CIS 345 Final Project: XML Query Engine Using Apache Lucene™

Fall 2014

Kevin Barthman
Ryan Bemowski

1) Description

XML Search with Apache Lucene written in C#

- Builds indices for XML files using Apache Lucene.
- Search indexed data with queries.
    - Based on Apache Lucene query syntax.
    - Return filename, element name and content of element.
- Allows search against any data schemas.
- Index path should contain 100 or more XML files.
- Use of Apache Lucene™, a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform.
    - More information at: http://lucene.apache.org/core/
- Utilizes IKVM to allow the usage of Apache Lucene™ libraries entirely from C#.s

2) How to Use the Program

- Installation
    - .NET 1.1 or later must be installed on the system
    - Extract the contents of RyanBemowskiKevinBarthmanProject.zip to any location
    - Place all XML documents to be indexed within the ‘xmlDocs’ directory, which is located in ‘Program/xmlDocs’ from the zip file.
- Usage
    - Run the CIS345FinalApplication.exe file located in the Program folder
    - Once the program has loaded and indexed all the files, there are three components to use to build the query for the XML files:
        - The ‘Search Tag’ dropdown allows the user to specify an XML tag to search within. Selecting the blank option searches against all XML tags

        - The ‘File to Search’ dropdown allows the user to specify a specific file to search within. Selecting the blank option searches against all XML files
        - The ‘Search Query’ textbox allows the user to build a query using the syntax used by Apache Lucene. Keep in mind:
            - An empty string will not return any results.
            - Use of quotations (“) must be used in pairs
            - If quotations are used, they must begin and end the string.
                - **Example:** “john”
            - Find more information about Apache Lucene query strings at: http://lucene.apache.org/core/2_9_4/queryparsersyntax.html
        - The ‘Number of Results’ textbox allows the user to alter the number of found items that are shown in the ‘Results’ listbox.
    - When done specifying search parameters, click the ‘Run Query’ button
    - The results are listed in the ‘Results’ data grid, with the columns File, Element, and Content for every hit returned by the query
    - The query parameters can then be rebuilt as many times as wanted, and the results will be updated whenever ‘Run Query’ is clicked

3) Test Cases

These tests can be ran to ensure proper error handling within the program

- **Test 1)**
    - Make sure the contents of the ‘xmlDocs’ directory is empty
    - Run the program
        - **Results**
            - When the program is run, the user sees a message that tells the user there are no XML documents in the ‘xmlDocs’ directory
            - *You must have XML documents in the ‘xmlDocs’ folder of the project.*
- **Test 2)**
    - Place an improperly formatted XML document in the ‘xmlDocs’ directory
    - Run the program
        - **Results**
            - When the program is run, the user sees a dialogue box that tells the user the XML file that is improperly formatted.

        - *All XML documents in the 'xmlDocs' directory must be properly formatted.*
- **Test 3)**
    - Place valid XML files in the 'xmlDocs' directory
    - Run the Program
        - **Results**
            - When the program is run, the user will see the application open with indexing updates in the bottom update textbox. The text 'Ready' will be shown when indexing has completed.
            - *With properly formatted XML documents in the 'xmlDocs' directory the application will run as expected.*
- **Test 4)**
    - Given a file named person.xml in the 'xmlDocs' directory:

```
<person>
    <firstName>John</firstName>
    <lastName>Smith</lastName>
<person>
```

    - With the program running
        - Type **john** in the 'Search Query' textbox
        - Then press the 'Run Query' button.
            - **Results**
                - After the query has completed the user will see the following record in the 'Results' listbox.
                    - **person.xml** | **firstName** | **John**
                - *A single item was found.*
- **Test 5)**
    - Given the same file as Test 4.
    - With the program running
        - Select **person** from the 'Search Tag' dropdown
        - Type **john** in the 'Search Query' textbox
        - Then press the 'Run Query' button
            - **Results**
                - After the query has completed the user will see a message in the update textbox near the bottom of the form telling the user that no records were returned.
                - *No Results*
- **Test 6)**
    - Given the same file as Test 4.

- With the program running
  - Select **firstName** from the ‘Search Tag’ dropdown
  - Select **person.xml** from the ‘File to Search’ dropdown
  - Type **john** in the ‘Search Query’ textbox
  - Then press the ‘Run Query’ button
    - **Results**
      - After the query has completed the user will see the following record in the ‘Results’ listbox.
        - **person.xml** | **firstName** | **John**

4) System Functionalities and Requirements

This application uses a simple C# GUI to utilize the text indexing abilities of Apache Lucene™. Prior to running this application, you must ensure your system meets the application requirements.

Requirements

- .NET 1.1 or later must be installed

Test Data

- Within the project directory there is a ‘xmlDocs’ directory. This directory contains over 100 XML files that are use for testing this application. These files can be altered, deleted or replaced.
  - **Note:** There must be at least 1 properly formatted XML document inside this directory for the application to function correctly. There must also be no improperly formatted XML documents within this directory.
- A file called ‘improper.xml’ will be in the ‘xmlDocs’ directory to demonstrate the application’s ability to handle improperly formatted XML. To use the application as intended, remove this file from the directory.
- Any properly formatted XML will work with this application.

Accompanying DLL Files

- The entirety of the IKVM framework’s DLL files must be in the same folder as the executable.
- The Apache Lucene essential JAR’s must be converted to DLL files using IKVM and placed in the same directory as the CIS345FinalApplication.exe file