COMP40020 Assignment 3
Name: Bhavesh Kumar
Student No. 16203173

Task 1:

Following are the observations after building language model using the Sphinx Knowledge base tool.

As per the requirement mentioned for the task, I have used **100west.txt** file to upload to the Sphinx server and compiled and download the generated language model packaged in tar file. Since data has been already preprocessed already I have not written code to preprocess explicitly to bring one sentence per line.

According to the Compilation report, there are 257 sentences and 3042 unique words exist in the file (100west.txt). Also, the dataset has 1377 Unigram, 2829 Bigram and 2984 Trigram in total.

Sphinx generated 5 files which are as follows:

1. 1777.dic
   This is Pronunciation Dictionary file which provides a mapping of vocabulary words to sequences of phonemes.

   For example:

   "COTTAGES" K AA T IH JH IH Z

   Here, Cottages is a word which is followed by pronunciations of it.

2. 1777.lm
   This is Language Model file which generates N-grams of the dataset. It calculates a total number of words and a possible pair of words for 1-grams, 2-grams, 3-grams and assigns log probability and backoff score to the individual word for 1-grams, 2-grams and 3-grams.

   There are a few important concepts while building a language model which are explained below:
   a.) Discount mass
   Sphinx language modeling tool uses quick lm statistical language modeler to create a language model. It was designed to work on very little data which make modeling system unreliable, to solve this issue and make model trustable [1]. To over this issue of having less dataset, it assigns a uniform ratio discounting factor to all ngrams to make it not depend on the frequency of words in the dataset. In another word, it is kind of giving lower preferences when using lower level ngrams.
   In this model (1777.lm), **discount mass 0.50** has been used.
   b.) Backoffs and log probability
   The idea of using Backoffs in the model building is that go for smaller N-Grams if not found in higher order N-Grams. In another word, if there is not a perfect match in Trigram, user Bigram and even if it is not found in Bigram then use Unigram form of the word. So when using lower level N-Grams if a particular word is available in higher order N-Grams there is a penalty associated with it which is called Backoff.

   For example:
   Let's see N-Grams generated for a word "**ALWAYS**" by the lm tool mentioned above.

1-grams
-3.8520 ALWAYS -0.2970

2-grams
-0.3010 ALWAYS IN -0.2977
-0.3010 TRACKS, ALWAYS 0.0000

3-grams
-0.3010 ALWAYS IN CONTACT
-0.3010 RAILROAD TRACKS, ALWAYS
-0.3010 TRACKS, ALWAYS IN

When we look at this example above, it can be clearly seen that Trigram has covered pretty much all the occurrences for a word "ALWAYS" with log probability of a sequence of words "ALWAYS IN CONTACT" is -0.3010 and similarly for remaining two sequences of words. Now when we see Bigram for the same word there is a backoff score -0.2977 associated with one of the pair "ALWAYS IN" and log probability is -0.0310 but there is no backoffs for using pair of words "TRACKS, ALWAYS" because there is almost 50% probability that this combination will come together in sentences so using 2-grams for this combination doesn't penalize the prediction model. And at the last if we can see 1-grams representation of the same word "ALWAYS" there is very high backoffs score (-0.2970) associated with it because there are better combinations possible in 2-grams and 3-grams for this word and also it has very low log probability (-3.8520) because there is rare chance that this particular word will occur alone in the sentences and hence a better pair will always be there for it in higher order N-grams.

3. 1777.log_pronounce
   This file contains Letter-to-Sound (LtoS) pronunciation for each individual word in the dataset.

   For example:
   COTTAGES - Morpheme: COT AGE ES

   In above example, "COT", "AGE" and "ES" are the morphemes for word "COTTAGES"

4. 1777.sent
   Language builder appends start <s> and end </s> string tag to every line in the dataset and saves in this file to use it further in language model building tasks.

   For example:
   <s> NORTH OF 53. </s>
   Above line has been generated for this sequence of words – "NORTH OF 53."

5. 1777.vocab
   This file contains a list of words used in the dataset.

Task 2:

My surname starts with letter ‘K’ and in the generated pronunciation dictionary file using lm tool, there are only six words so I chose to go with letter ‘L’ for Task 2 solution.

Here is the list of 10 words starting with letter ’L’ and their syllables from 1777.dic file.

LACK L AE K

LAKES L EY K S

LAND L AE N D

LANDING L AE N D IH NG

LARGE L AA R JH

LAST L AE S T

LAWN L AO N

LEARN L ER N

LIFE L AY F

LIKE L AY K

Transition diagram (Fig: 1) for a finite state transducer (FST) which models above syllables:

