CSCI 578 – ASSIGNMENT 2

Deadline: by 9 a.m. on Friday, March 10

Submission Information: Please submit on D2L (details near the end of this document).

WHAT THIS ASSIGNMENT IS ABOUT

In this assignment, you'll be taking a closer look at the RELAX recovery technique and how its results change based on which concerns are selected for training its classifier. Additionally, you will gain a general understanding of how text classification works.

BEFORE YOU GET STARTED

You will need the the virtual machine from the previous assignments (0.5 and 1). If you don't have it anymore, you can download it here.

Regardless of which version of the virtual machine you have, you'll need to run the upgrade script available here. **You will only be able to perform the steps necessary for this assignment after running this script!**

FIRST STEP – SELECTING YOUR SYSTEM AND ITS VERSIONS

For reasons of simplicity and familiarity, it is highly recommended that you stay with the two system versions you had selected for the previous assignment. If you would like to change your system and/or versions instead, please select two new versions and update your choice in this document. Please make sure that nobody else is already using this combination – your combination of system versions needs to be unique, as in the previous assignment.

SECOND STEP – LIST 20 CONCERNS

Making use of all knowledge you can gather about the system from publically available sources (e.g. documentation that comes with the distribution, the project's website, tutorials, secondary sources such as stackoverflow.com) and your own judgment, list 20 concerns that the system needs to address.

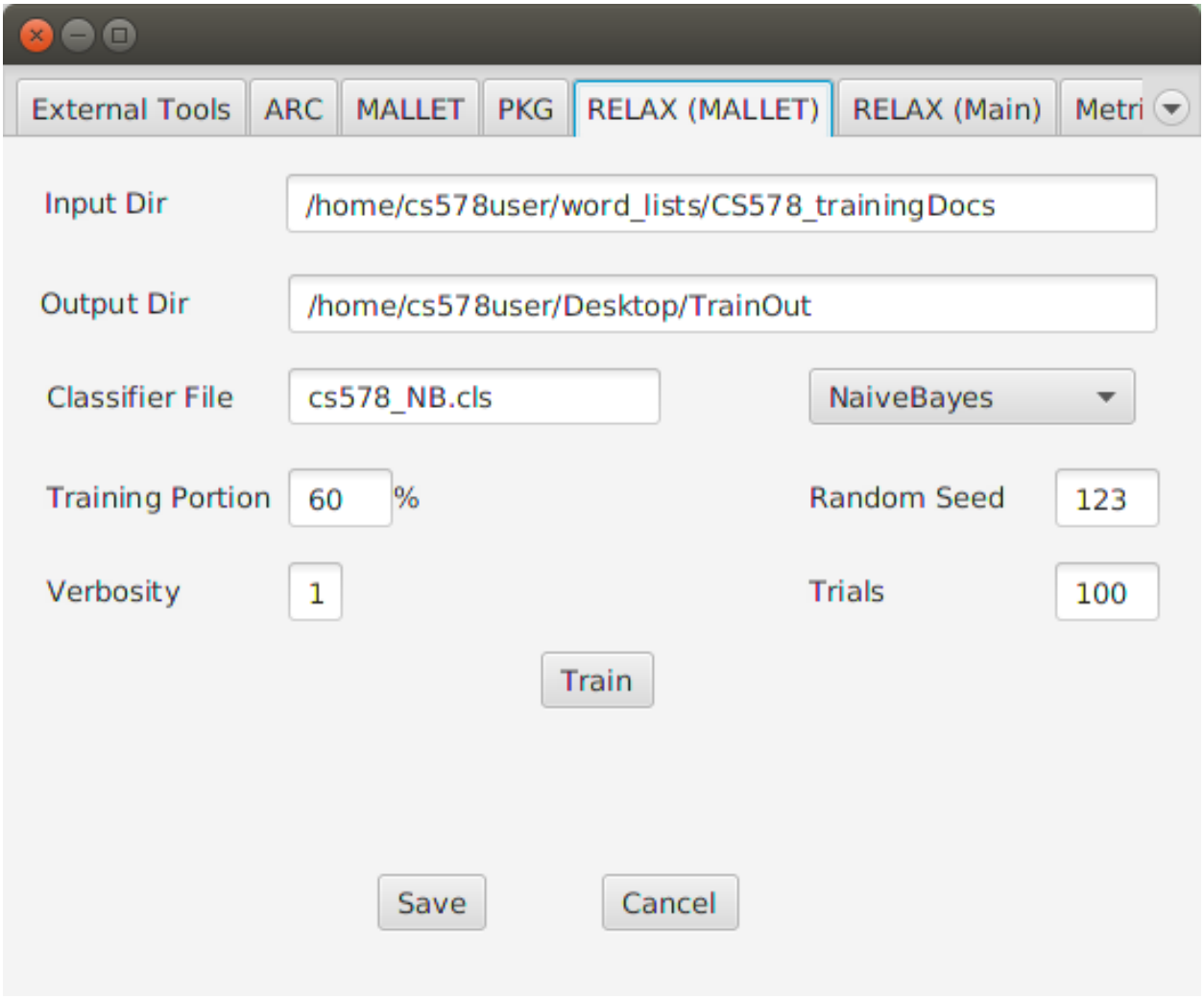
THIRD STEP – SELECT TOPICS FOR TRAINING

Running the upgrade script mentioned in the "Before you get started" section will create a "word_lists/trainingDocs" directory in your home directory and download 12 directories of training to it. Each of those 12 directories stands for a topic. Out of the 12 topics provided, select the 8 that match the concerns you have selected in the second step most closely and move the remaining 4 out of that directory.

