

000  
001  
002  
003  
004  
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

# E1246 - Natural Language Understanding

## Assignment1 : 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

Tapan Bhardwaj(14704)  
tapanb@iisc.ac.in

### Abstract

To designed a language model on **Brown(D1)** corpus and **Gutenberg(D2)** corpus using ngrams. Following are two tasks that we performed -

- **Task 1:** Divide both dataset into train, dev, and test and build the best LM in the following settings and evaluate.
  - **S1:** Train: D1-Train, Test: D1-Test
  - **S2:** Train: D2-Train, Test: D2-Test
  - **S3:** Train: D1-Train + D2-Train, Test: D1-Test
  - **S4:** Train: D1-Train + D2-Train, Test: D2-Test
- **Task 2:** Generate few sentences of 10 tokens.

## 1 Implementation

### 1.1 Language Model

I have implemented four language models for given datasets. Initially I have added "UNK" word to training corpus to make it closed vocabulary.

- First model is simple unigram back-off model. In this model we look for probability of unigram and if unigram is absent in training corpus we assign unigram probability of "UNK" word.
- Second model is simple bigram back-off model. In this model we look for probability of bigram and if it is absent we look for probability of unigram and if unigram is absent in train we assign unigram probability of "UNK" word.

- Third model is simple trigram back-off model. In this model we look for probability of trigram and if it is absent we look for probability of bi-gram and if it is absent then it look for the probability of it's uni-gram and if unigram is absent in train we assign unigram probability of "UNK" word.
- Fourth model is simple fourgram back-off model. In this model we look for probability of fourgram and if it is absent we do recursively like above.

### 1.2 Random Sentence Generation

For sentence generation i have used four-gram model. I have assumed three starting symbol(\*) to be present and then choosing a word according to four-gram probability distribution from the list of all words that are possible(i.e. words that follow these two words in training set) and repeating the same process to generate the token of required length.

## 2 Result

### 2.1 Task 1

For Simple unigram-backoff - Train: 90per Test: 10per  
For Simple bigram-backoff - Train: 90per Test: 10per  
For Simple trigram-backoff - Train: 90per Test: 10per  
For Simple fourgram-backoff - Train: 90per Test: 10per

100	<b>S1 : train = D1 train and test = D1 test</b>		150
101	Simple uni-gram backoff	1308.280	151
102	Simple bi-gram backoff	330.399	152
103	Simple tri-gram backoff	249.650	153
104	Simple four-gram backoff	237.871	154
105	<b>S2 : train = D2 train and test = D2 test</b>		155
106	Simple uni-gram backoff	753.285	156
107	Simple bi-gram backoff	155.347	157
108	Simple tri-gram backoff	97.505	158
109	Simple four-gram backoff	83.415	159
110	<b>S3 : train = D1 train+D2 train and test = D1 test</b>		160
111	Simple uni-gram backoff	1639.992	161
112	Simple bi-gram backoff	420.491	162
113	Simple tri-gram backoff	307.129	163
114	Simple four-gram backoff	287.738	164
115	<b>S4 : train = D1 train+D2 train and test = D2 test</b>		165
116	Simple uni-gram backoff	789.880	166
117	Simple bi-gram backoff	157.184	167
118	Simple tri-gram backoff	96.357	168
119	Simple four-gram backoff	81.269	169
120	<b>2.2 Task 2</b>		170
121	Some examples of token generated from Language Model:		171
122	<ul style="list-style-type: none"> <li>• The funeral for my husband was just what Letch deserved very</li> <li>• The operations of its other plant in Rochdale and Leesona's</li> <li>• It is recognized that all the obviously non-propagandistic aspects of</li> <li>• Outside the hall I anxiously looked around for the bubbles</li> </ul>		172
123	<b>3 Accuracy/Measures</b>		173
124	<b>3.1 Task 1</b>		174
125	Perplexity is used as the measure for this task. (All values in Result section table contain perplexity value)		175
126	<b>3.2 Task 2</b>		176
127	Human Evaluation.		177
128	<b>4 Git Hub Link</b>		178
129	<a href="https://github.com/TapanBhardwaj/NLP-projects/">https://github.com/TapanBhardwaj/NLP-projects/</a>		179
130			180
131			181
132			182
133			183
134			184
135			185
136			186
137			187
138			188
139			189
140			190
141			191
142			192
143			193
144			194
145			195
146			196
147			197
148			198
149			199