LING 406: Intro to Computational Linguistics

Spring 2021

Assignment #3: *Part of Speech Tagging*

Issued: Mar. 25, 2021
Due: Apr. 1, 2021 by 11:59PM
Credits: 50 points

This assignment presents you with a POS tagging problem (*a multi-class classification problem*) which will be accomplished with a supervised Machine Learning brute-force approach (i.e., considering a context window of 7 word-tokens: three to the left and 3 to the right of target word to be POS tagged). The data file needed to implement this approach is provided in the assignment folder (pos-eng-5000.data.csv) on github.

All code (using Scikit-Learn or a machine learning library of your choice) must be your own. You will be required to use your toolkit of choice to accomplish the following:

> Estimate the precision, recall, accuracy, and F-measure on the POS tagging task using 5-fold cross-validation. You need to do this for two machine learning models: Naïve Bayes and Decision Tree classifier.
>
> For each classifier compute the contribution of each attribute (this activity is called 'feature contribution') using the "leave one out" approach explained in class. The difference in performance should be captured in a table: as columns you will have the attributes and as rows you have the difference in performance (precision, recall, accuracy, and F-measure): (performance with all the features) - (performance with all the features minus the current feature).

Feature engineering: There are many ways you can represent the context of a target word (feature engineering) and the purpose of this assignment is to give you the opportunity to explore such feature representations of your choice, especially that during the lab sessions, Chase and Hayley, our TAs, took you through the process.

However, as for many tasks we looked at before, you have to start with a baseline system. This should be based on a bag-of-words representation -- a simple set of features to represent the context of a target word (i.e., term). For instance, for each data point, create a dictionary of features describing the context window the term is part of. Then, convert the dictionary features to vectors and use this vector representation to train and test your models. The implementation of your baseline model is worth 20 points.

After this, you are asked to improve over the baseline classifier. You may choose to add two more features to the baseline feature set, or you may want to use a completely different representation of your choice (which might result in a new classifier). This is called the improved classifier [10 points].

Write a report and answer the following questions (be sure to include your table) [5 points each]:

1. Which is the best machine learning model (classifier) for this task (for both the baseline and the improved classifier)? You need to discuss this per metric used to compute the performance.
2. Which features contributed the most to the performance (precision, recall, accuracy, and F-measure)? Which contributed the least? (you need to do this for each machine learning model and for each classifier considered here)
3. How good is this feature set for this task (for each classifier)? (based on the answer at Question 2)
4. If you had more time to work on this problem and do it perhaps more efficiently (in terms of performance), which features/text representation would you choose? (write 1-2 short paragraphs about the features sets you might want to try for this problem).

Extra credit: [15 points]

Train and test the POS tagger (both baseline and improved classifiers) with CRF (Conditional Random Fields) or LSTM (Long Short-Term Memory) model. You have to compare the performance of your system with those obtained with the previous machine learning models.

Deliverables:

- Provide a README.md file including a detailed note (i.e., one paragraph) about the functionality of each of the programs, and complete instructions on how to run them; make sure you include your name in each program and in the README file; make sure all your programs run correctly.
- Provide an answer.pdf file where you should include the results obtained with your table and the answers to the questions outlined above.
- Submit all of your Python code as Jupyter Notebook(s). Your code must be extensively commented using programming best practice.
- Using GitHub add, commit, and push your source code files, the README.md file and the answer.pdf file. 5 points will be deducted if any of these deliverable files is missing.