Homework 2 – Document Classification

CS 4705: Natural Language Processing

Due:

Assignment:

Your assignment is to run a number of machine learning experiments on a set of news

data and describe your experiments and findings. The task involves classifying news

stories in a number of different ways:

1. By Source Type

The training and testing material will include both newswire (NWIRE) text digests and

broadcast news (BN) transcripts. This is a binary classification: BROADCAST vs. TEXT.

2. By Source Language

The materials are drawn from news sources in three languages: Mandarin Chinese,

American English and Modern Standard Arabic. The Chinese and Arabic materials are

translated into English, so you will be classifying the translations. This is a three way

classification: Mandarin (MAN) vs. English (ENG) vs. Arabic (ARB).

3. By Source News Organization

There are 20 news organizations that contribute to the materials. No news organization

crosses source types or source languages so the previous two classifications may be

helpful here. E.g. The New York Times is an English Newswire organization; it will

never appear as the source of Mandarin text or BN. This is a 20-way classification. See

Appendix A for a listing of news organizations along with their source type and

language.

4. By Broad Topic

The data has been manually annotated for broad class of topics comprising general topics

like “Accidents” or “Sports News”. There are thirteen broad; some helpful information

about the classes can be found in Appendix B. Bear in mind, not every story is annotated

for topic. You are only asked to classify those that are. Therefore, only construct feature

vectors for a subset of the stories. You are only required to classify broad topics. You

may not use the narrow topic labels referring to specific events (for example, “Deadly

Fire in Bangladeshi Garment Factory”) as features!

You are to use the Machine Learning toolkit *weka* in order to run your classification

experiments. To this end, one part of your submission will be a program that generates

weka *.arff* formatted files. As discussed in class and in the *weka* documentation, these

files describe your data set as a series of class-labeled feature vectors. Your program

should read the data set and produce one .arff file for each classification, for a total of 4

files. The feature set that you extract for use in these classification experiments is

completely up to you; however, obviously, you must not use any of the document labels

(<SOURCE\_TYPE>, <SOURCE\_LANG>, <DOC\_DATE>, <NARROW\_TOPIC> etc.)

as features in your feature vector. You may extract different features for different

classification tasks, but you are not required to. You should try at least three different

classification algorithms for each task so you can see how they operate on different tasks.

For these classification experiments you should use 10-fold cross-validation. It is

essential that you use the *weka* package that can be found at /home/cs4705/bin/weka.jar to

run your experiments. If you do not, there is no guarantee that it will be possible to

evaluate your final models.

You must also export the model that yielded the best results for each task, and submit it

along with your feature extractor code – if you do not, evaluating your submission will be

impossible. Also, it is essential that you indicate the classifier and parameters that

generated the submitted model.

You may find that you want to use features that are calculated relative to the entire data

set. For example, “does this story have more or less words that the average story in the

training data?” These types of features can be very useful. However, you need to be

careful when using them in a cross-validation setting. These features should never be

calculated using any testing material. This may force you to run the cross validation

evaluation “manually”. That is, randomly dividing the training data into training and

testing sets for feature extraction *and* evaluation. For your submission you may build a

model on your entire training set.

For some of the classifications (Source Type, Source Language, Source News

Organization) every story in a document will have the same class. However, the

classification should still operate on the story level, not the document level. Therefore, it

might make sense for every story from a document to have an identical feature vector.

Submission:

Your submission should require as little human interaction as possible to test; therefore

you MUST follow these instructions:

In your submission you must include the following files generated by your system:

1) sourceType.arff and sourceType.model

2) sourceLanguage.arff and sourceLanguage.model

3) sourceNO.arff and sourceNO.model

4) topicBroad.arff and topicBroad.model

The following are crucial scripts:

1) Submit one script to compile your code: make.sh

2) Submit four additional scripts, one for each classification task. Each of these scripts

generates an arff file and runs weka on a given a directory that contains the input files.

