Installation, Running and Editing Instructions for the ClassificationBox Text Classification Tool

Technical Report: CMU-RI-TR-04-58

Michael D. Rectenwald, Young-Woo Seo, Kevin C. Lee, Joseph A. Giampapa, Katia Sycara

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA USA

This research has been supported by Computer Technology Associates, Inc. subcontract `BAA 03-02-CMU', under ARDA contract `BAA 03-02-FH'.

Introduction

Given that we are not familiar with a target environment for text classification (i.e., the particular characteristics of the target data set, the categories of documents represented, etc.), it is desirable to have a tool that is able to automatically determine which is the best classifier for a given data set.

ClassificationBox program determines the best classification method and its parameters for classifying a particular data set. ClassificationBox is a stand-alone java application for text classification. The program was developed in order to determine which classification methods and parameters are optimal for a given data set.

We have provided two data sets on which you can run the program: the one under the directory entitled "**text_data1"** and the other under the directory entitled " **text_data2"**

Installation and Running Instruction for the ClassificationBox (CBox.class) program

1. Save the `ClassificationBox` and one of data sets (either `text_data1` or `text_data2` folders to a local drive).
2. In `ClassificationBox`, open the task specification file (`text_classification.xml`). Note: the name of this file may be an arbitrary English name. Edit the parameters to reflect the location of the index file and the directory of data sets in your local drive, as loaded above.
    a. For example, assuming that you saved the above folders directly to the root of the C drive, change parameters as follows:

```
 <dataset name="relevant_subjects"
dataset_location="c:\text_data1\"
document_index_file="c:\ClassificationBox\cbox\data\document_i
ndex_text_data1.idx" dataset_type="text"
```

    If you are using "`text_data2`," you will need to edit the task specification file (`text_classification.xml`) to reflect the index file for that dataset (`document_index_text_data2.idx`).

    Note: The description of each of fields in a task specification file is described in the file named "Development_of_ClassificationBox.pdf," which is located in http://www.cs.cmu.edu/~softagents/textclasscd/

    These changes will allow `ClassificationBox` to find the data file you are running, in this case `text_data1` or `text_data2` and the index (`document_index_text_data1.idx or document_index_text_data2.idx`) of the data set you are running the file across.

Note: You will need to edit `text_classification.xml` and `document_index_text_data1.idx` or `document_index_text_data2.idx` if you are running the program on your own data set or on data. See following section for editing these files to run the program on your own data set. Again, the names of these files may be arbitrary English names (with the extensions). If you make changes to the file names, be sure to reflect the changes in the appropriate task specification file.

3. To run the ClassificationBox program (`CBox.class`), you must have Java installed on your computer. Go to http://java.sun.com and download Java 2 Platform Standard Edition 5.0. Note: The names of java executable files (i.e., class files) are case-sensitive. For example, the following are two different files: Cbox.class and CBox.class.

4. Once Java is installed, you can run `CBox.class` from a command prompt. In Windows, go to Start and in Run, type "cmd."

