LANGAUGE MODEL & N GRAMS

Basic Probability

- **Probability Theory**: predicting how likely it is that something will happen.
- **Probabilities**: numbers between 0 and 1.
- **Probability Function**:
- – P(A) means that how likely the event A happens.
- – P(A) is a number between 0 and 1
- – P(A)=1 => a certain event
- – P(A)=0 => an impossible even
- **Example**: a coin is tossed three times. What is the probability of 3 heads?
- – 1/8

- **Unconditional and Conditional Probability**
- Unconditional Probability or Prior Probability
- – P(A) – the probability of the event A does not depend on other events.
- Conditional Probability -- Posterior Probability -- Likelihood
- – P(A|B) – this is read as the probability of A given that we know B.
- Example:
- – P(put) is the probability of to see the word put in a text
- – P(on|put) is the probability of to see the word on after seeing the word put.

LANGUAGE MODEL

- Language Model is the development of probabilistic models that are able to predict the next word in the sequence given the words that precede it.
- It is a probability distribution over sequences of the words.
- Given a sequence, say length of n, it assigns a probability.
- **P(w1 ,w2 ,w3 ,w4 ,w5…wn)** to the whole sequence.
- The goal of probabilistic language modelling is to calculate the probability of sentence of sequence of word W.
- **P(W) = P(w1 ,w2 ,w3 ,w4 ,w5…wn)**
- To find the probability of the next word in the sequence:
- **P(w5 |w1 ,w2 ,w3 ,w4)**

- A model that computes either of these:
- **P(W) or P(wn |w1 ,w2…wn-1) is called a language model.**
- **Methods for calculating Probabilities:**
- 1. Conditional Probabilities-
- Let A and B are two events, then the conditional probability of A given on B is:
- $\mathbf{P(A|B) = P(A,B) / P(B)}$
- $\mathbf{P(A,B) = P(B)\ P(A|B)}$
- Conditional Probabilities with More Variables:
- $\mathbf{P(A,B,C,D) = P(A)\ P(B|A)\ P(C|A,B)\ P(D|A,B,C)}$

- 2. Chain Rule
- The Chain Rule applied to compute joint probability of words in sentence.

$$P(w_1 w_2 \ldots w_n) = \prod_i P(w_i \mid w_1 w_2 \ldots w_{i-1})$$

- P(w1… wn) = P(w1) P(w2 |w1) P(w3 |w1w2)… P(wn |w1…wn-1)
- For Example: P("its water is so transparent") =

P(its)*

P(water|its)*

P(is|its water)*

P(so|its water is)*

P(transparent|its water is so)

- To compute probability of the:

$$P(\text{the} \mid \text{its water is so transparent that}) = \frac{Count(\text{its water is so transparent that the})}{Count(\text{its water is so transparent that})}$$

- To compute the exact probability of a word given a long sequence of preceding words is difficult (sometimes impossible).
- We are trying to compute P(wn |w1…wn-1) which is the probability of seeing wn after seeing w1 n-1 .
- Too many possible sentences and we may never see enough data for estimating these probability values.
- So, we need to compute P(wn |w1…wn-1) approximately.

- **Markov Assumption**
- A stochastic process has Markov property if the conditional probability distribution of future states of the process depends only upon the present state not on the sequence of events that precede it.
- A process with property is called a Morkov Process.
- The probability of the next word can be estimated given only the previous k number of words.
- Ex: k=1:

$$P(\text{the} \mid \text{its water is so transparent that}) \approx P(\text{the} \mid \text{that})$$

- Ex: k=2:

$$P(\text{the} \mid \text{its water is so transparent that}) \approx P(\text{the} \mid \text{transparent that})$$

- For k-i, we compute the probability as follows:

$$P(w_i \mid w_1 w_2 \Box \;\; w_{i-1}) \approx P(w_i \mid w_{i-k} \Box \;\; w_{i-1})$$

- A bigram is called a first-order Markov model (because it looks one token into the past);
- A trigram is called a second-order Markov model;
- In general a N-Gram is called a N-1 order Markov model.

N-Gram

- Models that assign probabilities to sequences of words are called language models (LMs).
- The simplest language model that assigns probabilities to sentences and sequences of words is the n-gram.
- An n-gram is a sequence of N words:
- – A 1-gram (unigram) is a single word sequence of words like "please" or " turn".
- – A 2-gram (bigram) is a two-word sequence of words like "please turn", "turn your", or ”your homework".
- – A 3-gram (trigram) is a three-word sequence of words like "please turn your", or "turn your homework".
- We can use n-gram models to estimate the probability of the last word of an n-gram given the previous words, and also to assign probabilities to entire word sequences.

- Lets start with equation:
- **P(w|h)**: The probability of word w, on given some history h.
- For Ex:
- P(the| its water is so transparent that)
- Here,
- w- the
- h- its water is so transparent that
- Instead of computing the probability of a word given its entire history, we can approximate the history by just the last few words.

$P(w_n \vert w_1 \dots w_{n-1}) \approx P(w_n)$	**unigram**
$P(w_n \vert w_1 \dots w_{n-1}) \approx P(w_n \vert w_{n-1})$	**bigram**
$P(w_n \vert w_1 \dots w_{n-1}) \approx P(w_n \vert w_{n-1} w_{n-2})$	**trigram**
$P(w_n \vert w_1 \dots w_{n-1}) \approx P(w_n \vert w_{n-1} w_{n-2} w_{n-3})$	**4-gram**
$P(w_n \vert w_1 \dots w_{n-1}) \approx P(w_n \vert w_{n-1} w_{n-2} w_{n-3} w_{n-4})$	**5-gram**

In general, **N-Gram** is

$$\mathbf{P(w_n|w_1 \dots w_{n\text{-}1}) \approx P(w_n|w_{n-N+1}^{n-1})}$$