CS 5322 Spring 2023

Program 2 – Word sense disambiguation (WSD)

Due date: 4/26 (Wed) 1:00pm (see below)

The goal of this program is to apply machine learning and NLP techniques to work on a small scale problem of word sense disambiguation – given a word with multiple word senses, determine which sense of the word is being used for a sentence.

This program is to be done by groups. You are encouraged to do it with your project group. If you want to form other groups that is ok.

Data

You are given three files. Each file corresponds to a word (rubbish, yarn and tissue respectively). For each word, the file listed two senses of each word, used as a noun, as represented by WordNet. Some of these words may have more than 2 senses; those are ignored. Each file contains the word, the gloss of the two senses that are to be disambiguated. Also for each sense, there are 25 sentences that contain the word and has the specified sense (notice that the word may be in plural form).

Stage 1: Beefing up the data set (10 points)

Each group should pick one word, and produce 4 new sentence each for the two senses of that word, that is sufficiently different than the sentences already in the data set. There will be a discussion board on Canvas where you can post the sentences. Every group are allowed to use the new sentences anyway it pleases.

Stage 2: Building a WSD tool for each word (90 points)

You are to build a WSD tool (in python) that take in a sentence with the word involved, and determine which of the two senses are being used in that sentence:

- You should build one tool for each word separately.
- You are welcomed to use any machine learning method, and any text-preprocessing method that is used
- You are NOT allowed to download any existing WSD code and run it as your own.
- If you decide to take the machine learning route and decide to train a model, the training time of your model can be as long as you like 9but have to be done before the due date of your program). However, you need to be able to save your model, and in subsequent runs, load the model directly without having to retraining the model from scratch.
    - The size of the saved model should be reasonable (less than 100MB), and the loading time of the model should also be reasonable (within 5 minutes in Mameframe).
    - However, after training, then given a sentence, your program should be able to very quickly determine the sense of the word given a new sentence to test.
- You should implement a python module called “cs5322s23.py”. The module should have the following functions:
    - WSD_Test_Rubbish(list), where list is a list of string, each string is a sentence that contain the word “rubbish”. You should output a list of numbers, each number

corresponds to the sense of the word “rubbish” for each sentence (either 1 or 2). Notice that you are responsible for the number to match with the corresponding sente4nce, and each number is either 1 or 2 (NOT 0 or 1).

  - WSD_Test_Yarn(list) and WSD_Test_Tissue(list): are defined the same way.
  - You should load your model inside the function. It is OK if you have to reload the model every time the function is called.

What to hand in:

- You should hand in the program that you used to train the machine learning tool.
- You should hand in a version of your saved model (if you decide to take a “unsupervised learning route”, or your method does NOT build any model, you can skip this).
- You should hand in your “cs5322s23.py” module.
- You should hand in a report detailing your method for classification (including all preprocessing steps, machine learning algorithm that you use and the step you do to disambiguate a sentence.

Grading:

- The writeup is worth 15 points
- Each of the three tools (one for each word) is worth 25 points.
  - 15 points for correct implementation
  - You program will be tested with a different test set. The accuracy will be measured.
    - If your accuracy is less than 25%, you get 0 point
    - If your accuracy is between 25-50%, you get 2 points
    - If your accuracy is above 50%, you get 1 point for each 2.5% of accuracy above 50% (so if your accuracy is 70% or better, you get full marks)

Testing procedure:

I will have three test file ready on 4/26 (Wed) at 1:00pm. They will appear on Canvas. Each file will be named “test_<word>.txt”, and will contain 50 sentences to be disambiguated. Once you download the file, you have 10 minutes to submit the result of your classifier. For each word, you should have a file name “result_<word>_<first and last name of one member of your group”.txt” that store the result. Each of your result file should have 50 lines, with each line storing the sense of the corresponding word in each sentence. You are responsible to ensure the order is correct. You need to upload the 3 files before 1:10pm.

If your group have problem making the 1:00pm time slot on 4/26, let me know and we will schedule it sometime BEFORE that date.