Answers - CS 545, Fall 2012, Homework 3

Kumaresh Visakan Murugan
mvisakan@cs.wisc.edu

1. Answer to Question 1

- The unigram, bigram and trigram models are implemented and trained using the training data. The cross entropy and perplexity on both the training and test set are obtained as below:

Unigram model

Data Set	Cross Entropy	Perplexity
Training Set	8.8436334228	459.4083991399
Test Set	9.8050854962	894.5916586641

Bigram model

Data Set	Cross Entropy	Perplexity
Training Set	4.8282423580	28.4083347015
Test Set	10.6672662283	1626.1743496037

Trigram model

Data Set	Cross Entropy	Perplexity
Training Set	2.1127575514	4.3251721367
Test Set	12.1313606192	4486.4562246066

As we move from unigram to bigram to trigram model the cross entropy and perplexity of the training set decreases and hence there is improvement from unigram to bigram to trigram in the training set. But the cross entropy and perplexity of the test set increases as we move from unigram to bigram to trigram.

We notice improvement in the training set as we move from unigram to bigram to trigram model because we are able to fit the training data set correctly as we increase the n grams. We notice an increase in cross entropy and perplexity in test set as we move across higher n grams model because, higher the n is, more data is needed to train the model to correctly fit any test set. The probability of the n gram word which appears in the training set to appear in the test set is quite lower if the training set is small.

- An interpolated language model is implemented with probability function as,

$$P_{int} = l_1 P_{uni}(w_3) + l_2 P_{bi}(w_3 \mid w_2) + l_3 P_{tri}(w_3 \mid w_1, w_2)$$

Where $l_1 = 1/3$, $l_2 = 1/3$ and $l_3 = 1/3$

The cross entropy and perplexity of this model is obtained as:

Interpolated model

Data Set	Cross Entropy	Perplexity
Training Set	3.2011040322	9.1966219254
Test Set	7.8716052670	234.2012989160

The interpolated language model shows an improved result when compared with the previous un-interpolated models if we consider the cross entropy and perplexity of both the training and test set.

We can further improve the interpolated model by changing the values of l_1, l_2, l_3.

For improving the cross entropy and perplexity of the train set, consider the values

l_1 = 0.001,

l_2 = 0.001,

l_3 = 0.998

The values obtained are :

Data Set	Cross Entropy	Perplexity
Training Set	2.1149370972528594	4.331711311639885
Test Set	11.162063088801997	2291.478586688283

We can also improve the cross entropy and perplexity of the test set considering the values,

l_1 = 0.3840000000000003,

l_2 = 0.3840000000000003,

l_3 = 0.2319999999999994

The values obtained are :

Data Set	Cross Entropy	Perplexity
Training Set	3.506990703966780	11.368663091015772
Test Set	7.852420929844638	231.10760581366597

- The sentences are generated using the various language models and a few samples are as follows :

  Unigram Model :

  1. broke , were share because questions . in the and of his . abruptly , , . , of rhyme is moment he as what <S>
  2. instinctive fed to - , period , on and life and . of <S>
  3. , wish , he let the , look why . of or winston by <S>

  Bigram Model :

  1. good party prisoners one person spoke his mother 's shop could not see . <S>
  2. he had been developed , with the kettle and when he felt towards an aeroplane as one side of his body down a habit of paper on a piece of one another sweltering summer air of the bench , tapped him , and paraded through the last copy of kilometres over her first led back into his penholder and a word , physicist , and was merely destroy the forced-labour camp along the girl hopped over winston , easily encircled by the infallibility with the arm . <S>
  3. the scarlet banners , he had got out of the two might complicate the knocking of beauty and grabbed up immediately . <S>

  Trigram Model :

  1. the phrase generally used - had burst in every line of trucks which travelled over a cliff solves nothing . <S>
  2. not here , she felt in the assumption that every word , but he knew it had been taken off his cap and address him as very important or interesting . <S>
  3. it was inextricably mixed up with fear and anger automatically . <S>

  Interpolated Model :

  1. the old-fashioned clock told him i tell canteen and with eurasia pushing him rumours circulated , photographs - child 's life when month of in control matter momentary contact with difficulty winston and began its strange cylindrical hats still <S>
  2. life just as kick them innocent man to be three thousand years ago . <S>
  3. in 'll twenty years at one time behind . <S>

  We could find the language model from the sentences in case of un-interpolated models, but it is difficult to identify the interpolated model on looking at the sentence as it may contain a valid n gram phrase as it was generated by applying a probability distribution on top of the previous un-interpolated models.

