

000  
001  
002  
003  
004  
005  
006  
007  
008  
009  
010  
011  
012

# E1246 - Natural Language Understanding

## Assignment2 : Language Models

050  
051  
052  
053  
054  
055  
056  
057  
058  
059  
060  
061  
062  
063  
064  
065  
066  
067  
068  
069  
070  
071  
072  
073  
074  
075  
076  
077  
078  
079  
080  
081  
082  
083  
084  
085  
086  
087  
088  
089  
090  
091  
092  
093  
094  
095  
096  
097  
098  
099

005  
006  
007  
008  
009  
010  
011  
012  
013  
014  
015  
016  
017  
018  
019  
020  
021  
022  
023  
024  
025  
026  
027  
028  
029  
030  
031  
032  
033  
034  
035  
036  
037  
038  
039  
040  
041  
042  
043  
044  
045  
046  
047  
048  
049

Tapan Bhardwaj(14704)

tapanb@iisc.ac.in

### Abstract

In Assignment-1 I had developed a four-gram based language model on **Brown** and **Gutenberg** corpus and then calculated perplexity on both the corpus using 90-10 split. This assignment Assignment is also related to the development of Language model only, but this time, it is implemented using Long Short Term Memory (LSTM) Neural Network, a variant of Recurrent Neural Network (RNN).

## 1 Introduction

Current assignment is divided into three tasks which are mentioned below:

### 1.1 Task 1

This task is to implement token level LSTM-based language model using Gutenberg Corpus (D2) in following settings:

- S2: Train: D2-Train, Test: D2-Test

### 1.2 Task 2

This task is to implement character level LSTM-based language model using the same settings as mentioned in Task-1.

### 1.3 Task 3

Using the best model implemented in **Task 1** and **Task 2**, we need to generate a sentence of 10 tokens by running a script with the name generate-sentence.sh.

## 2 Dataset

Gutenberg Corpus is chosen for the development of both Language Models. Due to limitations of the computation resources, only 40 percent of the corpus is used. The division of the corpus (from 40 percent) is given below:

- Training : 90
- Testing : 10

## 3 Baseline Language Model

N-Gram Language Model of Assignment-1 is used as a baseline for **Task 1** and **Task 2**. Baseline model is run against the new corpus splits and results are used to compare the outputs of the **Task 1** and **Task 2** models.

## 4 Model Description

### 4.1 Character level LSTM Model

I have used single layer of 128 LSTM units and then dense layer of softmax is used to predict the probability of next character. Used 8 characters as input and predicted the next character probability and thus calculated the loss based on this probability.

### 4.2 Word level LSTM Model

I have used single layer of 128 LSTM units and then dense layer of softmax is used to predict the probability of next word. Model is also learning word embeddings during training and dimension for each word embedding used is 50. Model takes sequence of 20 words as a input and predict the probability of next word and thus loss is calculated based on the predicted probabilities.

## 5 Language Model

LSTM Model is developed using Keras library which facilitates different variants. In absence of GPUs, model is configured with CPU only. In addition, model is using the keras state saver utility, to save model parameters on each epoch. Basic understanding of the LSTM RNN is understood from blogs of Chris Olah, Andrej Karpathy and Goldberg Book(Goldberg, S, 2017).

100	<b>5.1 Preprocessing</b>	150
101	• Raw files of gutenberg corpus are concatenated and the text is changed to lower case.	151
102		152
103	• Punctuations are removed from raw text.	153
104		154
105	<b>5.2 Generating Sentences</b>	155
106	• Picking a seeding sequence of word or characters from training data uniformly at random.	156
107		157
108	• Generating a probability distribution of next word or character from the trained model	158
109		159
110	• Using numpy multinomial method to pick next word or character which follows above mentioned probability distribution.	160
111		161
112		162
113		163
114		164
115		165
116	<b>6 Result</b>	166
117		167
118	<b>6.1 Word-Level Language Model Results:</b>	168
119	perplexity = 63.6	169
120		170
121	<b>6.2 Character-Level Language Model Results:</b>	171
122	perplexity = 78.9	172
123		173
124	<b>6.3 Four Gram Language Model Results:</b>	174
125	perplexity = 110.6	175
126		176
127	<b>6.4 Some examples of sentences generated from Character Language Model:</b>	177
128	• but you mean what she had been the blain to	178
129		179
130	• ended the think her own while they were so anxious	180
131		181
132	• any pleasure to a little infime resolved to be made	182
133		183
134	• it was a great dear miss bates that i had been	184
135		185
136	• so much belong to her friends when he was all the convenience of her	186
137		187
138		188
139		189
140	<b>7 Acuuracy/Measures</b>	190
141		191
142	<b>7.1 Task 1</b>	192
143	Perplexity is used as the measure for this task.	193
144		194
145	<b>7.2 Task 2</b>	195
146	Human Evaluation.	196
147		197
148	<b>8 Github Link</b>	198
149	<a href="https://github.com/TapanBhardwaj/NLP-projects">https://github.com/TapanBhardwaj/NLP-projects</a>	199