ECE 289 Probability and Statistics for Data Science

Final Project

N-gram mixture model for word suggestion

Jose Joy

A53230620

Table of Contents

Project Description

Word suggestion is the task to predicting next likely words, given some words. Applications of this include sentence completion, text suggestion in messaging services, code completion in Integrated Development Environments (IDE) etc. In this project, an n-gram mixture model was created to accomplish this task.

An n-gram is a contiguous sequence of n items from a sequence of text or speech. An n-gram model is a type of probabilistic language model for predicting the next item in such a sequence in the form of a Markov model, with n-1 states. Benefits of n-gram models include simplicity and scalability – with larger n, a model can store more context with a well-understood space–time tradeoff, enabling small experiments to scale up efficiently.

The model makes use of a combination of unigrams (single words), bigrams (2 word combinations) and trigrams (3 word combinations) to predict the next likely word to come after a sequence of words. This mixture model makes weighted predictions based on three individual models (unigrams, bigrams and trigrams respectively) to suggest next likely word.

For this project, I collected text data from the following open sources

1. UCI Machine Learning Repository (Sentence Classification Data Set)
2. Reuters-21578 Text Categorization Collection
3. Ana Cardoso Cachopo's Homepage - http://ana.cachopo.org/
4. Peter norvig's website - http://norvig.com/
5. Brown corpus - http://www.sls.hawaii.edu/bley-vroman/brown_corpus.html

Collating from these sources resulted in a text corpus of about eleven million words. The text files obtained were parsed in python to convert all words to lower case, remove punctuations etc. Each individual word, two word sequences (bigrams) and three word sequences (trigrams) were stored in three different lists. Three separate dictionaries were also created to store the counts of each unique n-gram.

