Natural Language Processing (CS 6120/4120)

Assignment 2

Trees, Augmented Parser, Probabilistic CKY, CKY Algorithm, Sentiment Analysis (Naive Bayes, Multilayer Perceptron)

Instructor: Professor Lu Wang

Due Date: 6pm, October 30th, 2017

For the programming question, you can use Java, Python or C/C++. You also need to include a README file with detailed instructions on how to run the code and commands to facilitate understanding for TAs. Failure to provide a README will result in points deduction. No need to print out the code while submitting the hard copy, just turn it in on Blackboard.

1 Trees (15 points)

Draw tree structures for the following sentences based on Context-Free Grammar.

1. [**3 points**] I will have an ice cream tomorrow.
2. [**3 points**] May I submit the assignment after a week?
3. [**3 points**] The next Red Sox match will be held at the Fenway Stadium
4. [**3 points**] When did Federer win his first Grand Slam?
5. [**3 points**] I dived into a pool without lifejackets

2 Augment Parser (10 points)

Discuss how to augment a parser to deal with input that may be incorrect, for example, containing spelling errors or mistakes arising from automatic speech recognition.

Some way of solutions:

1. One approach is to take the partial syntactic structures that the parser was able to identify and join them together to form full parses.
2. Another approach is to use one of the probabilistic approaches discussed in Chapter 14.

Please write down you solution, and discuss why it would work.

3 Probabilistic CKY (15 points)

Draw probabilistic CKY chart for a sentence **"the train has a bunker"** using the following grammar and their respective probabilities:

1. S → NP VP .80
2. Det → the .50
3. NP → Det N .30
4. Det → a .20
5. V P → V NP .20
6. N → bunker .01
7. V → has .03
8. N → train .02

4 CKY algorithm (10 points)

Sketch how the CKY algorithm would have to be augmented to handle lexicalized probabilities.

5 Sentiment Analysis (50 points)

In following two programming assignments, you will perform sentiment analysis on a popular dataset from `http://www.cs.cornell.edu/people/pabo/movie-review-data/`.

To know more about this movie review dataset and the sentiment analysis problem, read Section 3 and 4 of "Thumbs up? Sentiment Classification using Machine Learning Techniques" by Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan (Proceedings of EMNLP 2002) found at `http://www.cs.cornell.edu/home/llee/papers/sentiment.pdf`.

You have been given the dataset in "hw2-sa-ds.zip". It contains two directories "train" and "test". Each of them contains "pos" and "neg" directories where positive and negative reviews are stored respectively. Each review is a single text file.

5.1 Naive Bayes Classifier [20 points]

Build a Naive Bayes (NB) classifier to predict whether a given document (review) is positive or negative. For Sentiament Analysis, NB classifier can be derived as follows (**eq 6.10** in book - `https://web.stanford.edu/~jurafsky/slp3/6.pdf`). To know more, read **section 6.1** of this chapter as well.

$$c_{NB} = \underset{c \in C}{argmax} \log P(c) + \sum_{i \in positions} \log P(w_i|c)$$

Here, positions refer to all word positions in the document. C is a set of all possible sentiment classes (positive and negative for us). c_{NB} refers to class c chose by NB for given document, where $c \in C$.

Furthermore, for training we can derive $\hat{P}(c)$ and $\hat{P}(w_i, c)$ using following equations **6.11** and **6.12** from book - `https://web.stanford.edu/~jurafsky/slp3/6.pdf`.

$$\hat{P}(c) = \frac{N_c}{N_{doc}}$$

$$\hat{P}(w_i|c) = \frac{count(w_i, c)}{\sum_{w \in V} count(w, c)}$$

V is a union of all word types in all classes. Please read **section 6.2, 6.3** from book - `https://web.stanford.edu/~jurafsky/slp3/6.pdf` - for more details.

1. **[10 points]** Apply this classifier using your favorite smoothing technique for training using given train dataset.
2. **[10 points]** Report evaluation of given test set with Precision, Recall and F1 scores along with the evaluation script.

- You need to submit your code, README.txt file explaining how to run your code, and the output file with predictions based on which you calculated Precision, Recall and F1 scores.

5.2 Multilayer Perceptron [20 points]

- Implement a Multilayer Perceptron (MLP) for sentiment analysis for given train and test dataset using the tool of your choice. Some suggestion are listed below. You can call the training and testing functions for MLP or feedforward neural network from the tool. You do not need to implement the learning (i.e., backpropagation) algorithm.
    - TensorFlow `https://www.tensorflow.org/`
    - theano `http://deeplearning.net/software/theano/`
    - Weka `https://www.cs.waikato.ac.nz/~ml/weka/`
    - scikit-learn `http://scikit-learn.org/stable/index.html`
    - PyTorch `http://pytorch.org/`
- Try four different network structures (e.g. with different number of nodes, layers, or activation functions). It is up to you for the design!
- For each structure, report evaluation of given test set with Precision, Recall and F1 scores. Which one performs the best? Please provide your evaluation script as well.
- You need to submit your code, README.txt file explaining how to run your code, and the output file with predictions based on which you calculated Precision, Recall and F1 scores.

5.3 [10 points]

Compare the above two types of classifiers based on your observations. Show pros and cons of each. For MLP, you can pick the one with best performance.