These scripts will be used to to test your models on unseen data, for example:

./runSourceType.sh sourceType.model /home/nlp/hw2-testfiles

=> It will extract features from all \*.input files in /home/nlp/hw2-testfiles => generates

sourceTypeTest.arff file

This script will also run weka using sourceType.model and sourceTypeTest.arff ==>

weka result report. To get these results from command line you can use the following:

java -Xmx1G -cp /home/cs4705/bin/weka.jar weka.classifiers.trees.J48 –l sourceType.model -T sourceType.arff

(assuming that J48 algorithm was used when you built your model)

2) ./runSourceLang.sh sourceLang.model /home/nlp/hw2-testfiles

…

3) ./runSourceNOLang.sh sourceNO.model /home/nlp/hw2-testfiles

…

4) ./runTopicBroad.sh topicBroad.model /home/nlp/hw2-testfiles

…

You must also produce a write-up of your experiments. This write-up should describe

your experiments and also must include a discussion of the processes you took and the

results you obtained. Some questions that should be addressed can be found in the

grading section below. This write-up should definitely include the cross-validation result

report of the experiments you ran. Make your discussion empirical rather than

impressionistic (i.e. refer to specific results/statistics, performance comparisons)

whenever possible.

Materials:

You are provided with 3,626 files to develop your classifiers. This data set can be found

at: /home/cs4705/corpora/tdt4/ The dataset is made up of a mix of broadcast news transcripts

and newswire digests (a set of newswire stories concatenated together). These files will

have class information embedded in them that you will need to extract in order to

construct the .arff files for training and classification. The file format is as follows:

<DOC>

<SOURCE\_TYPE> {BN|NWIRE} </SOURCE\_TYPE>

<SOURCE\_LANG> {ARB|ENG|MAN} </SOURCE\_LANG>

<SOURCE\_ORG> {News org. identifier from Appendix A.}

</SOURCE\_ORG>

<DOC\_DATE> YYYYMMDD </DOC\_DATE>

<BROAD\_TOPIC> {Broad Topic Index from Appendix B.} </BROAD\_TOPIC>

<NARROW\_TOPIC> {Narrow Topic Index from Appendix B.}

</NARROW\_TOPIC>

<TEXT>

all text news material.

</TEXT>

</DOC>

<DOC>

...as above ...

</DOC>

If there ever is any material outside <DOC> tags, this can be ignored. Additionally, text

information outside of <TEXT> tags can also be disregarded. The class information can

identified from within the corresponding tags in the training material.

Source Type <SOURCE\_TYPE>

Source Language <SOURCE\_LANG>

Source News Organization <SOURCE\_ORG>

Broad Topic <BROAD\_TOPIC>

Narrow Topic <NARROW\_TOPIC>

Requirements:

Note: same language restrictions hold as in HW 1.

Functionality (25pts)

o Does the feature extractor compile?

o Does the feature extractor produce well-formed *arff* files?

o Did you include trained model files for each classification task?

o Submit any supporting scripts.

o How much do they limit the required human interaction?

Results (25pts)

o How well does the submission classify the supplied training data?

o Document Type (Broadcast News or Newswire)

o Source Language

o Source News Organization

o Major Topic

o How well does the submission classify the unseen testing data?

Write-up (25pts)

o Include the cross-validation accuracy for each experiment

o Which classifiers did you use on each of the tasks? Why?

o Which were the fastest? Most accurate? Easiest to use?

o Which do you prefer and why?

o Which classification task was the easiest/hardest? Why?

o Within tasks, were some classes easier to classify than others? (NB: Examine

the weka output for this information.) Why?

o Which classifications were the most similar/most different? Why?

o What features did you use? Why?

o Did they perform better or worse than expected?

o Did early experiments guide your thinking for your final submission? How?

o Which features were the most/least useful? Why?

o If you used any external resources, which did you use? How did they

contribute to the success of your submission?

Documentation (15pts)

o README file: This must include the following.

How to compile the feature extractor (if necessary)

How to run the feature extractor

What features are extracted? How, in broad strokes?

Which submitted model corresponds to which classification task?

Which machine learning algorithm (and parameters) generated the

model?

Any particular successes or limitations of the submission should be

highlighted here.

o Within-Code Documentation

Every method should be documented.

Coding Practices (10pts)

Informative method/variable names

Efficient implementation

Programmer, Memory, and Processor efficiency – don't sacrifice one

unless another is improved

Extra Credit may be awarded for particularly inventive or successful

approaches to the assignment.

Academic Integrity:

Copying or paraphrasing someone's work (code included), or permitting your own work

to be copied or paraphrased, even if only in part, is not allowed, and will result in an

automatic grade of 0 for the entire assignment or exam in which the copying or

paraphrasing was done. Your grade should reflect your own work. If you believe you are

going to have trouble completing an assignment, please talk to the instructor or TA in

advance of the due date.

(Appendices on the following pages)