Answer to Bonus Question :

The trigram model will have trigrams with noun phrases with high probability. As these noun phrases could easily be extended with other noun phrases, it may result in very long sentences with irrelevant noun phrases.

2. Answer to Question 2 :

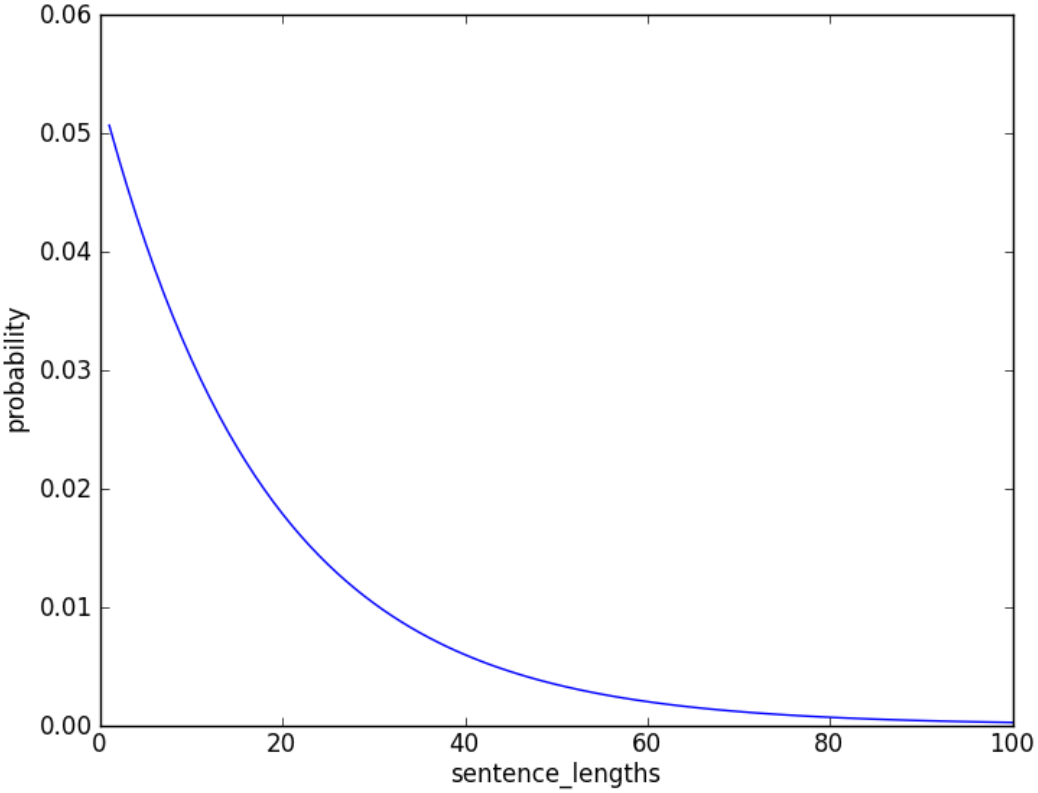
Geometric Distribution model :

