Assignment Report

Praveen Gupta
Computer Science and Automation
Indian Institute Of Science
praveengupta@iisc.ac.in

Abstract

Build a language model on **Brown**(D1) corpus and **Gutenberg**(D2) corpus. And then perform following two task:

- **Task 1:** Divide both dataset into train, dev, and test and build the language model and evaluate the perplexity in the following settings:-
  - **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 a sentence of 10 tokens.

1 Preprocessing

I divide the both datasets D1 and D2 in test,dev and train in the ratio of 8:1:1. Then i remove the punctuations except from full stop and comma. Now i split datasets into sentences and append each sentence with two START symbol in beginning and one STOP symbol at the end. I generated unigrams from the train set and find its conditional distribution. After this i replace all the single appeared word in unigram with UNK symbol. This i done to handle out of vocabulary words in test data Now i took test set and replace all those words by UNK symbol which has never appeared in train set.

2 Implementation

I have implemented two language models for given datasets.

2.1 Stupid Back Off

In this model. Now i generate bigrams and trigrams from the train set and find their conditional frequency distribution. Then for every word of every sentence in test set if it present in trigram i find the its probability of trigram otherwise i go for bigram if it is also not present there then go for unigram.

2.2 Intepolation

In this model i took dev set and find hyper parameters for each of the setting. Then i use those hyper parameters to find the perplexity on test set.

2.3 Sentence Generation

For sentence generation i took two START symbol as initializers then find best words corresponding to these using trigram. Every time i generate a word, if new trigram is not present in my distribution the i use bigram for generation and if also not present in bigram then i use unigram.

3 Result

3.1 Task 1

Train:Dev:Test = 8:1:1

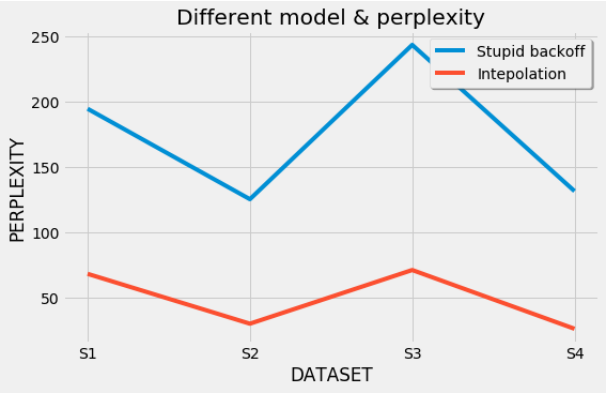

Figure 1: Model Vs Perplexity

S1 : train = D1 train and test = D1 test	
Stupid back off using trigram	194.95
Interpolation of unigram,bigram and trigram	68.59
S2 : train = D2 train and test = D2 test	
Stupid back off using trigram	125.73
Interpolation of unigram,bigram and trigram	30.41
S3 : train = D1 train+D2 train and test = D1 test	
Stupid back off using trigram	243.87
Interpolation of unigram,bigram and trigram	71.51
S4 : train = D1 train+D2 train and test = D2 test	
Stupid back off using trigram	131.91
Interpolation of unigram,bigram and trigram	26.52

Graph plotted for each setting in the two case which clearly shows that interpolation model perform better than stupid back off in every setting.

3.2 Task 2

Some examples of token generated from Language Model:

- straight or to witness the hysterical agitations of his very
- shall furnish to the millionaire ireton todd is entertaining in
- still done with expecting any course must have heard your

4 Acuuracy/Measures

4.1 Task 1

Perplexity is used as the measure for this task. (All values in Result section table contain perplexity value)

4.2 Task 2

Human Evaluation.