SIMON FRASER UNIVERSITY

Assignment 2 - Phrasal Chunking

Thudercats Deluxe - Nelson Nauata & Jonathan Smith

October 11, 2016

1 THE BASELINE

The baseline algorithm consists of a perceptron learning approach that provides a sequence of chunk tags for each input sentence. The features are extracted locally for each word in a left-to-right fashion and for each feature in our feature vector we define a single weight. In this perceptron model the weights are updated interacting with the training data, which corresponds to labeled words as shown in the assignment description. So each sentence in our train set is predicted using the Viterbi algorithm to compute the argmax over the trained weights outputting the sequence of chunk tags. These predictions are then compared to the target labels and a delta is added or discounted from the corresponding weights that led to these predictions.

2 BATCH LEARNING AND AVERAGED PERCEPTRON

The final model is trained using mini-batches of sentences, which interact with the perceptron elements and provide the required information for performing the updates for the weights. Worth mentioning that the size of the mini-batches were chosen by testing many possible values on the validation set and we picked the one that provided the best results. At the end of the interaction with one mini-batch the updates from each sentence within the mini-batch are averaged and added to the weights.

3 EARLY STOPPING

The early stopping consists of a technique to perform validation efficiently and store the best model on the validation set. For implementing this we used the module subprocess from python to call the scripts **perc.py** and **score-chunks.py** from inside the training routine and obtain the score for the weights after seen N mini-batches. The score obtained from **score-chunks.py** was then compared to the best score obtained so far from previous iterations and the best model was stored.

4 Trigram Features

5 Results

In this section we present the results obtained for the model after including each new feature. Also, the score using for this comparison was obtained from **score-chunks.py**.

Model	Score
Baseline	91.00
Minibatch + Average	93.85
Minibatch + Average + Trigram	X