Total Number of words parsed = 10,950,156

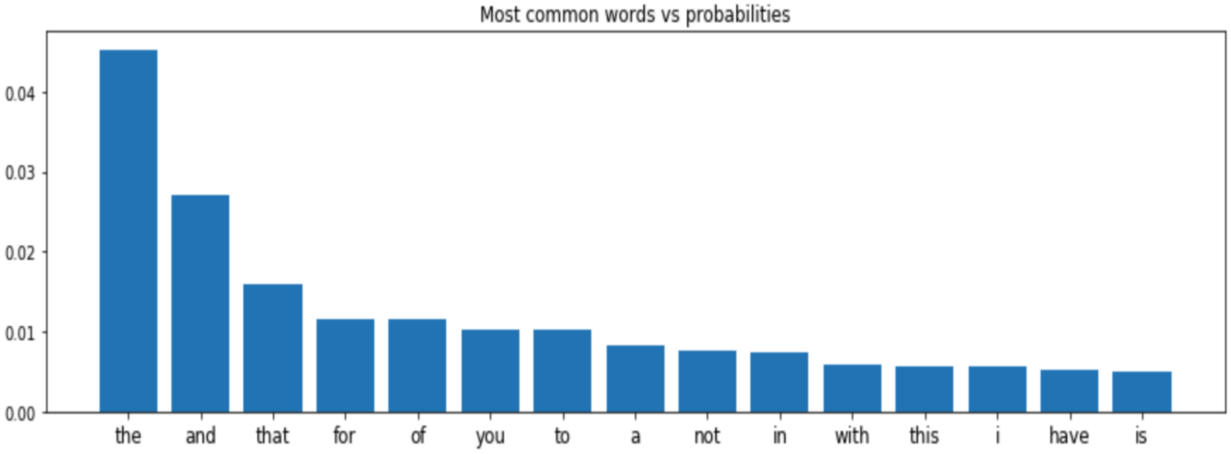
Data properties

Number of unigrams = 165,211

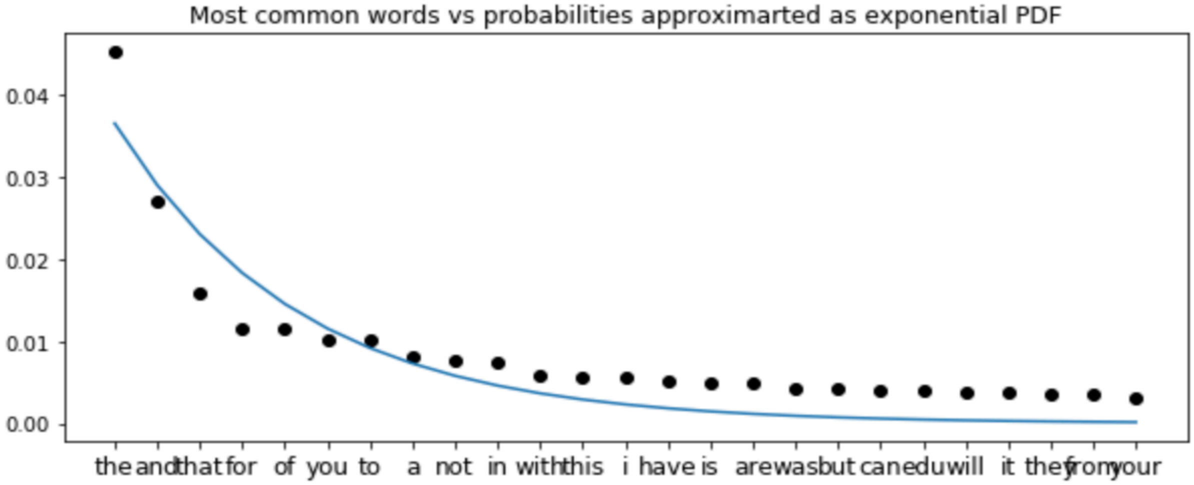
Number of bigrams = 2,988,017

Number of trigrams = 6,515,17

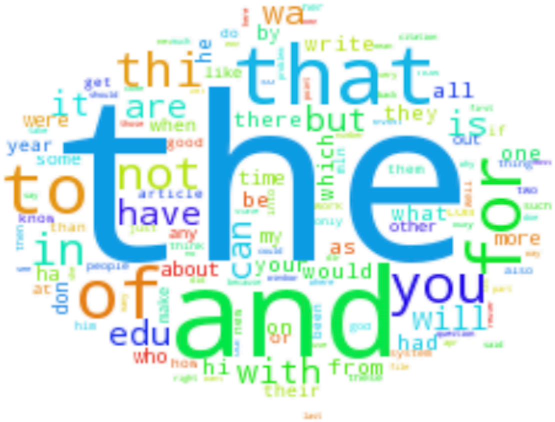
The most common unigrams and their probabilities (word count/ total number of words) was seen as follows

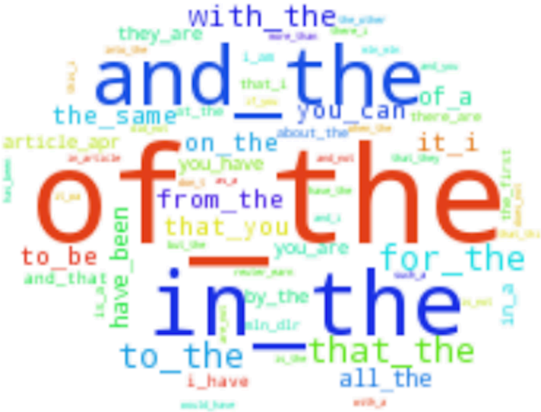


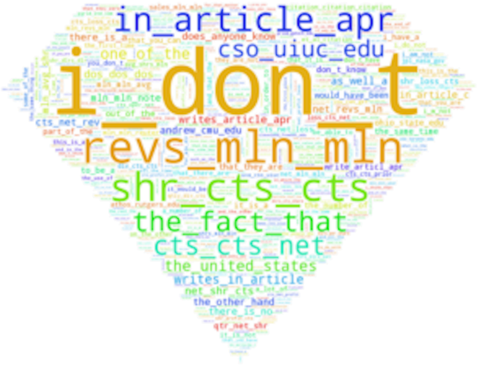
'the' is the most commonly occurring word, followed by 'and' and 'that'. This distribution was similar to an exponential distribution, with lamda = 0.0366



Word clouds (an image composed of words used in a particular text or subject, in which the size of each word indicates its frequency or importance.) were drawn for each n-grams, (unigrams, bigrams and trigrams respectively)