$$P_{geom}(l;p) = p^{l}(1-p)$$

Where 1 – p = unigram probability of stop word,

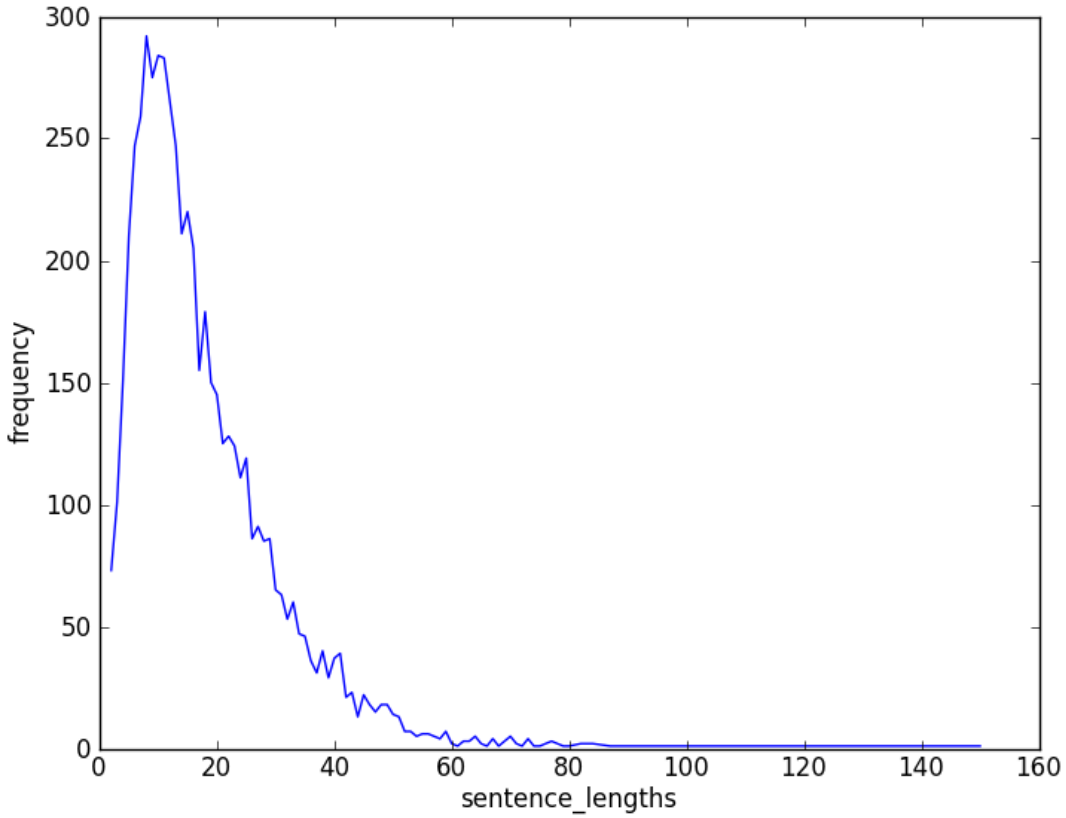
p = sum of unigram probabilities of all words except stop word

- Plotting the probability of sentence lengths 1 through 100, the plot obtained is,



2.1 Probability of sentence lengths 1 through 100 using Geometric Model

- Plotting the frequency of the actual sentence lengths appearing in orwell-train.txt,



2.2 Frequency of actual sentence lengths in orwell-train.txt

The plot of probability of sentence lengths 1 through 100 and the plot of frequency of the actual sentence lengths in orwell-train.txt decrease in a similar fashion after a particular initial value. Both the plots differ initially as the frequency of very small length sentences are very less in the orwell-train.txt, but the geometric distribution depicts high probability for sentences of small length. But for sentences of increasing length both the graphs depicts a low probability and low frequency respectively.

- The cross entropy and perplexity of the test-set sentence lengths are as follows:

