Salman Anwaar
I18-1410
Assignment 1

Introduction

A statistical language model is a probability distribution over sequences of words. an **n-gram** is a contiguous sequence of **n** items from a given sample of text or speech. Or in simple words N-grams are simply all combinations of adjacent words or letters of length *n* that you can find in your source text.

The goal of this assignment was to find unigram, bigram, trigram, backward bi gram and bi-directional bi gram from the given corpus and then apply this n-grams to make a statistical model to generate a poetry that end with rhyming word.

Dataset Details

The dataset in json format include text of poetry with each gid to uniquely identify one poetry. There were total of 3085117 verses in training data. Whereas there were only 61 verses in testing data.

Work Done

First, we have separated ngram (unigram, bigram, trigram) words from the corpus. The following results are obtained.

Model	**Count**
Unigram Words	379059
Bigram Words	3950188
Trigram Words	9152318
Backward Words	3950079
Bi-Directional Bigram Words	7197052

Then we have computed probabilities of each n gram models using the following formula.

Model	**Count**
Unigram Words	$P(wi)$
Bigram Words	$P(wi\|wi\text{-}1)$
Trigram Words	$P(wi\|wi\text{-}1wi\text{-}2)$
Backward Words	$P(wi\text{-}1\|wi)$
Bi-Directional Bigram Words	$P(wi\|wi\text{-}1) + P(wi\text{-}1\|wi)$

Then after finding the probabilities we generate stanza of poetry using the n gram models one by one. The rhyming end of stanza was done by a python library pronunciation, given an input it returns all the rhyme of the words.

. Stanza was generated by the following technique

- Load the Gutenberg Poetry Corpus
- Tokenize the corpus in order to split it into a list of words
- Generate ngram models
- For each of the four stanzas
  - For each verse
    - Generate a random number between 5 and 10
    - Select first word
    - Select subsequent words until end of verse
    - If not the first verse, make sure the last word rhymes with the previous verse
- Print verse

The Output of the Bi gram model was

```
thy gaunt hungry hawks conceit im grab
that elysium gone home shall blast stab
are stiffer every cycle are creeping rust should resign your
not seye and julep would lack means honest living or
```

The Output of the Tri gram model was.

```
cockadoodles curses can patient to north vaporous and
craftily to from his quivering that tiny worthi
displeasing unto hat speaking conscience
intercession of manhoods standing passion
```

Results

Perplexity is a measurement of how well probability model predicts a sample. A low perplexity indicates the probability distribution is good at predicting the sample. Perplexity was calculated for test dataset. Results are as follow

Model	**Perplexity Results**
Bigram Words	0.9810643849879754
Trigram Words	0.9811659439338061
Backward Words	0.9745617040607808
Bi-Directional Bigram Words	0.9625825478631995

Conclusion

In the end I would say that this is the very basic language model used in text mining, but it's the core of processing text.

References:

[1] https://en.wikipedia.org/wiki/N-gram

[2] https://en.wikipedia.org/wiki/Perplexity