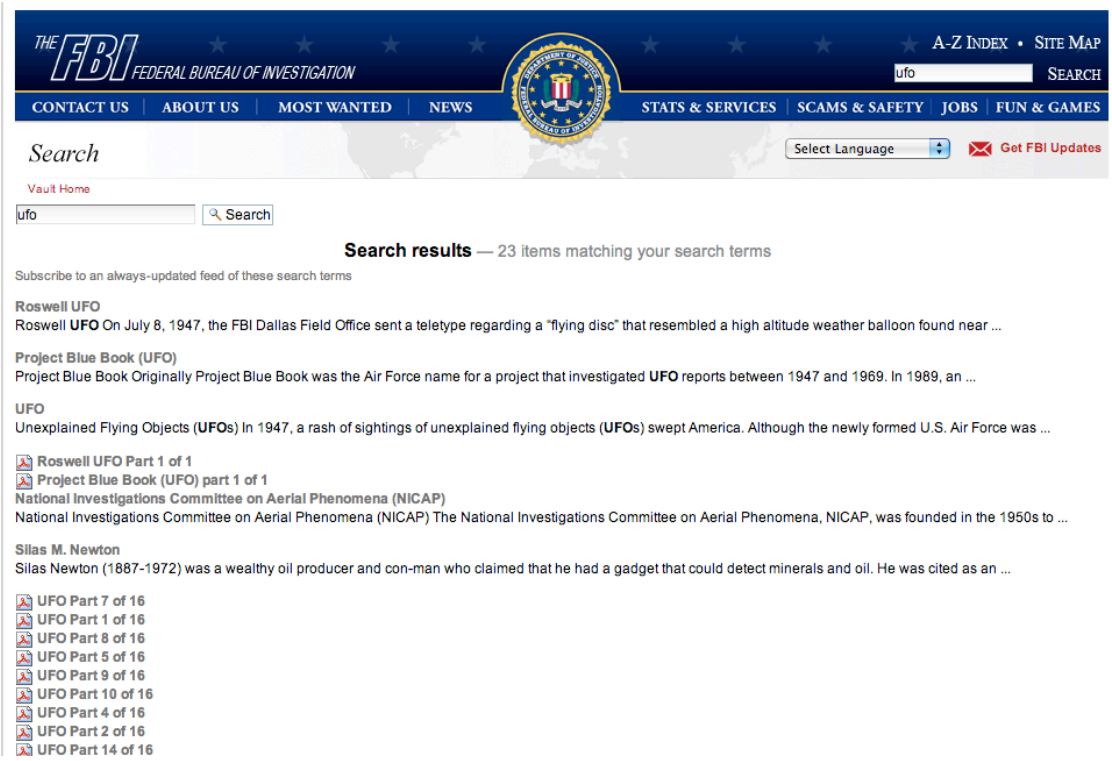
Homework: Content extraction and search using Apache Tika
Due: March 25, 2014

---

1. Objective

In April 2011, the U.S. Federal Bureau of Investigation launched its *Vault* site (http://vault.fbi.gov). The site contains over 6000 declassified documents that have been scanned from paper and made available via a digital content management system.

One of the many documents on the FBI web site having to do with the subject of UFOs (Unidentified Flying Objects) We went to the search portion of the site and did a query for UFO:



23 hits come back from the search[1]. Of the 6000 or more documents, 23 is a fairly small number (it amounts to less than 0.5% of the total corpus) and it seems like there would be a larger number of documents in the FBI's vault concerning the subject of UFOs. In this assignment, you are going to use Apache Tika to help us test our theory.

2. Preliminaries

2.1 Downloading the dataset

The entire corpus of PDF files from the FBI's vault web site has already been collected and packaged into a tarball. Download it from http://baron.pagemewhen.com/~chris/vault.tar.gz and unzip it onto your local hard drive.[2]

---

[1] In January 2012 when this query was first entered, 269 hits came back.
[2] Note: this is a 13GB file – ensure that you have sufficient disk space. A wired internet connection is recommended.

2.2 Downloading provided source code and files

Some skeletal source code (`TikaSkeleton.java`) has been provided to get you started on this assignment. An initial set of search keywords (`keywords.txt`) is also available. Obtain both from the "Assignments" section of our course webpage at http://www-scf.usc.edu/~csci572/.

2.3 Downloading Apache Tika

The quickest and best way to get Apache Tika up and running on your machine is to grab the `tika-app.jar` from: http://tika.apache.org/download.html. You should obtain a jar file called `tika-app-1.4.jar`. This jar contains all of the necessary dependencies to get up and running with Tika by calling it your Java program.