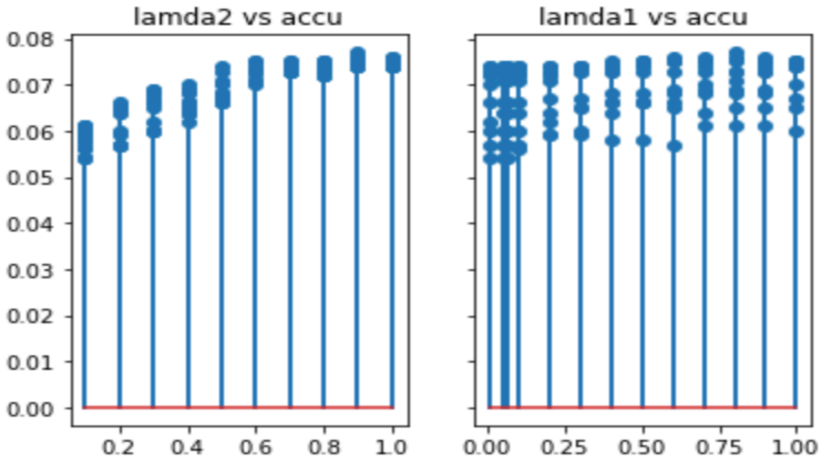




Training

The mixture model works by running three separate models that uses unigrams, bigrams and trigrams respectively, to predict next likely words with some probabilities. The mixture model combines these words such that the ones with the most weighted sum of probability will be suggested. The weights given to each model were hyper parameters which had to be obtained through training.

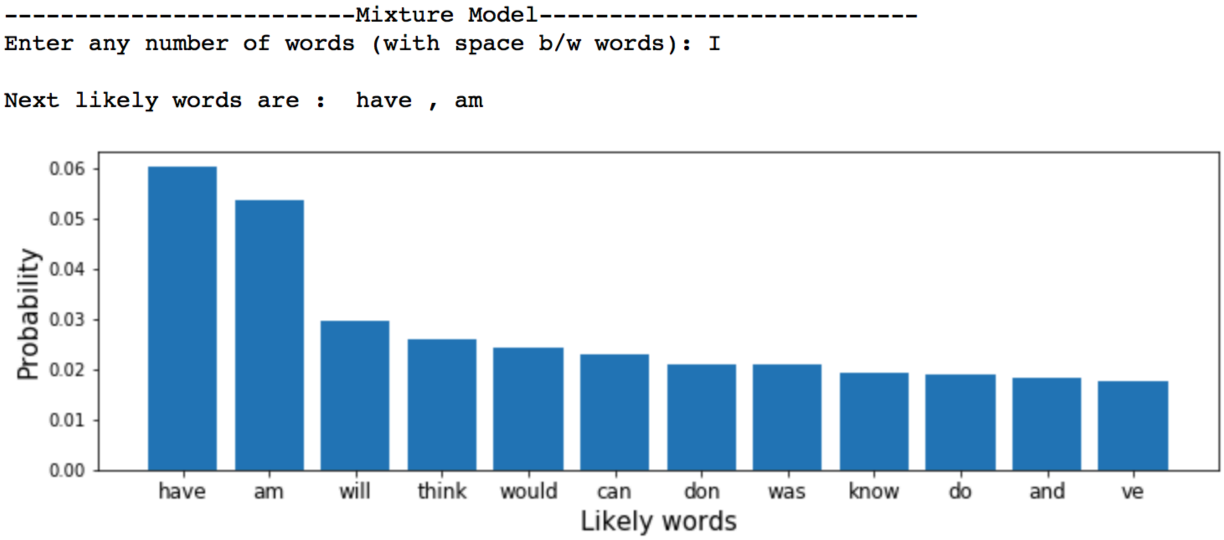
To train the mixture model, 1000 trigrams were randomly chosen from the Brown corpus (trigrams contains unigrams and bigrams). The trigram model was given a fixed weight of unity (as trigrams are usually the most accurate compared to unigrams and bigrams) and the other weights were varied to find the best fit.



Lamda2 was the weight of bigram model and Lamda1 that of unigram model. From training results, the best accuracy was obtained for Lamda2 = 0.9 and Lamda1 = 0.8.

