Slot Tagging of Natural Language Utterances

The goal of this homework is to train a model to tag slots in natural language utterances of users addressing a virtual personal assistant. The tagged slots and the associated values are traditionally used to accomplish user requests. For example, if a user asks to learn about movies of a specific director, we would need to identify the name of the director (in addition to the user’s intention to find out about movies, as we did in homework 1), and issue a query to the backend knowledge graph to get information for formulating a response back to the user.

You will be using the same dataset as in homework 1, but your labels, this time, will be the slots for each token in the input utterance. For example:

show me movies directed by Woody Allen recently.

There are two slots in this utterance:

1. director = “Woody Allen”
2. release\_year = “recently”

The correct labels for this utterence would then be:

show me movies directed by Woody Allen recently.

O O O O O B\_director I\_director B\_release\_year

Note that the start of a slot has tags prefixed with B (e.g., B\_director) and the interior of slots has tags prefixed with I (e.g., I\_director).

The goal is to use PyTorch to develop and train your own deep neural network models and tag the sequential data using the training examples. You will have to try different techniques to improve upon a baseline model (which you need to define in your report). Here are some techniques you may try:

* Multilayer Perceptron (MLP)
* Recurrent networks (Elman RNN, LSTM, GRU)
* CNNs
* Different core features such as bag-of-words, word embeddings (random or pretrained)
* Additional information derived from the text
* Different loss functions
* Different optimizers such as SGD, AdaGrad, or Adam
* Preventing overfitting via dropout or weight-decay/regularization

To train a robust model, you will need to compare the performance of different models to identify their suitability. Test out key hyper-parameters and find what works best. Look at how similar problems are solved in research papers and try to implement those approaches. It may require some ingenuity in choosing features and developing/picking the right models.

Remember, each token in the input corresponds to one label in the output. Your model should be predicting exactly the right number of output labels, and if it isn’t, it may indicate a bug in your architecture.

Dataset

This dataset is generated based on film schema of Freebase knowledge graph.

[Link to hw2 filesLinks to an external site.](https://drive.google.com/drive/folders/1pBXAY5qpKDbmdjDr4xUENPUpPlvVc9Xg?usp=sharing)

There are three CSV files.

hw2\_train.csv The file has three columns:

1. ID: the id for each row
2. UTTERANCE: the natural language text from you will extract relations
3. IOB Slot Tags: The targeted sequence of IOB tags corresponding to the utterance.

hw2\_test.csv

This file is similar to hw2\_train.csv, but only contains the ID and UTTERANCE columns. Your model will predict the IOB Slot Tags.

Evaluation

Evaluated based on the F1-score of recognized entites.