FOURTH STEP – TRAIN CLASSIFIERS

Using the ARCADE GUI, you will now train 200 classifiers. (This may sound daunting, but it's actually really easy and fast, as shown in the lecture.)

1. Start the ARCADE GUI in Eclipse.
2. From the Menu, click Edit -> Preferences. Then select the "RELAX (MALLET)" tab. This should bring up the dialog below (fields that you are shown filled here may still have to be filled out by you):



3. Similar to the way you did in the first assignment, double-click on the text fields near "Input Dir" and "Output Dir" to fill them out with the directory that contains your word lists and the one that will the classifier trial files, respectively.
4. Select the "NaiveBayes" algorithm from the algorithms list.
5. Change the Random Seed to an integer of your choice.

6. Click on "Train". Your console output will look similar to this:

```
------------------- Trial 23  -------------------

Trial 23 Training NaiveBayesTrainer with 49 instances
Trial 23 Training NaiveBayesTrainer finished
Trial 23 Trainer NaiveBayesTrainer training data accuracy= 1.0
Trial 23 Trainer NaiveBayesTrainer Test Data Confusion Matrix
Confusion Matrix, row=true, column=predicted  accuracy=0.9696969696969697
         label   0   1   2   3   4   5   6   7  |total
  0   graphics   5   .   .   .   .   .   .   .  |5
  1        gui   .   3   .   .   .   .   .   .  |3
  2         io   .   .   4   .   .   .   .   .  |4
  3 networking   .   .   .   6   1   .   .   .  |7
  4   security   .   .   .   .   3   .   .   .  |3
  5      sound   .   .   .   .   .   4   .   .  |4
  6        sql   .   .   .   .   .   .   4   .  |4
  7       text   .   .   .   .   .   .   .   3  |3

Trial 23 Trainer NaiveBayesTrainer test data accuracy= 0.9696969696969697

------------------- Trial 24  -------------------

Trial 24 Training NaiveBayesTrainer with 49 instances
Trial 24 Training NaiveBayesTrainer finished
Trial 24 Trainer NaiveBayesTrainer training data accuracy= 1.0
Trial 24 Trainer NaiveBayesTrainer Test Data Confusion Matrix
Confusion Matrix, row=true, column=predicted  accuracy=0.8787878787878788
         label   0   1   2   3   4   5   6   7  |total
  0   graphics   3   .   .   .   .   .   .   .  |3
  1        gui   .   5   .   .   .   .   .   .  |5
  2         io   .   .   5   2   .   .   .   .  |7
  3 networking   .   .   .   5   .   .   .   .  |5
  4   security   .   .   .   1   2   .   .   .  |3
  5      sound   .   .   .   .   .   2   .   .  |2
  6        sql   .   .   .   .   .   .   1   .  |1
  7       text   .   .   1   .   .   .   .   6  |7

Trial 24 Trainer NaiveBayesTrainer test data accuracy= 0.8787878787878788
```

7. Scroll back and forth in the Eclipse console buffer until you find one or more trials whose confusion matrices contain the lowest number of false predictions. In the above example, this would be trial 23, which has one false prediction (whereas trial 24 has four). Alternatively, you can find the same information in the ARCADE log file, which is located in `~/arcade_userdir/arcade.log` and contains the console output. IMPORTANT: Do not select a classifier whose confusion matrix has any empty rows. Any accuracy rating computed for it by the training tool will be irrelevant since it will not be based on all topics from the training data.

8. Find your selected classifier in the classifier output folder that you selected in the GUI and copy it to your home directory.
9. Delete all remaining files in your classifier output folder.
10. Remove the ARCADE log file by issuing the following command:
    `rm ~/arcade_userdir/arcade.log`
11. Repeat steps 4-10 with the "MaxEnt" algorithm.
12. After you've selected your two classifiers, remove the log4j2 configuration by issuing the following command in a terminal: `rm ~/arcade_userdir/log4j2.xml`

IMPORTANT: You must give your two classifier data files different names from each other, and they must NOT be named "relax.classifier". It is recommended that you use the following naming scheme:

`CS578HW2_<First_Name>_<Last_Name>_<Algorithm>.cls`

For example, if your name is Jane Doe, your classifier files should be called:

`CS578HW2_Jane_Doe_NB.cls` and

`CS578HW2_Jane_Doe_ME.cls` (where NB stands for Naïve Bayes and ME for Maximum Entropy)

FIFTH STEP – RUN RELAX

For each classifier that you selected, run RELAX on your two system versions (this part works just like in the first assignment, only with the different classifiers.) Selecting a classifier works as shown in the first assignment.