Documentation is available on the Apache Tika webpage at http://tika.apache.org/. API documentation can be found at http://tika.apache.org/1.4/api.

You can also get more information about Tika by checking out the book written by Professor Mattmann called "Tika in Action", available from: http://manning.com/mattmann/.

2.4 Environment Setup

Place the provided source code and files along with the Tika jar file that you have downloaded into your desired working directory. Rename the source code for the program to `TikaHW.java`. Unpack the corpus of PDF files in the same location. You should now have a directory structure looking like this:

```
<working directory>
 +-- TikaHW.java           // rename the file!
 +-- tika-app-1.4.jar
 +-- keywords.txt
 +-- vault
      +-- <2000+ files>
```

You should compile your program like this[3]:

```
javac -classpath .:tika-app-1.3.jar *.java
```

And execute it like this:

```
java -classpath .:tika-app-1.4.jar TikaHW
```

The grading environment will be set up in an identical fashion, so please ensure that your program compiles and executes correctly under this setup.

[3] Windows users: use semicolon (;) in place of colon (:) in the classpath.

3. Task description

With the corpus of PDF files in hand, you are going to write a Java program `TikaHW.java` that will:

- Accept an input file containing any number of search keyword(s) and/or phrase(s)
- Iterate through the corpus of data on your local computer.
- Call Apache Tika to extract all of the text from each PDF file
- Scan the extracted text to search for the given search keyword(s)
- Count and output statistics about
  - the number of documents that contain the keywords
  - the total number occurrences of the keywords
  - list of files containing the keywords

You will run the program that you have written and analyze the results to help us decide whether or not the FBI's search is telling us the truth when it says < 1% of its vault corpus has to do with UFOs, or whether or not something more is hiding in that rich treasure trove of information.

3.1 Using Tika from within your Java program

To use Tika in your Java program, simply include the `tika-app-1.4.jar` jar file in the classpath when you compile and execute your program, as has been demonstrated in Section 2.4.

3.2 Input file

The provided source code will read in a text file `keywords.txt` containing the search keywords or phrases to be used. This file can contain any number of keywords or phrases, each on a separate line. You do not need to modify this part of the program.

This file should be placed in the same working directory as the rest of your program, as described in Section 2.4.

`keywords.txt`

```
UFO
flying disc
disc
saucer
extraterrestrial craft
flying saucer
```

The provided `keywords.txt` file already contains an initial six keywords. You will need to come up with additional keywords, append them to the file, and include it with your submission. See Section 3.6.

3.3 Iterating through the corpus

The provided source code will iterate through the `vault` subdirectory and process each PDF file that it finds. You do not need to modify this part of the program.

3.4 Extracting text from PDF files using Tika

You will need to modify the `processfile` procedure in `TikaHW.java` so that it will take each PDF file that is passed to it, and extract its textual content using Tika. Refer to Professor Mattmann's presentation in class for details: http://www-scf.usc.edu/~csci572/Slides/Mattmann_Tika.pdf.

Additionally, you may find these resources useful:

`Parser` interface documentation: http://tika.apache.org/1.4/parser.html
`PDFParser` API: http://tika.apache.org/1.4/api/org/apache/tika/parser/pdf/PDFParser.html
`BodyContentHandler` API: http://tika.apache.org/1.4/api/org/apache/tika/sax/BodyContentHandler.html

3.5 Counting statistics and program output

Once the text has been extracted from the PDF file, you will need to search through it for the keywords from the `keywords.txt` file. How you do this is up to you – you may use Java's built-in String functions, regular expressions, etc.

Your program should keep track of statistics and output the following:

- `Output.txt` – will contain
  1) List of keyword(s) used
  2) No of files processed
  3) No of files containing keyword(s)
  4) No of occurrences (count) of each keyword(s)

- `Log.txt` – will contain an entry every time a keyword is found in a file:
  1) Filename of PDF file containing keyword
  2) Timestamp (time when the file was processed)

Much of the output functionality has already been implemented for you. You do not need to modify those parts of the program that have already been provided.

However, you will need to increment the counters `num_fileswithkeywords` and in `keyword_counts` when your program finds one of the keywords in the extracted text. You should also call the `updatelog` procedure to update the log file with each found keyword.

