Every document is represented as a mathematical vector.

Each vector has exactly the same length. To convert a document into a mathematical vector, we use the hash table and normalize it to have unit length. We do not convert mathematical vectors to documents, but we do make sure that we "attach" the document's name with its vector so we can remember the correspondence.

Two documents are similar if their dot product is large.

3.4 Misleading Comment About Using `for_each`

Jeffrey may have misled CS1321-1 during Tuesday's class. Using `for_each` or `transform` to multiply each vector component by a particular number will not work using what we currently know. The same is true for performing the dot product between the search vector and each of the document vectors. Just write a loop to do the work.

3.5 Explicitly Creating Iterators

A reminder on how to declare iterators:

`string:iterator` creates an iterator that permits walking through a string, reading, and modifying the string.

`string:const_iterator` creates an iterator that permits walking through the string and reading, but not writing.