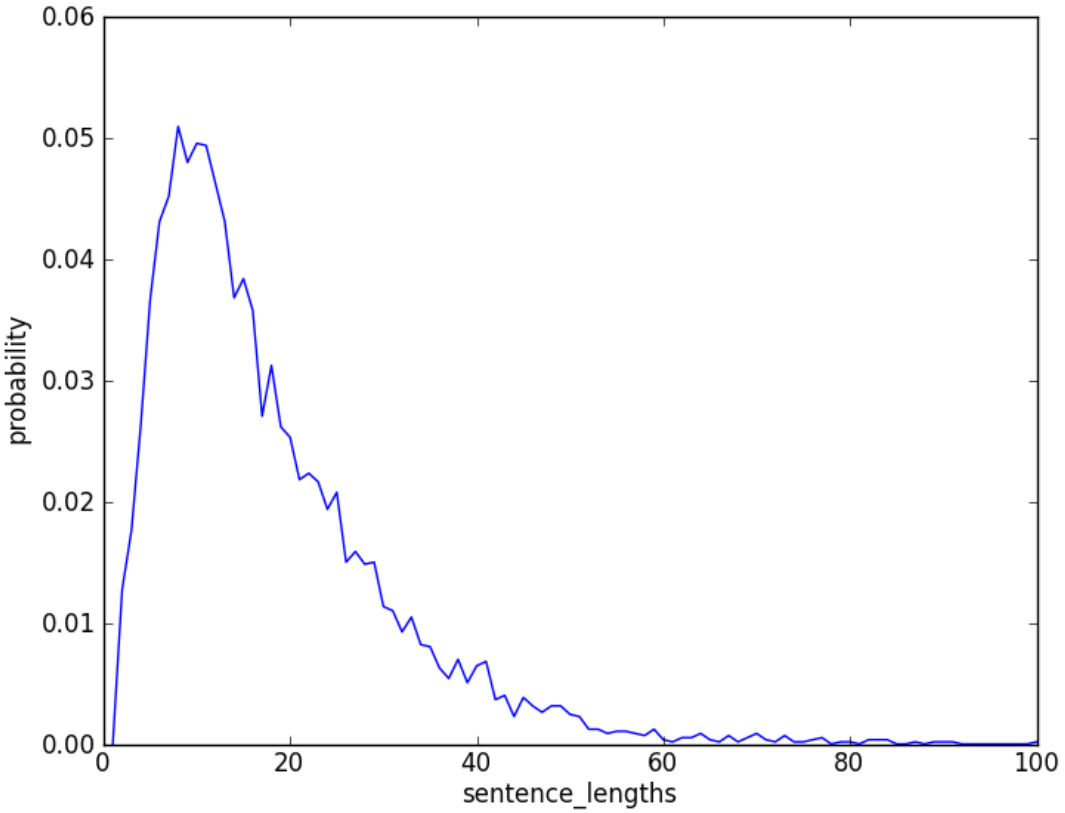
Data Set	**Cross Entropy**	**Perplexity**
Test Set	5.5614789986	47.2250035068

Multinomial Distribution Model:

$$P_{mult}(l) = count(l) / N$$

Where count(l) is the number of sentences of length l in the training data and N is the total number of sentences in the training data.

- The multinomial model is implemented and trained using the training data. Now the plot of probability of sentence lengths 1 through 100 is,



2.3 Probability of sentence lengths 1 through 100 using Multinomial Model

Now this plot looks **similar** to the plot of frequency of actual sentence lengths in the orwell-train.txt.

- The cross entropy and perplexity of the test set lengths is now obtained as,

Data Set	**Cross Entropy**	**Perplexity**
Test Set	5.2392656423	37.7725333732

The multinomial distribution model gives a slightly lower values for cross entropy and perplexity when compared with the previous geometric distribution model. Still the values look almost similar to each other as they do not differ significantly.

Negative Binomial Distribution Model :