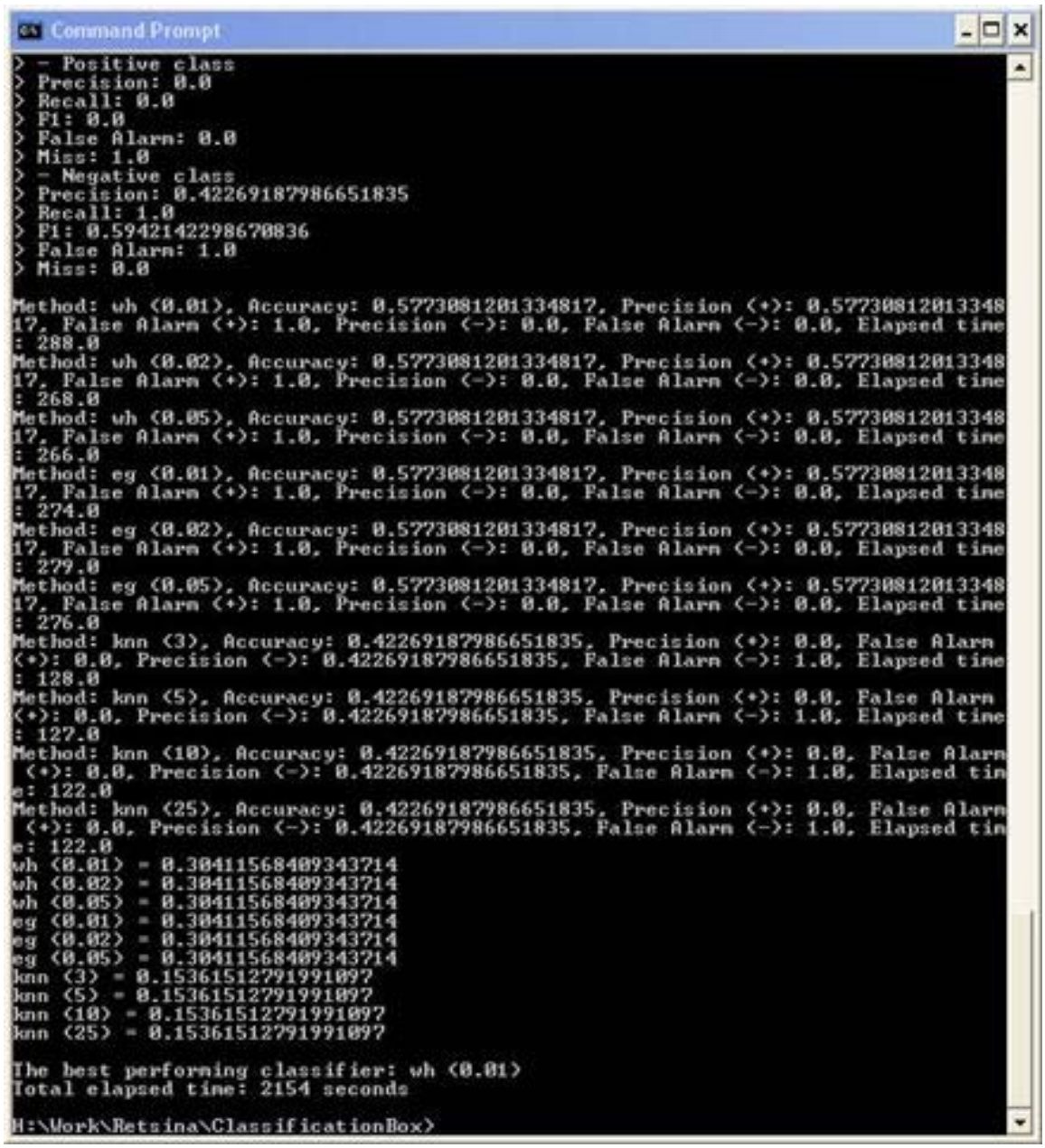
5. Type `java -version` to make sure Java is running. Any java version higher than 1.2.x. will work.

6. Assuming that you wish to run the program using the provided classification file, type

```
java CBox.class text_classification.xml
```

7. The program will run on the data set. The process is very computation intensive and can take 30 minutes or more (depending on data set used). The program takes approximately 30 minutes to find out the best classification method for four categories in the data set `text_data1` containing 1,997 documents using ten different classification methods/parameters: terrorism (1,153), al qaeda (272), arms proliferation (331), and narcotics (241).

8. As the result of text learning, `CBox.class` will generate a text file, [*task_name*]`.out`, which lists all test scores. "`task_name.out`" is one of parameters in the `text_classification.xml`. The output shows the performance of each of the classification methods in terms of accuracy, precision (positive class /negative class), false alarm (positive class/negative class; a.k.a. "false positives"), elapsed time in seconds, as follows:


