Homework 1: Part-of-Speech Tagging with HMMs

Due date: March 9^{th} (before the end of the day) with partial delivery on March 7^{th}

1. **Task description**. In this assignment you will implement a first order Hidden Markov Model (HMM) that will perform Part-of-Speech (POS) tagging on code-switched data. The tags correspond to the Universal POS tags. Your system has to be implemented from scratch. The dataset is available in the following Dropbox link: `https://www.dropbox.com/s/cmfltcpt7bbtrw0/dataset.zip`.

2. **Dataset**. The dataset is divided in *train/dev/test* partitions. You will train your HMM model on the *training* set. The *development* set is for you to try your model and iterate over it by tweaking parameters and improving features until you get reasonable results. Once you get the final model, you will have to tag the test set. The data files are in CoNLL format. That is, each line represents a token, a language ID, and a POS tag all separated by tabs. Sentences are separated from each other by empty lines. Note that for the test set we only provide the first two columns, you have to generate the third one. This is an example of the format:

   ```
   sent1word1 eng DET
   sent1word2 eng NOUN


   sent2word1 spa PROPN
   sent2word2 eng VERB
   ```

3. **CodaLab**. We have set up this assignment as a shared task using CodaLab. You will need to create an account in CodaLab and register for our competition (COSC 6336 - Part-of-Speech Tagging) by accepting the Terms and Conditions under the Participate tab. Access to the shared task is through this link. Please check regularly this link as we may change it if we need to modify the competition setup. For successfully completing your submission in the CodaLab platform, you need to submit a text file (submission.txt) with your predictions in a zip file by the submission deadline. Your code must be submitted on the Dropbox link provided in the Deliverables section. The submission.txt file must have the same format of the test set but including the POS tag column as shown in the example above. Keep the same order of lines as well. It is important that the CodaLab zip file only contains the submission.txt file. Do not submit a zip containing a folder with the content within it; this will cause the evaluation script to fail.

   NOTE: CodaLab is an open source framework for running competitions. Your system submissions will be ranked according to accuracy of the system but the ranking will be public and thus it is extremely important that the username you choose for the submission is not disclosing your identity. In order to identify which student gets credit for which system submission, please note in your report, and in your source code, the name you used to identify your submission in CodaLAb (username).

4. **Evaluation**. We will measure your system with the accuracy metric. Your goal is to reach as high an accuracy as possible. State of the art approaches reach ∼97% per token accuracy. Keep in mind that your POS tagger needs to do at least better than the baseline (shown in CodaLab). There are several things you can do to improve your POS tagger but please don't touch the test set until you are done tweaking the model. You can do a trigram HMM, or you can try to add other features on your model.

Deliverables

You will need to turn in the following:

1. Your system predictions in CodaLab by March 7^{th}. As explained above, it has to be a zip only containing the submission.txt file. Do not change the order of the tokens nor the number of lines of the test file.

2. Your source code (properly documented) in this Dropbox link by March 7^{th} before midnight. Submit your code using a zip file and include a README.txt with your username and all scripts needed to run your system. Name your submission as follows: `lastname_COSC6336_assg_1.tar.gz` or `lastname_COSC6336_assg_1.zip`

3. Technical Report (PDF) describing the system and relevant experiments and analysis by March 9^{th}. Submit your report here. This is the most important part and should include an error analysis. A good report will contain some examples of missclassifications together with some possible explanations for them and suggestions on how to fix them. Do not include code on your report, rather a good description of your model, description and motivation of the features chosen, try to add informative plots and figures, and an interesting error analysis.

4. The report needs to follow the guidelines here `https://www.dropbox.com/s/as1tebetiafcc3k/guidelines.pdf`