CIS*4500 (Fall 2019) Machine Learning for Natural Language Processing

Requirements for the Implementation Project

Topic Selection:

For the implementation project, you should focus on one problem in natural language processing and select three different techniques from existing machine learning packages so that you can do a comparative study on these techniques for the related problem. What's described in the following is a default topic. If you are interested in a different topic, you should do some literature search first and then consult with the instructor so that we can make sure that the topic you selected is feasible for implementation and comparable to the default topic in terms of scope and complexity. *You can drop by during the instructor's office hours or email him for an appointment as soon as you can and no later than Nov. 7, 2019 by 4:30 pm. Beyond that time, we will assume that you are going to work on the default topic*.

Default Topic: Sentiment Analysis of Movie Reviews

(1) The problem: Sentiment analysis automatically classifies a review document into positive or negative sentiment (thumbs up or thumbs down). For the default topic, we are going to use the movie review dataset released at Cornell University in 2004 for sentiment analysis. A copy of the dataset named "review_polarity.tar.gz" is available in the "Assignments and Project" module of the "Content" page in our CourseLink account along with a research paper by Pang, et al. (2002) "SentimentAnalysis2002.pdf" that provides a useful background about the dataset and the related problem using machine learning techniques.

(2) Data Analysis: The movie review dataset is balanced in that it consists of 1,000 positive reviews and 1,000 negative reviews. Some data preprocessing is already done, including sentence splitting, tokenization, and case normalization. Based on the input, you can do some data analysis like we did in Assignment One so that we will know the major characteristics of the dataset.

(3) Feature selection: For a classification task, it's often useful to identify a subset of discriminative features so that we can stretch the differences between the related classes. For sentiment analysis, POS-tagging can also be helpful for selecting features since sentiment-expressing words are mostly made of adjectives and adverbs, and sometimes verbs and nouns. For the comparative study, you should try two different feature selection methods and different feature sizes at 500, 1000, 1500, 2000, 2500, and 3000.

(4) Classification techniques: choose three different machine learning techniques for the comparative study. You can use either Java with the Weka package or Python with the Scikit-Learn package for your implementation (see "WekaPackage.pdf" and "PythonPackages.pdf" in the "Assignments and Project" module on CourseLink for how to get started with them). Please note that the Weka package can be run independently with its own interfaces, but for this project, we want to import the related packages and

integrate their functions into your own Java applications. That way, we can run such applications directly without relying on Weka's built-in interfaces.

(5) Model tuning/optimization: The movie review dataset is already given in a random order; so you can easily apply the cross-validation along with a development subset to optimize the model parameters. We recommend using the top 15% as the development set, and the remaining 85% for a five-fold or ten-fold cross-validation. For each feature selection method, feature size, and model combination, we will use the development set to optimize the training process so that the parameter values that lead to the best performance on the development set can be computed (called the best models), which are then evaluated on the test folds to generate the average performance in terms of F-measures. Furthermore, for sentiment analysis, the F-measures are normally computed individually for both positive and negative classes, and the average of the two will be used to evaluate the final performance.

(6) Experimental results and analysis: For the comparative study, we need to compare different machine learning techniques as well as different feature selection methods along with different feature sizes. As a result, you need to gather all the related results from your experiments, and then summarize the results in tables or figures so that we can offer some insights about the differences and similarities. You can refer to the paper by Pang, et al. (2002) as a guide for presenting your results in your project report.

(7) Write a detailed project report using the format suggested in the following section.

Format for the Project Report:

The detailed report should be reasonably complete in describing your project (about 12 pages in length). The recommended format is shown in the following, but you are free to change it based on the need of your project:

- Title page: title, name, student ID, and date.
- Introduction: What's the problem and its major applications? What major techniques have been used for your project? What data sets are used for your experiments and what are the major results?
- Description of the techniques used, including the basic ideas (maybe with examples), all the steps and formulas, and major advantages and disadvantages.
- Highlights of the implementation details: major data structures and modules; assumptions and limitations; and important design decisions.
- Data sets and experiments: explain the data sets, show the experimental results, and discuss/analyze the results.
- Conclusions and possible future improvements.
- Build and installation guide: how to compile the source code and run your system by providing several scenarios to illustrate the process.
- References.

Incremental Plan for Implementation:

Based on the experience of Assignments One and Two, we learned about data preprocessing and analysis, and the use of inverted files as the index. For the project, we will take advantage of the machine learning packages for different classification techniques. So, the main learning curve is to get familiar with these existing packages. Listed below is a recommended incremental development plan that you can try in order to go through the process as smoothly as possible.

(1) Download the dataset and do the data analysis

(2) Implement a very basic system:
   (a) Split the dataset into a validation set (15%), a training set (85x0.80=68%), and a test set (85x0.20=17%)
   (b) Use one feature selection method to select top 1000 and 2000 features, respectively.
   (c) Use one classification method for the 1000 and 2000 features, respectively.
   (d) Evaluate the results for 1000 and 2000 features, respectively with the validation set.
   (e) Evaluate the best combination with the test set to measure the real performance.

(3) Gradually add another feature selection method and two other classification methods and try all combinations with different feature sizes.

(4) Repeat the experiments with different folds using the cross-validation method introduced in the lecture notes.

(5) Gather all the results and write the project report.

Submission Requirements:

A complete submission for the project should include: (1) the source code for your implementation along with any additional packages other than Weka or Scikit-Learn; (2) a readme file that briefly describes the organization and each individual file, as well as the build-and-test instructions; (3) all the data files used in your experiments; and (4) the detailed project report. Tar and gzip all the related files into one compressed file and upload it into the related dropbox in our CourseLink account by the due time. *Please note that the due time for the project submission is Nov. 28, 2019 by 11:59 pm, and in the following day on Nov. 29, we will do one-on-one meetings of about 15 minutes so that you can demonstrate the major features of your implementations.* Several days before the demo date, a shared file will be made available so that you can sign up for the available time slots for your demos.