SIXTH STEP – ANALYSIS

Your analysis will be based on a three-way comparison between

1. The results of your original run (HW1),
2. The results from your first selected classifier, and
3. The results from your second selected classifier.

Different from the first assignment, you'll stay within versions for your comparisons, i.e. If you have selected to recover chukwa-0.1.2 and chukwa-0.2.0, you'll first compare your three results for chukwa-0.1.2 to each other and then later you'll be comparing your three results for chukwa-0.2.0 to each other.

ANALYSIS PART I

List the concerns that you selected to train your classifiers on.

What are the major differences you are seeing between the recovered architectures for each version? Do you think the recovered architectures are accurate? Why or why not? How would you rank the three results in terms of their accuracy, and which criteria would you use to make that determination?

ANALYSIS PART II

Assuming you have never seen the system you selected before and wanted to get a useful overview of it, do you think the three views can be helpful? Why or why not? How would you rank the three results in terms of their utility, and which criteria would you use to make that determination?

Specifically, what do you think is missing from the recovered architectures? For example, can you determine where the system's connectors are or what the architectural style of the system is? What is missing or stands out in the visualizations you get from RELAX?

ANALYSIS PART III

To the extent that you didn't find the results to be helpful, how would you improve the results so they'd represent the system better or be more helpful?

PLEASE NOTE

- When ARCADE starts up, you may get some console errors about the StatusLogger and the GUI. These are expected and not a reason for concern.
- You'll need to restart ARCADE after each recovery run by closing the ARCADE Runner dialog and restarting it. (This is because not all data structures may be reinitialized after the first run).
- If your screen resolution is different from the screenshots shown here, it will lead to variations in how things will look on your screen. This is no reason for concern.
- Make sure you are registered on our piazza.com class discussion forum. Instructions for registering have been posted to D2L in the announcements.
- If you encounter any issues with any of the steps discussed here, please check on Piazza whether your issue has already been addressed. If not, please post a question on Piazza.

DELIVERABLES

Please submit the following files on D2L:

1. Your two different classifier files that you have used for this assignment, along with the confusion matrices for each. (Submit the confusion matrices in an ASCII text file.)
2. Create separate ZIP files for each recovery result (one for each run), to be named as follows: CS578HW2_<your_name>_<classifier_name>.zip
3. Your analysis as one PDF file. The three parts of your analysis need to have a minimum word count of 250 each. The maximum word count of your PDF document must not exceed 2500 words. (Only your own words figure in the word count. Lines you cite from RELAX output files or that occur in diagrams do not count as word for the purposes of the word count.)

GRADING

Grading consists of the following:

1. 5% for the set of files named in the first section of "Deliverables" above,
2. 5% for the set of files named in the second section of "Deliverables" above,
3. 30% for each part of your analysis, and
4. 10% **extra credit** for adding two or more new topics of your own in addition to the existing 8 you've already selected, as well as training and selecting your own two classifiers (Naïve Bayes and MaxEnt) based on those new topics. This will result in two classifiers that are trained on 10 or more topics. If you are going for this, please submit a zip file containing the training data for your new topics. **Note**: The classifiers and your analysis **take the place** of the regular ones in this case. You ONLY need to submit the items mentioned in 1., 2. and 3. for your 10+ topics based work.

PENALTIES:

1. For each part of the analysis, you lose 0.5% of the overall assignment score per word that you are short of 250.
2. For the overall analysis, you lose 0.1% of the overall assignment score per word that you are over 2500.

Please note that it is possible that both penalties can apply at the same time – you could have one or two parts that are too short and a third part that takes the word count of the whole analysis over 2500 words.

LATE SUBMISSIONS:

- Up to 48 hours after the deadline: 1% off for each hour late
- More than 48 hours after the deadline: No submissions accepted anymore

APPENDIX A – DESCRIPTIONS OF THE THREE RECOVERY METHODS

All three algorithms operate on the source code of a system and how it's organized in directories. That is, they're basing their architectural views on the system at rest, as opposed to looking at it while it's running.

ACDC clusters source files by following patterns that commonly occur in manual decompositions (manual recoveries) of software systems. It tries to identify subsystems using a collection of criteria that are based on connections between entities (such as directory structure, headers and body, support libraries and others), names the subsystems and then creates clusters from them.

ARC applies topic modeling, an NLP technique, to the source code. It treats the source code of a given system version as a body of text and determines which groups of words occur together most frequently. These groups of words are called "topics". For each source file it then determines which topic it is most aligned with and groups the source files into clusters for that topic. Parameters that need to be set for each run are the number of topics and the number of clusters. The meaning of a topic that has been found by ARC can only be determined by humans. (Note that we will not ask you to set any parameters or to determine the meaning of topics in this assignment. The parameters are already pre-set for you.)

RELAX applies text classification, another NLP technique, to the source code. It looks for predefined topics in each source file and determines which topic it is most aligned with. It then groups the source files into clusters for each topic. For it to be able to run, a classifier has to be trained from a set of groups of documents where the documents in each individual group are all related to one topic. No parameters have to be set for each individual run. (Note that we will not ask you to train any classifiers – those will be provided to you by us.)