# Overview of Ngrams

1. **What are n-grams and how they are used to build a language model**
   - Ngram is a sliding window of size n in over the text. These ngram tokens can be used to create ngram dictionaries. Then we can use these dictionaries to create the probabilistic model of language

2. **List of a few applications where n-grams could be used**
   - Language identification
   - Spelling correction
   - Machine translation
   - Speech recognition
   - Auto suggestion

3. **A description of how probabilities are calculated for unigrams and bigrams**
   - ***For unigrams***: the probability of word w1_P(w1) is calculated by take the count of w1 divided by the number of tokens in text
     $P(w1) = C(w1) / N$ (Where N is the number of tokens in text)

   - ***For bigrams***: the probability of w1,w2_P(w1,w2) is calculated by multiply the probability of w1_P(w1) and the probability of w2 given that w1 was the previous word_P(w2|w1)
     $P(w1, w2) = P(w1) * P(w2|w1)$

4. **The importance of source text in building language model**
   - The source text is important because it greatly infuence the model. Different source text will produce different n-gram dictionaries. The type of words, the count number of words in these dictionaries will then be used to calculate the probability. Moreover, each source text will have different context.

5. **The importance of smoothing, and describe a simple approach for smoothing**
   - Smoothing is important because what would happen if one of the probabilities was zero, since we mutiply these probabilities together what would zero out the whole probability
   - A simple approach for smoothing is we are going to take a piece of the probability space and redistribute it so that we don't have zero probabilities. There are several ways to do that such as: Laplace smoothing or Good-Turing smoothing

- ***Laplace smoothing*** mean we add 1 to the numerator and to even this out we add our vocabulary size the denominator**.**

$$P(w_i) = \frac{C(w_i)+1}{N+V}$$

- Where V is the vocabulary size

- ***Good-Turing smoothing*****:** mean we replace zero counts/probabilities with counts/probabilities of words that occur only once

$$P(w_0) = \frac{N_1}{N}$$

6. **Describe how language model can be used for text generation, and the limitations of this approach**
   - The language model such as n-grams dictionary help machine understand how words are put together. Then use that information to generate the text. For example, given a start word, the language model will keep finding the next most likely word until there is nothing left to find.
   - The limitation of this approach is that we need a big corpus to actually generate a meaningful output. Another limitation is that the language model might not be build from the data with correct context.

7. **Describe how language models can be evaluated**
   - Language model can be evaluated by calculating the perplexity index. The perplexity specify how good a model can predict unseen word from the test data (The higher the better). The perplexity can be calculated by multiplicative inverse the probability of the test set then normalize it by number of word.

$$PP(W) = P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}}$$

$$= \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}$$

Chain rule:

$$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_1 \ldots w_{i-1})}}$$

For bigrams:

$$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_{i-1})}}$$

8. **Quick introduction to Google's ngram viewer and show an example**
   - The Google's ngram viewer display the how poplular the input phrases/words have occurred in a corpus of books over the seleted year. It also includes some advanced features such as: wildcard search, case insensitive search, part-of-speech tags.
   - Below show some interesting phrase during the space race between US and USSR. The peak in 1960s happen because that is the high of cold war, and the Moon landing happened on July 20, 1969