Analysis Report

1. Introduction

The system aims to classify and filter the type of documents and return the ones that are in the catalogue(type) the user wants. In the information age, paper works are gradually replaced by the bytes and bits in the disk storage. We may need to be buried in those myriads of documents stored in order to find the ones we need. It would be realizable but time-consuming and painstaking if we read all of them and classify them manually. With this system, just several steps and seconds, all the documents will be smartly located into the catalogues they belong to. User thus can easily get the information they want.

Various applications can be realized based on this system.

Example 1: Given a pile of news documents, the system can classify the fields the news belong to and recommend some to the user according to the requirement.

Example 2: In a folder with papers of different fields, the system can classify them and only return the papers the user needs.

Example 3: There are many type of user manuals of a series of equipment(e.g. electrical appliances), the system can classify them into different functions. (e.g. indoor use)

\*(Actor TABLE)

2. User Story & Use Case

Our system can “learning” to be more powerful with a local dictionary that can be expanded.

There are two stages: training - generate/expand the dictionary and functioning - perform system classification and filtering function.

2.1 Training Stage

2.1.1 User Story

| # | Name | Description |
| --- | --- | --- |
| US1 | Select the dictionary | User can choose one of the two modes to set the dictionary according to which the program will classify the passages into different catalogues. |
| US2 | Input the text files/URLs | If user choose to construct customized dictionary, he/she will need to place the passages found in txt or URLs for needed content into designated folders. |
| US3 | Build up dictionary | User input keyword to start the program and call the critical method to build the dictionary. |
| US4 | \*Add new classification | User can add a new classification and define the update the dictionary for it. |

2.1.2 Use Case

| User Responsibility | System Responsibility |
| --- | --- |
| 1.1.1User chooses using pre-defined dictionary setup  1.2.1User chooses to define the dictionary |  |
|  | 1.1.2 System detects the input and will use the pre-defined dictionary settings to build the dictionary  1.2.2 System detects the input and require the user to provide documents |
| 1.1.3 User place the required documents and catalogues |  |
| 2. User inputs key word “R” to run the dictionary setup |  |
|  | 3. System build the dictionary with the setups and files |
| 4.1.1. User can add more classification after dictionary building(expand the dictionary) |  |
|  | 4.1.2 The system detects the input and find the newly added catalogues and expands the dictionary. |

2.2 Functioning Stage

| # | Name | Description |
| --- | --- | --- |
| US1 | Select source files | User can select the files he/she wants to classify. For example, a folder(or .zip file) containing 100 papers |
| US2 | Do classification and get feedback | User can input the category of the files and get feedback of that category or input keywords for viewing a file of designate category. The program will automatically classify all the files input by the user and then store the basic information for each classification like number or name of the files. |
| US3 | \*Get Recommendation | The program will note down the history of the user,user can choose to get the categories he/she views very often. |

2.2.1 User Story

2.2.2 Use Case

| User Responsibility | System Responsibility |
| --- | --- |
| 1. User input the path of the documents to be classified |  |
|  | 2. System detects the input and doing classification according to the dictionary built in the previous stage. |
| 3.1.1 User input the catalogue needed  3.2.1 User input keyword to get recommendation of random documents |  |
|  | 3.1.2 System detects the input and return the list of files of that catalogue  3.2.2 System detects the input and looks up the use history of the user and recommend the documents from most visited catalogues |

3. Combined User Story

Mr. M has thousands of papers stored in his disk drive, but they are in a mess, all in one folder. Mr. M meets with problem when he wants to find the papers about socialization. Then M starts the system and inputs the dictionary he wants to use. Several seconds later, the papers are automatically classified and the folder containing socialization papers in prompted to M. Now M can happily read the papers.

4. Domain Modeling

4.1 Image(TBC)

4.2 Domain Entity(template)

|  |
| --- |
| Class Name: |
| Description: |
| Required: |
| Provide: |

4.3 Sequence Diagram