SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 97
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Artificial Intelligence and Data Science

Course Name – 19AD601 – Natural Language Processing

III Year / VI Semester

Unit 2 – WORD LEVEL ANALYSIS

Topic 2- N-grams

N-grams

Models that assign probabilities to sequences of words are called language models or 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 or Bi-gram is a two-word sequence of words like “please turn”, “turn your”, or “your homework”, and
A 3-gram (a 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

N-grams

N-Grams

The task of computing P(w|h), the probability of a word w given some history h. Suppose the history h is "its water is so transparent that" and we want to know the probability that the next word is the: P(the | its water is so transparent that):

One way to estimate this probability is from relative frequency counts: take a very large corpus, count the number of times we see its water is so transparent that, and count the number of times this is followed by the.

This would be answering the question "Out of the times we saw the history h, how many times was it followed by the word w", as follows:

$$P(the|its\ water\ is\ so\ transparent\ that) = \frac{C(its\ water\ is\ so\ transparent\ that\ the)}{C(its\ water\ is\ so\ transparent\ that)}$$

N-grams

Chain Rule Probability

Suppose if we compute probabilities of entire sequences like P(w1;w2; :::;wn), we can do is decompose this probability using the chain rule of probability, given below

$$P(X_1 \ldots X_n) = P(X_1)P(X_2|X_1)P(X_3|X_{1:2}) \ldots P(X_n|X_{1:n-1})$$
$$= \prod_{k=1}^{n} P(X_k|X_{1:k-1})$$

Applying the chain rule to words, we get

$$P(w_{1:n}) = P(w_1)P(w_2|w_1)P(w_3|w_{1:2}) \ldots P(w_n|w_{1:n-1})$$
$$= \prod_{k=1}^{n} P(w_k|w_{1:k-1})$$

N-grams

The intuition of the n-gram model (simplifying assumption):
– instead of computing the probability of a word given its entire history, we can approximate the history by just the last few words.

$$P(w_n|w_1 \ldots w_{n-1}) \approx P(w_n) \quad \text{unigram}$$
$$P(w_n|w_1 \ldots w_{n-1}) \approx P(w_n|w_{n-1}) \quad \text{bigram}$$
$$P(w_n|w_1 \ldots w_{n-1}) \approx P(w_n|w_{n-1}w_{n-2}) \quad \text{trigram}$$
$$P(w_n|w_1 \ldots w_{n-1}) \approx P(w_n|w_{n-1}w_{n-2}w_{n-3}) \quad \text{4-gram}$$
$$P(w_n|w_1 \ldots w_{n-1}) \approx P(w_n|w_{n-1}w_{n-2}w_{n-3}w_{n-4}) \quad \text{5-gram}$$

In general, **N-Gram** is

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

N-grams

N-Grams and Markov Models

The assumption that the probability of a word depends only on the previous word(s) is called Markov assumption.

Markov models are the class of probabilistic models that assume that we can predict the probability of some future unit without looking too far into the past.

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-grams

Estimating N-Gram Probabilities

Estimating n-gram probabilities is called maximum likelihood estimation (or MLE). We get the MLE estimate for the parameters of an n-gram model by getting counts from a corpus, and normalizing the counts so that they lie between 0 and 1.

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_w C(w_{n-1}w)} = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

where C is the count of that pattern in the corpus

$$P(w_n \mid w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1} w_n)}{C(w_{n-N+1}^{n-1})}$$

THANK YOU