CS6501-005 Natural Language Processing Homework 04-05 Name: Kai Qu Computing ID: kq4ff Date: 12/6/2019

1 Word Embeddings:

1. With the default parameters of the logistic regression and mapping the words that do not exist in the pretrained embeddings to '<unk>', I get the 54.24% as the accuracy for the development set.

2. With default parameters of the logistic regression, along with the concatenation of the sentence embedding and feature vectors by CountVectorizer with the default parameters as the input, I get 61.40% as the accuracy for the development set.

3. From previous the homework, the best hyperparameters I get for CountVectorizer are lowercase=True, min_df=7e-5, ngram_range=(1,2), max_features=10000. I try the following parameters and get the results as below :

C	Solver	Development Set Accuracy
0.05	lbfgs	64.92%
0.01	lbfgs	64.98%
0.001	lbfgs	58.92%
0.1	lbfgs	64.44%

Therefore, the best hyperparameters are C = 0.01 with Solver = lbfgs.

2 Recurrent Neural Network Language Models:

1. Data Preprocessing: In order to reduce the number of categories and lower the perplexity, I map the words whose frequencies are less than 5 to 'UNK'. After doing such reduction, the number of categories is reduced from 23747 to 20378.

2. After 10 epochs of training, with batch_size = 1, learning_rate = 0.5, gradient_clipping = 1, I get the perplexity on the training set as 398.871, and the perplexity on the development set as 388.061.

3. The value of n in the better model is 3. After 10 epochs of training, with batch_size = 1, learning_rate = 0.5, gradient_clipping = 1, the perplexity on the training set based on the better model is 360.83. The perplexity on the development set based on the better model is 344.863.

4. The optimization methods I use in the better model is stochastic gradient descend, with learning rate = 0.5, gradient clipping = 0.25, and momentum = 0.8. The perplexity on the training set is 149.557, and the perplexity on the development set is 233.674.

5. The input/hidden dimensions used in the better model is 256. With learning_rate=0.5, batch_size= 1, gradient_clip=1, the perplexity on the training set based on the better model is 145.729. The perplexity on the development set is 140.729.

6. The difference of mini-batches size makes a slight difference with regard to the computed perplexity. With the model defined in the simple-rnnlm.py, the

mini-batch size associated with the lowest perplexity in the training set and development set is 16. The perplexity on the training set is 373.399, and the perplexity on the development set is 362.341.