3.6 UFO, Saucer, Flying Disc?

You might see that there is not a direct mention of the word "UFO". That's because it wasn't widely used until later history. Terms used to describe UFOs back in the 1940's including "flying saucers"[4], "flying discs", "discs", "saucers", "extraterrestrial craft", and the like. We'd like you to include these terms as well in your search. You should also come up with an additional 5 aliases for UFOs, bringing your total to at least 11 total words that you scan for in the extracted text per document. Append your search terms to the provided `keywords.txt` and include the file in your submission.

[4] Which coincidentally was a term first coined by reporters after speaking with Kenneth Arnold regarding a set of 7-9 flying triangles he saw on a routine flight to Washington state, referring to the triangles as "saucers, skipping on a pond.". http://en.wikipedia.org/wiki/Kenneth_Arnold_UFO_sighting

3.7 Hints

Tika might not extract fully-formed text on each of the documents, depending on the quality of the OCR scanning that was used. You *will* encounter a wide variety in the extracted text, depending on the nature of the original files – ranging from those with clean, embedded text, to those with some OCR errors, to files filled with OCR noise. This is expected given the unsanitized nature of this "realistic" data set.

4. Report

Write a short 1 page report describing your observations, i.e. what you noticed about the dataset as you tweaked your program and tried to search different keywords. Did the corpus really only contain 23 documents about UFOs? What do you think are some reasons for the discrepancy between the number of documents you found as compared to the reported figure of 23?

Also include your thoughts about Apache Tika – what was easy about using it? What wasn't?

5. Options (Extra Credit)

For extra credit, you may consider using a string edit-distance computation (such as Levensthein distance http://en.wikipedia.org/wiki/Levenshtein_distance) to find words similar to UFO or saucer, but perhaps not exactly in your list of 11 and use that to increase the possibility of detecting documents that truly contain information about UFOs. Note that you do not have to implement the Levenshtein distance algorithm yourself – it's freely available in the Apache Commons Lang library, here:
http://commons.apache.org/lang/apidocs/org/apache/commons/lang3/StringUtils.html

You should also prepare a short write-up briefly describing the additional features that you implemented, and quantify their impact – did they help improve your program's ability to find UFO-related documents? What new documents did your improved program find (that were not found by your original program)? Why?

Other ideas are also welcome – however please consult Professor Mattmann about viability before proceeding with any original ideas.

Note that extra credit submissions are handled separately, so if you choose to do the extra credit, you should first keep a copy of your original program for the standard submission – do not overwrite the original with the extra credit!

6. Submission Guidelines

This assignment is to be submitted ***electronically, by 12pm PT*** on the specified due date, via Gmail to chris.mattmann@gmail.com. Use the subject line: CSCI 572: <Horowitz|Mattmann>: Spring 2014: Tika Homework: <Your Lastname>: <Your Firstname>. So if your name was Lord Voldemort, and you were taking Dr. Horowitz's course, you would submit an email to chris.mattmann@gmail.com with the subject "CSCI 572: Horowitz: Spring 2014: Tika Homework: Voldemort: Lord" (no quotes).

- All source code is expected to be commented, to compile, and to run. You should have (at least) one Java source file: `TikaHW.java`. You should also include other java source files that you added, if any. Do **not** submit `*.class` files. We will compile your program from submitted source.
- Include your `keywords.txt` containing the 11 keywords that you scanned, along with your output files `output.txt` and `log.txt`.
- Also prepare a `readme.txt` containing any notes you'd like to submit.
- Do **not** include `tika-app-1.4.jar` and the PDF repository in your submission. We already have these. Your code will be compiled and executed according to the environment described in Section 2.4.
- However, if you have used any external libraries other than Tika, you should include those jar files in your submission, and include in your `readme.txt` a detailed explanation of how to use these libraries when compiling and executing your program.
- Save your report as a PDF file (`Lastname_Firstname_TIKA.pdf`) and include it in your submission.
- Compress all of the above into a single zip archive and name it according to the following filename convention:

  <lastname>_<firstname>_CSCI572_HW_TIKA.zip

  Use only standard zip format. Do **not** use other formats such as zipx, rar, ace, etc.

Important Note:

- Make sure that you have attached the file the when submitting. Failure to do so will be treated as non-submission.
- Successful submission will be indicated in the assignment's submission history. We advise that you check to verify the timestamp, download and double check your zip file for good measure.

6.1 Late Assignment Policy

- -10% if submitted within the first 24 hours
- -15% for each additional 24 hours or part thereof

6.2 Extra Credit Submissions

Extra credit submissions should be according to the previous submission guidelines and included with the delivered aggregated zip file mentioned in Section 6 above.