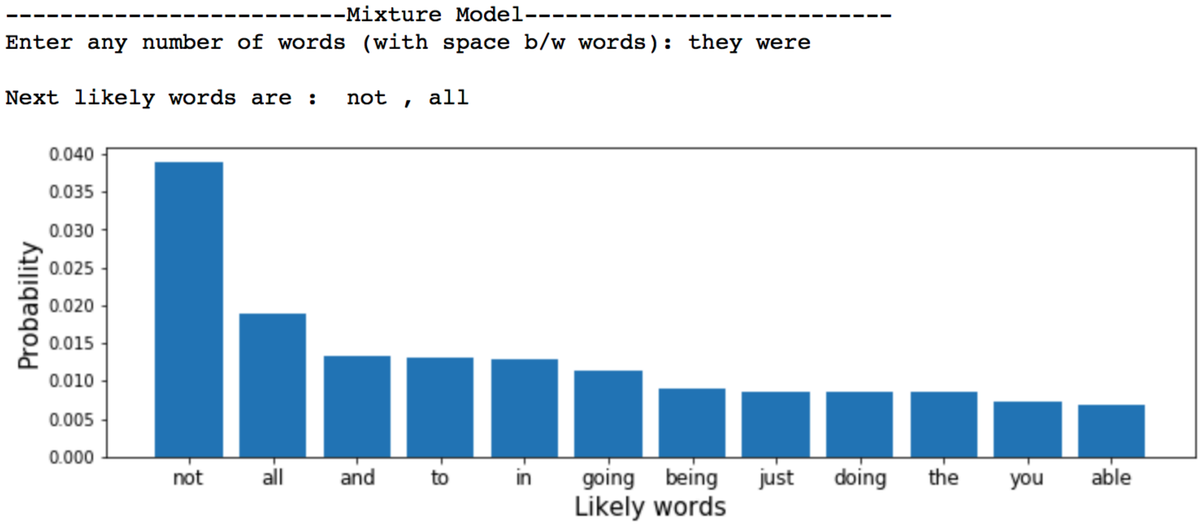
Results

The model was run with the hyper parameters for model weights. The model would ask a user to enter any number of words and suggests the next likely words.



```
-------------------------Mixture Model--------------------------
Enter any number of words (with space b/w words): I

Next likely words are :  have , am
```

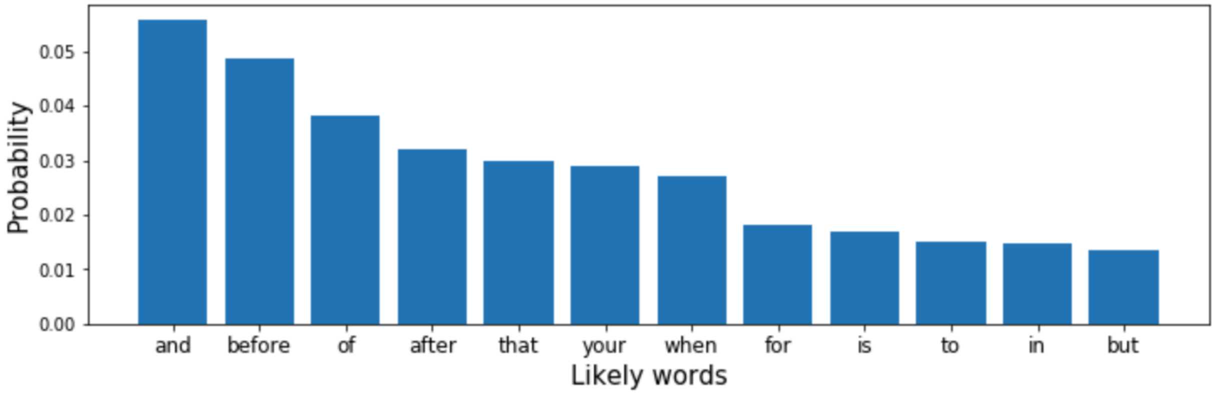


```
-------------------------Mixture Model--------------------------
Enter any number of words (with space b/w words): they were

Next likely words are :  not , all
```

```
-------------------------Mixture Model---------------------------
Enter any number of words (with space b/w words): Today is the day

Next likely words are :  and , before
```



Conclusions and Future Work

In this project, an n-gram mixture model was created that uses unigrams, bigrams and trigrams to suggest the next likely words, given few words. The model was trained to select proper weights and the results are shown above.

For future work, this model can be used with a speech to text converter to suggest next likely word from a person’s speech. It can also be used with search engines and messaging applications to autocomplete queries and messages. It can also be used in IDEs for code completion.