Chapter 5

Implementation of classification of unstructured data from online course web pages

Implementation of any software is always preceded by important decisions regarding selection of the platform, the language used, etc. these decisions are often influenced by several factors such as the real environment in which the system works the speed that is required, the security concerns, other implementation specific details etc.

5.1 Programming Language Selection

The programming language plays a major role in the efficiency as well as the future development of the project. As such, Java has been chosen as the programming language to be used.

5.2 Platform Selection

The platform selected for the project is Windows 7, as it is a compatible OS for running Eclipse, NetBeans, Weka and other tools. Eclipse is selected for performing web crawling and machine learning related code with high level of dependency on java libraries. NetBeans is used to develop a fresh and intuitive user interface which can process and present the data generated after running the Eclipse code. Weka is used to visualize the graphical evaluation and tabulated results of the various machine learning algorithms.

5.3 Code Conventions

Coding conventions are a set of guidelines for a specific programming language that recommend programming style, practices and methods for each aspect of a piece program written in this language. These conventions usually cover file organization, indentation, comments, declarations, statements, white space, naming conventions, programming practices, programming principles, programming rules of thumb, architectural best practices, etc. These are guidelines for software structural quality. Software programmers are highly recommended to follow these guidelines to help improve the readability of their source code and make software maintenance easier. Coding conventions are only applicable to the human maintainers and peer reviewers of a software project. Conventions may be formalized in a documented set of rules that an entire team or company follows, or may be as informal as the habitual coding practices of an individual. Coding conventions are not enforced by compilers.

As a result, not following some or all of the rules has no impact on the executable programs created from the source code [52].

5.3.1 Naming Conventions

All class, function and variable names have been chosen keeping in mind that they must reflect the purpose of that class/function/variable. These have been followed with the exception of local counters which are used and discarded almost immediately. For example, the counter of a FOR-loop [53].

5.3.2 File Organization

Efficient file organization for the Classification System may be done as follows:

- Use a separate folder for each project
- Write header comments

The above points are further explained as follows:

Managing Folders

The Current Folder Browser provides a few features to make managing separate folders easier. The tree views multiple projects from a root directory. Having a top-down hierarchical view makes it easy to move files between project directories. The address bar is used for quickly switching back and forth between project directories. This allows keeping only one of these folders on the Classifier search path at the same time. If there is a nested folder structure of useful functions that needs to be accessed (for example a hierarchical tools directory), “Add with Subfolders” from the context menu can be used to quickly add a whole directory tree to the Classifier search path.

Write Header Comments

Having comment lines in files enables the functions, scripts, and classes to participate in functions like MainControlCenter and CrawlerController When a directory is supplied to the help function it reads out a list of functions in that directory. [52]

5.3.3 Properties Declarations

The Classification System makes use of a limited number of operators and properties. The maximum scope of the allowed properties are similar to those found in Delicious REST connection API which requires an account to assign social tags to crawled web pages while featurizing them.

5.3.4 Class Declarations

The Classification System is completely object-oriented. Thus every block of code resides in a class. Standard Class Naming conventions are used. The Classification System classes use the following words to describe different parts of a class definition and related concepts.

- Class definition - Description of what is common to every instance of a class.
- Properties - Data storage for class instances
- Methods - Special functions that implement operations that are usually performed only on instances of the class
- Events - Messages that are defined by classes and broadcast by class instances when some specific action occurs
- Objects - Instances of classes, which contain actual data values stored in the objects' properties
- Subclasses - Classes that are derived from other classes and that inherit the methods, properties, and events from those classes (subclasses facilitate the reuse of code defined in the superclass from which they are derived).
- Superclasses - Classes that are used as a basis for the creation of more specifically defined classes (i.e., subclasses).
- Packages - Folders that define a scope for class and function naming [52].

5.3.5 Comments

Comment lines begin with the character '//', and anything after a '//'character is ignored by the interpreter. The // character itself only tells the interpreter to ignore the remainder of the same line. In the Classification system, commented areas are printed in green by default, so they should be easy to identify. There are two useful keyboard shortcuts for adding and removing chunks of comments. Select the code to be commented or uncommented, and then press Ctrl-K + Ctrl-C to place one //' symbol at the beginning of each line and Ctrl-K + Ctrl -U to do the opposite.

Comments for blocks of code are started by a ‘/*’ and are delimited by a ‘*/’.
In WPF, the symbol ‘<!--‘ is used to begin a comment block while ‘-->’ is used to end it. The ‘///’ comment character is used just before major functions to indicate the role of the specific function. Unlike the ‘//’ comment character, it is grey by default.

Common uses

Comments are useful for explaining what function a certain piece of code performs especially if the code relies on implicit or subtle assumptions or otherwise perform subtle actions. For example,

```
// Calculate average velocity, assuming acceleration is constant
// and a frictionless environment.
//force = mass * acceleration [52].
```

5.4 Difficulties Encountered and Strategies Used to Tackle

This project had several key challenges, which had to be addressed and resolved. The most pertinent few have been listed here.

5.4.1 Crawling and Classification Logic

The task of crawling the entire web page and extracting content and HTML related features from each URL obtained was extremely time consuming and difficult. Each web page has its own structure of embedding data and most web pages are not crawler or search-engine friendly in terms of metadata and indexing.

To solve this problem, domain specific parsing was used to narrow down the scope of pages along with filters given to the crawler so as not to crawl stylesheets, blogs, downloadable content or other non MIME data which could not be processed.

While classifying this data, the challenge encountered was the presence of ‘outliers’ or rogue characters that appear in data due to incorrect webpage design. These characters could abruptly terminate a string, cause exceptions and obfuscate data. In order to normalize and clean the data, these characters have to be eliminated at an early stage. Hence the feature string was made to be of uniform and consistent format for all links crawled in order to solve this problem and classify efficiently.

5.4.2 Storage and Display Logic

Since the system aims to display a large amount of information, it must be able to store this information efficiently. It must be ensured that data is saved after the Bookmark button is pressed. Also, it must be ensured that all data is once again displayed when the system is re-opened.

To solve this problem, a CSV file is used to store data which is appended with new bookmarks each time. Another CSV file is used to store all the data used for the system, as the output of classification. This way data can be uniformly stored and queried using simple SQL queries.

5.5 Summary

This chapter has presented a concise explanation of the various objected oriented concepts and theoretical fundamentals used in implementing the various functions and features envisioned in this project. The use of various standards and conventions in variable naming, use of modularity, object oriented programming and storage or display logic have been detailed to describe the efficiency of the project in complying with standard industry practices in coding,