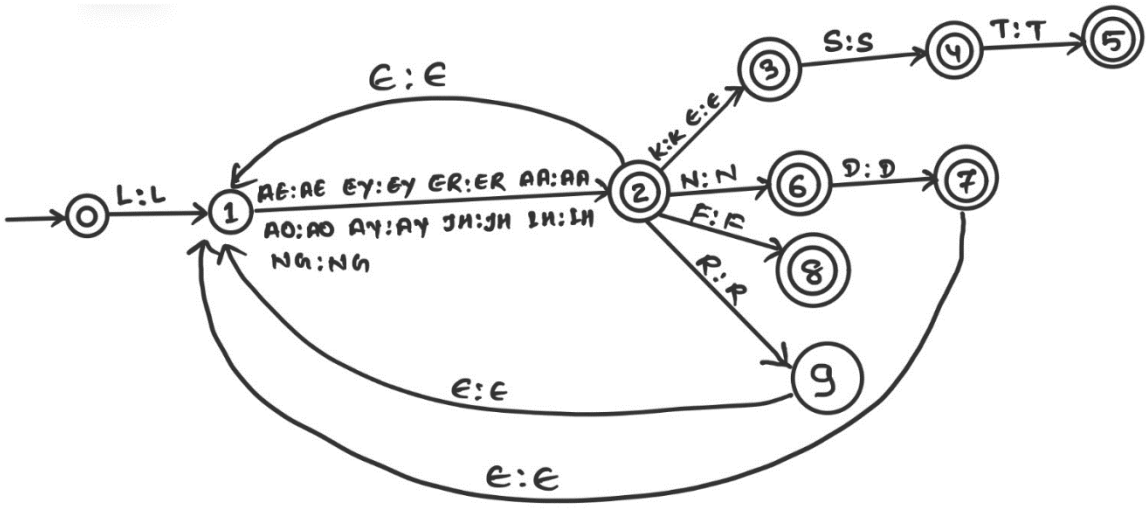

Fig: 1

I have uploaded a file with only syllables of 10 words shown above and I have copied the output of .lm file below.

Output:

The (fixed) discount mass is 0.5. The backoffs are computed using the ratio method.
This model based on a corpus of 10 sentences and 19 words

\data\
ngram 1=19
ngram 2=30
ngram 3=31

\1-grams:
-1.0569 </s> -0.3010
-1.0569 <s> -0.2612
-2.0569 AA -0.2972
-1.4548 AE -0.2653
-2.0569 AO -0.2855
-1.7559 AY -0.2855
-1.7559 D -0.2570
-2.0569 ER -0.2855
-2.0569 EY -0.2894
-2.0569 F -0.2612
-2.0569 IH -0.2972
-2.0569 JH -0.2612
-1.5798 K -0.2527
-1.0569 L -0.2612
-1.4548 N -0.2527
-2.0569 NG -0.2612
-2.0569 R -0.2972
-1.7559 S -0.2570
-2.0569 T -0.2612

\2-grams:
-0.3010 <s> L 0.0000
-0.3010 AA R 0.0000
-0.9031 AE K -0.1249
-0.6021 AE N -0.1761
-0.9031 AE S -0.1761
-0.3010 AO N -0.1761
-0.6021 AY F 0.0000
-0.6021 AY K -0.1249
-0.6021 D </s> -0.3010
-0.6021 D IH 0.0000

-0.3010 ER N -0.1761
-0.3010 EY K -0.2218
-0.3010 F </s> -0.3010
-0.3010 IH NG 0.0000
-0.3010 JH </s> -0.3010
-0.4771 K </s> -0.3010
-0.7782 K S -0.1761
-1.3010 L AA 0.0000
-0.6990 L AE 0.0000
-1.3010 L AO 0.0000
-1.0000 L AY 0.0000
-1.3010 L ER 0.0000
-1.3010 L EY 0.0000
-0.6021 N </s> -0.3010
-0.6021 N D 0.0000
-0.3010 NG </s> -0.3010
-0.3010 R JH 0.0000
-0.6021 S </s> -0.3010
-0.6021 S T 0.0000
-0.3010 T </s> -0.3010

\3-grams:
-1.3010 <s> L AA
-0.6990 <s> L AE
-1.3010 <s> L AO
-1.0000 <s> L AY
-1.3010 <s> L ER
-1.3010 <s> L EY
-0.3010 AA R JH
-0.3010 AE K </s>
-0.3010 AE N D
-0.3010 AE S T
-0.3010 AO N </s>
-0.3010 AY F </s>
-0.3010 AY K </s>
-0.3010 D IH NG
-0.3010 ER N </s>
-0.3010 EY K S
-0.3010 IH NG </s>
-0.3010 K S </s>
-0.3010 L AA R
-0.9031 L AE K
-0.6021 L AE N
-0.9031 L AE S

-0.3010 L AO N
-0.6021 L AY F
-0.6021 L AY K
-0.3010 L ER N
-0.3010 L EY K
-0.6021 N D </s>
-0.6021 N D IH
-0.3010 R JH </s>
-0.3010 S T </s>

\end\

From this output, it can be clearly seen that syllables if use as unigram then there is higher backoffs associated with them but when we see Bigrams model, there are 14 pairs of syllables where backoffs score in 0 that means these are the best combination where 2-grams form can be used without compromising the backoffs.

For example:
1-grams:
-2.0569 AA -0.2972
-2.0569 R -0.2972

2-grams:
-0.3010 AA R 0.0000
-0.6021 AY F 0.0000

3-grams:
-1.3010 <s> L AA
-0.6990 <s> L AE
-0.3010 D IH NG

In 1-grams, If syllable ''AA" is used alone then there is backoff -0.2972 but when used with another syllable such as "R", there is 0 backoff score for this combination. As the fundamental concept of the N-grams that higher the N-grams better the meaning of the sentence, and this can be validated using Finite State Transducer drawn (Fig: 1) above. If we see 3-grams "<s> L AA" and run these sequence of syllables through FST, it follows maximum path than Unigram and Bigrams and reaches closer to the expected syllables faster than the remaining N-grams. Differences between FST and language model generated for syllables is that FST gives the path to traverse and check for acceptance and rejection of a word where Language model generates the probability of words or sequence of words for N-grams and assigns the backoff score.

References

[1] "Building a language model," Carnegie Mellon University, [Online]. Available: https://cmusphinx.github.io/wiki/tutoriallm/. [Accessed April 2018].