ASSIGNMENT 3: NAMED ENTITY RECOGNITION FOR REAL ESTATE TEXT

Motivation: The motivation of this assignment is to get practice with sequence labeling tasks such as Named Entity Recognition. More precisely you will experiment with the BiLSTM-CRF model and various features on real estate text.

Scenario: Different real estate agents share noisy text messages on a real estate platform to inform buyers about new properties available for sale. We will call these text messages, shouts. As a company interested in automating real estate information, our first step is to perform NER on these shouts so that important and relevant information can be extracted downstream.

Problem Statement: The goal of the assignment is to build an NER system for shouts. The input of the code will be a set of tokenized shouts and the output will be a label for each token in the sentence. Labels will be from 8 classes:

Locality (L)

Total Price (P)

Land Area (LA)

Cost per land area (C)

Contact name (N)

Contact telephone (T)

Attributes of the property (A)

Other (O)

Training Data: We are sharing a labeled dataset of shouts. Each shout is divided into words, one per line, each followed by a space and its token label. Blank lines indicate the end of a sentence.

The Task: You need to write a sequence tagger that labels the given shouts in a tokenized test file. The tokenized test file follows the same format as training except that it does not have the final label in the input. Your output should label the test file in the same format as the training data.

First make a sequence tagger by using a BiLSTM in PyTorch. Improve it by implementing a BiLSTM-CRF. In order to improve the tagging accuracy, create additional features that might be useful for the task. For example, you could add features of capitalization and whether the current token is a number, etc.

Here are some suggestions on features:

1. Try features from lower level syntactic processing like POS tagging or shallow chunking. You may need to use Twitter-trained chunkers/taggers. Resources: Twitter NLP at Noah's Ark, and Twitter NLP at Alan Ritter.
2. Define task-specific features such as specific regular expressions indicative of specific types.
3. Use existing gazetteers of locations or bootstrap one. We can promise not to test you on locations outside Delhi NCR area.
4. We also provide a larger unlabeled corpus of shouts. You can train a word2vec model and use embeddings as inputs instead of training them from scratch.
5. You may define word shape features or word substring features.
6. You may use char n-grams.
7. Your idea here…

What to submit?

Submit your best code (best if trained on all training data and not just on a subset) by Tuesday, 7th May 2019, 11:55 PM. The code should not need to train again. You should submit only the testing code, after the models have been trained. That is, you should not need to access the training data anymore.

You will need to submit your submission files in a zip folder in the format **<EntryNo>.zip**. It should produce the following files on unzip in the present working directory:

```
EntryNo
|___compile.sh
|___run.sh
|___Writeup.pdf
```

1. We will first compile your code by running: **./compile.sh**
2. Your code will be run as: **./run.sh inputfile.txt outputfile.txt**

You will be penalized if your submission does not conform to this requirement.

The outputfile.txt should have the same number of lines as inputfile.txt. And it should have two additional characters per token line (space and labeling). Here is a format checker. Make sure your code passes format checker before final submission.

Your code should work on HPC. Since there is not that much time available for grading, we will not do any demos for this assignment.

The writeup.txt should have first line that mentions names of all students you discussed/collaborated with (see guidelines on collaboration vs. cheating on the course home page). If you never discussed the assignment with anyone else say None. After this first line you are welcome to write something about your code, though this is not necessary

Evaluation Criteria

(1) 12.5 points for performance of your code for each NER (including Other). A total of 100.
(2) Bonus points awarded for outstanding performer

What is allowed? What is not?

1. The assignment is to be done individually.
2. You must use PyTorch for this assignment.
3. You must not discuss this assignment with anyone outside the class. **Make sure you mention the names in your write-up in case you discuss with anyone from within the class.** Please read academic integrity guidelines on the course home page and follow them carefully.
4. Feel free to search the Web for papers or other websites describing how to build named entity recognizers. Cite the references in your writeup.
5. We will run plagiarism detection software. Any team found guilty will be awarded a suitable penalty as per IIT rules.
6. Your code will be automatically evaluated. You get significant penalty if it is does not conform to output guidelines. Make sure it satisfies the format checker before you submit.

Disclaimer: The dataset and problem is brought to you courtesy Plabro Networks, a Delhi-based startup. So, this assignment gives you a taste of the real, real world. Kindly refrain from sharing the dataset outside the class.