E1246 - Natural Language Understanding
Assignment1 : Language Models

Rupali Dakhane (SR Number 14601)

Abstract

In this assignment I have experimented with various language models on **D1: Brown D2: Gutenberg** .
The best language model is obtained for the given datasets. The tasks performed are:

- Splitting datasets into train, dev and test and building best language model using the evaluation criterion of perplexity.
  - 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
- Generating a sentence of 10 tokens using the built language model

1 Text Preprocessing

The following tasks are performed on text obtained from Gutenberg and Brown Corpus

- Word Tokenisation
- Removal of Punctuations

2 Handling Unknown Words

Unknown words are handled using absolute discounting method. In which I subtracted 0.75 from each word count and then this count was given to unseen words of test data.

3 Implementation

I have tried following experiments with corpus-

$$P_{bo}(w_i \mid w_{i-n+1} \cdots w_{i-1}) = \begin{cases} d_{w_{i-n+1} \cdots w_i} \dfrac{C(w_{i-n+1} \cdots w_{i-1} w_i)}{C(w_{i-n+1} \cdots w_{i-1})} & \text{if } C(w_{i-n+1} \cdots w_i) > k \\ \alpha_{w_{i-n+1} \cdots w_{i-1}} P_{bo}(w_i \mid w_{i-n+2} \cdots w_{i-1}) & \text{otherwise} \end{cases}$$

where

$C(x)$ = number of times x appears in training

w_i = ith word in the given context

Figure 1: Kartz Backoff

$$\beta_{w_{i-n+1} \cdots w_{i-1}} = 1 - \sum_{\{w_i : C(w_{i-n+1} \cdots w_i) > k\}} d_{w_{i-n+1} \cdots w_i} \frac{C(w_{i-n+1} \cdots w_{i-1} w_i)}{C(w_{i-n+1} \cdots w_{i-1})}$$

Figure 2: Value of Beta for Kartz Backoff

$$\alpha_{w_{i-n+1} \cdots w_{i-1}} = \frac{\beta_{w_{i-n+1} \cdots w_{i-1}}}{\sum_{\{w_i : C(w_{i-n+1} \cdots w_i) \le k\}} P_{bo}(w_i \mid w_{i-n+2} \cdots w_{i-1})}$$

Figure 3: Value of Alpha for Kartz Backoff

- Stupid Backoff with absolute discounting on **trigram** model
  - α = 0.4 is used in figure 1
- Kartz Backoff with absolute discounting on **trigram** model
  - To compute α, the value of β is calculated as given in figure 2, And figure 3 gives equation to compute α

4 Perplexity

I have used **perplexity** as metric which is given by following equation.

$$Perplexity(C) = \sqrt[N]{\frac{1}{P(s_1, s_2, ..., s_m)}}$$

$$P(s_1, ..., s_m) = \prod_{i=1}^{m} p(s_i)$$

$$Perplexity(C) = \sqrt[N]{\frac{1}{\prod_{i=1}^{m} p(s_i)}}$$
$$= 2^{\log_2 [\prod_{i=1}^{m} p(s_i)]^{-N}}$$
$$= 2^{-\frac{1}{N} \log_2 [\prod_{i=1}^{m} p(s_i)]}$$
$$= 2^{-\frac{1}{N} \sum_{i=1}^{m} \log_2 p(s_i)}$$

5 Result

S1 : train = D1 train and test = D1 test	
Stupid Backoff	334.50
Karts Backoff	85.16
S2 : train = D2 train and test = D2 test	
Stupid Backoff	350.34
Karts Backoff	123.65
S3 : train = D1 train+D2 train and test = D1 test	
Stupid Backoff	200.95
Karts Backoff	90.30
S4 : train = D1 train+D2 train and test = D2 test	
Stupid Backoff	385.86
Karts Backoff	85.74

6 Sentence Genration

I have used **Shannon method** for generating sentences

- Generate random sentences:
- Choose a random bigram “<s> w” according to its probability
- Now choose a random bigram “w x” according to its probability
- And so on until we randomly choose “</s>”
- Then string the words together
- <s> I
  I want
  want to
  to eat
  eat Chinese
  Chinese food
  food </s>

7 Git Link

https://github.com/Rupali0408/NLU_Assignment_1