Command Prompt
> - Positive class
> Precision: 0.0
> Recall: 0.0
> F1: 0.0
> False Alarm: 0.0
> Miss: 1.0
> - Negative class
> Precision: 0.42269187986651835
> Recall: 1.0
> F1: 0.5942142298670836
> False Alarm: 1.0
> Miss: 0.0
Method: wh (0.01), Accuracy: 0.5773081201334817, Precision (+): 0.57730812013348
17, False Alarm (+): 1.0, Precision (-): 0.0, False Alarm (-): 0.0, Elapsed time
: 288.0
Method: wh (0.02), Accuracy: 0.5773081201334817, Precision (+): 0.57730812013348
17, False Alarm (+): 1.0, Precision (-): 0.0, False Alarm (-): 0.0, Elapsed time
: 268.0
Method: wh (0.05), Accuracy: 0.5773081201334817, Precision (+): 0.57730812013348
17, False Alarm (+): 1.0, Precision (-): 0.0, False Alarm (-): 0.0, Elapsed time
: 266.0
Method: eg (0.01), Accuracy: 0.5773081201334817, Precision (+): 0.57730812013348
17, False Alarm (+): 1.0, Precision (-): 0.0, False Alarm (-): 0.0, Elapsed time
: 274.0
Method: eg (0.02), Accuracy: 0.5773081201334817, Precision (+): 0.57730812013348
17, False Alarm (+): 1.0, Precision (-): 0.0, False Alarm (-): 0.0, Elapsed time
: 279.0
Method: eg (0.05), Accuracy: 0.5773081201334817, Precision (+): 0.57730812013348
17, False Alarm (+): 1.0, Precision (-): 0.0, False Alarm (-): 0.0, Elapsed time
: 276.0
Method: knn (3), Accuracy: 0.42269187986651835, Precision (+): 0.0, False Alarm
(+): 0.0, Precision (-): 0.42269187986651835, False Alarm (-): 1.0, Elapsed time
: 128.0
Method: knn (5), Accuracy: 0.42269187986651835, Precision (+): 0.0, False Alarm
(+): 0.0, Precision (-): 0.42269187986651835, False Alarm (-): 1.0, Elapsed time
: 127.0
Method: knn (10), Accuracy: 0.42269187986651835, Precision (+): 0.0, False Alarm
(+): 0.0, Precision (-): 0.42269187986651835, False Alarm (-): 1.0, Elapsed tim
e: 122.0
Method: knn (25), Accuracy: 0.42269187986651835, Precision (+): 0.0, False Alarm
(+): 0.0, Precision (-): 0.42269187986651835, False Alarm (-): 1.0, Elapsed tim
e: 122.0
wh (0.01) = 0.30411568409343714
wh (0.02) = 0.30411568409343714
wh (0.05) = 0.30411568409343714
eg (0.01) = 0.30411568409343714
eg (0.02) = 0.30411568409343714
eg (0.05) = 0.30411568409343714
knn (3) = 0.15361512791991097
knn (5) = 0.15361512791991097
knn (10) = 0.15361512791991097
knn (25) = 0.15361512791991097
The best performing classifier: wh (0.01)
Total elapsed time: 2154 seconds
H:\Work\Retsina\ClassificationBox>

Editing Parameters for Running Your Own Data Set

1. The file `Development_of_ClassificationBox.pdf` in the same directory (in this case `C:\ClassificationBox`) describes the parameters in the `text_classification.xml` file used by the `CBox.class` program. Using this file, you can learn how to generate your own task specification file. It describes each of parameters in a task specification file. The parameters listed in bold type are mandatory parameters whose values must be specified in the task specification file.

2. The task specification file can be edited to run on a new set of data files. Keep in mind that there are mandatory parameters. See the bolded parameters in the `Development_of_ClassificationBox.pdf file` whose values must be specified (if only to set them to "none").

3. You need not edit the actual parameters (red type) in the task specification file (in this case `C:\ClassificationBox\text_classification.xml`), but may only change the values for parameters (the black type), in order to change the parameters of your task.

4. If you wish to run `CBox.class` on a new data set, you will also need to generate a new index file to list the information of each of the files in your data set. (Again, thenames for the index and the task specification file can be arbitrary English names, followed by the extension. The index file name can be changed, but must remain in `\ClassificationBox\cbox\data\`. If you change the index name, be sure to change its name in the task specification file as well).

    a. The index file has the following structure. The columns are separate by horizontal tabs `(\t)` and the rows are separated by a carriage return (i.e., hit the "Enter" key):

    Sequence`\t` ID`\t` class`\t` filename`\t` t/e

    Sequence is the numeric sequence of the document in the file, starting from 1. The ID is the same as the sequence number. Class is the name of the category to which you assign documents. File name is the name of the document file. "t" tells CBox.class to use the document for training purposes (training the program to identify it) and "e" tells the program to evaluate the document. The class and file names must be continuous and cannot contain spaces.