Assignment 2

Due on April 20, 2020 (23:59:59)

Click here to accept your Assignment 2

Introduction

Named-entity recognition (NER) is a subtask of information extraction that seeks to locate and classify named entity mentioned in unstructured text into pre-defined categories such as person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc.

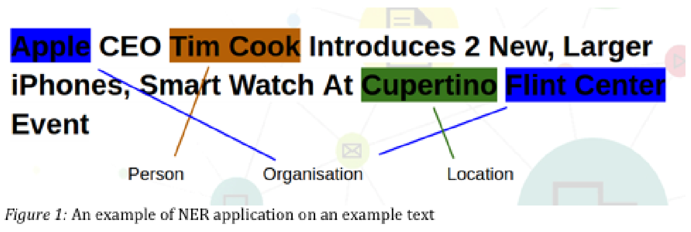
Apple CEO Tim Cook Introduces 2 New, Larger iPhones, Smart Watch At Cupertino Flint Center Event
Person
Organisation
Location
Figure 1: An example of NER application on an example text

Figure 1: An example of NER application on an example text

In this assignment, you will implement a NER tagger using Hidden Markov Models (HMMs). Therefore, you will practice HMMs and Viterbi algorithm in this assignment.

0.1 Task 1: Build a Bigram Hidden Markov Model (HMM)

We need a set of observations and a set of possible hidden states to model any problem using HMMs. As for the NER problem, the observations are the words in a given sentence. The hidden states are the NER tags of the words, which will be predicted by the model.

You will implement a bigram HMM tagger for thisk task. A bigram HMM consists of three components:

1. **Initial probability** $p(t_1)$ is defined as the probability of the tag t_1 of the first word w_1 that a sentence begins with. For example, $p(B-LOC)$ is the probability of a sentence to begin with a B-LOC tag.
2. **Transition probability** $p(t_{i+1}|t_i)$ is defined as the probability of seeing the tag t_{i+1} following the tag t_i. For example, $p(B-LOC|I-LOC)$ is the probability of assigning B-LOC tag to the current word given that the previous tag is I-LOC.
3. **Emission probability** $p(w_i|t_i)$ is the probability of generating the word w_i from the hidden state t_i. For example, $p(school|B-LOC)$ is the probability of observing

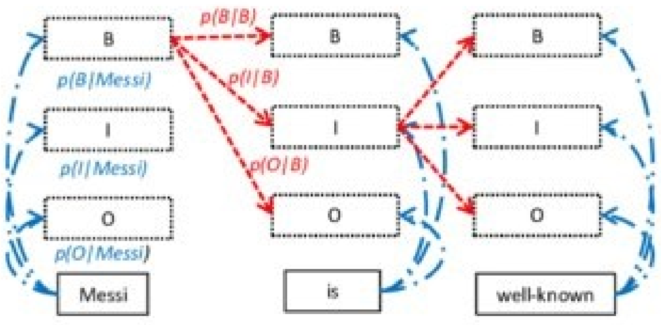

Figure 2: An illustration of the Viterbi algorithm.

the word *school* from the hidden state tagged with B-LOC.

To build your own hidden Markov Model, you must calculate the initial, transition, and emission probabilities by using the given training data. You may use various preprocesssing steps on the dataset (lowercasing the tokens, ...). You can also use various techniques for unknown words. You can try out different methods to improve your model.

0.2 Task 2: Viterbi Algorithm

Once you build your HMM, you will use the model to predict the NER tags in a given raw text that does not have the NER tags. To this end, you will implement the Viterbi algorithm that will assign the most probable NER tag sequence $t_1 \cdots t_m$ for a given a word sequence $w_1 \cdots w_m$ as follows:

$$\underset{t_1 \cdots t_m}{argmax} p(t_1 \cdots t_m | w_1 \cdots w_m) \tag{1}$$

The Viterbi algorithm is performed in two steps:

1. You will compute the probability of the most likely tag sequence.
2. You will trace the back pointers to find the most likely tag sequence from the end of the sentence till the beginning.

An illustration of the Viterbi algorithm is given in Figure 2. Your output format will be the same as the input text format.

0.3 Task 3: Evaluation

Your program will compute the accuracy of the NER tagger. The accuracy is the ratio of the correctly assigned tags to the total number of words in the test data:

$$A(W) = \frac{\#of\_correct\_found\_tags}{\#of\_total\_words} \tag{2}$$

To this end, you will implement a simple HMM class with the following interfaces:

- **dataset():** It takes only one argument: folder path and returns the list of sentences.
- **HMM():** It takes list of sentences and returns HMM Model (initial, transition and emission probabilities)
- **viterbi():** It takes test sentences and and HMM model and returns tagged test sentences
- **accuracy():** It takes gold sequences and predicted sequences and returns the accuracy

Dataset

The dataset is a CoNLL 2003 NER dataset that is composed of sentences with words per line 5 possible labels : "LOC, ORG, MISC, PER, O", that represent the NER classes. The "LOC, ORG, MISC, PER, O" labels are abbreviation of "LOCATION, ORGANIZATİON, MISCELLANEOUS, PERSON, OTHER" respectively. The dataset link

Submit

You are required to submit all your code. You will implement the assignment in **Python** (Python 3.5). You will submit a report in latex format template). The codes you will submit should be well commented. Your report should be self-contained and should contain a brief overview of the problem and the details of your implemented solution. Give the answers of all questions raised in the definition of the assignment above. You can include pseudocode or figures to highlight or clarify certain aspects of your solution.

- report.pdf
- code/ (directory containing all your codes as Python file .py)

CHALLENGE (For Bonus Points)

In addition to the default tasks defined above, you can improve your NER tagger more to compete with other participants in the class. The accuracy of your NER tagger is the main criteria to be compared with others. The participants in the class that obtain a higher tagging accuracy will deserve more points.

To enter the competition, you have to register kaggle with your department email account. The webpage of the challenge will be announced later. Top 5 assignments will earn extra points (10 points).

Grading

- Code (90 points): Task 1: 35, Task 2: 50, Task 3: 5
- Report (10 points)

Note: Preparing a good report is important as well as the correctness of your solutions!

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.