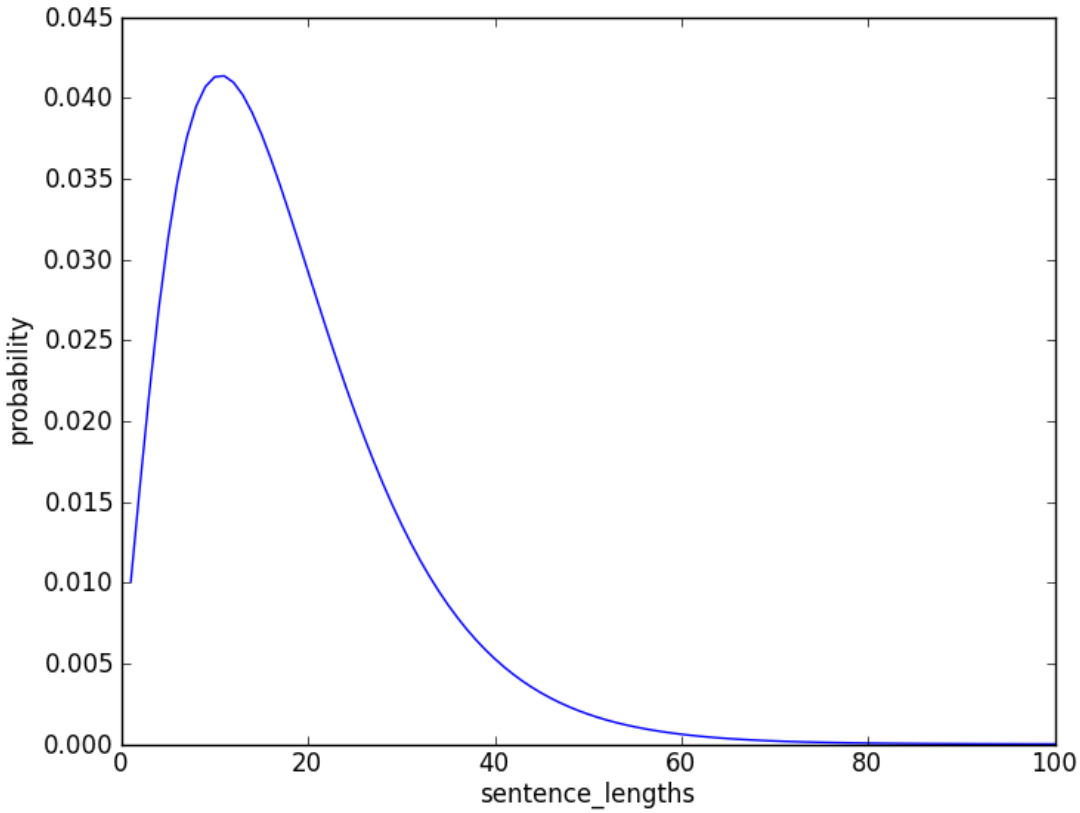
$$P_{nbin} = (\Gamma(r + l) / (\Gamma(r) \Gamma(l + 1))) p^{l} (1 - p)^{r}$$

Where Γ is the gamma function,

p = 0.8682

r = 2.6878

- Now the probability of sentence lengths 1 through 100 looks like:



2.4 Probability of sentence lengths 1 through 100

Now this plot looks **similar** to the plot of frequency of actual sentence lengths in orwell-train.txt. This plot is almost like an **approximation over the previous plot** obtaining a smooth plot.

- Now the cross entropy and perplexity of the test set sentence lengths looks like,

Data Set	**Cross Entropy**	**Perplexity**
Test Set	5.2783035090	38.8085737594

Though the cross entropy and perplexity of all the three models discussed are almost in the same range, the cross entropy and perplexity of the negative binomial distribution model looks very much similar to multinomial distribution model than geometric distribution model.

Scripts used to generate the models and data :

Config.py : has the configurable paramaters used by the language models and data manipulation.

languageModels.py : has all the models implemented with functionalities for calculating cross entropy and perplexity.

loggerUtils.py : has logging related functionalities.

orwellDataParser.py : has functionalities for parsing orwell-train.txt and orwell-test.txt

Question1.py : Used the functionalities and models defined to analyse the models related to answering question 1.

Question2.py : Used the functionalities and models defined to analyse the models related to answering question 2.

Plots saved :

Geometric_Probability_sentence_lengths_1_100.png : Plot of probability of sentence lengths 1 through 100 using geometric distribution model.

Frequency_sentence_lengths_orwell_train.png : Plot of frequency of actual sentence lengths in orwell-train.txt.

Multinomial_Probability_sentence_lengths_1_100.png : Plot of probability of sentence lengths 1 through 100 using multinomial distribution model.

Negative_Binomial_Sentence_lengths_1_100.png : Plot of probability of sentence lengths 1 through 100 using negative binomial distribution model.