Natural Language Processing and Machine Translation

Language Models

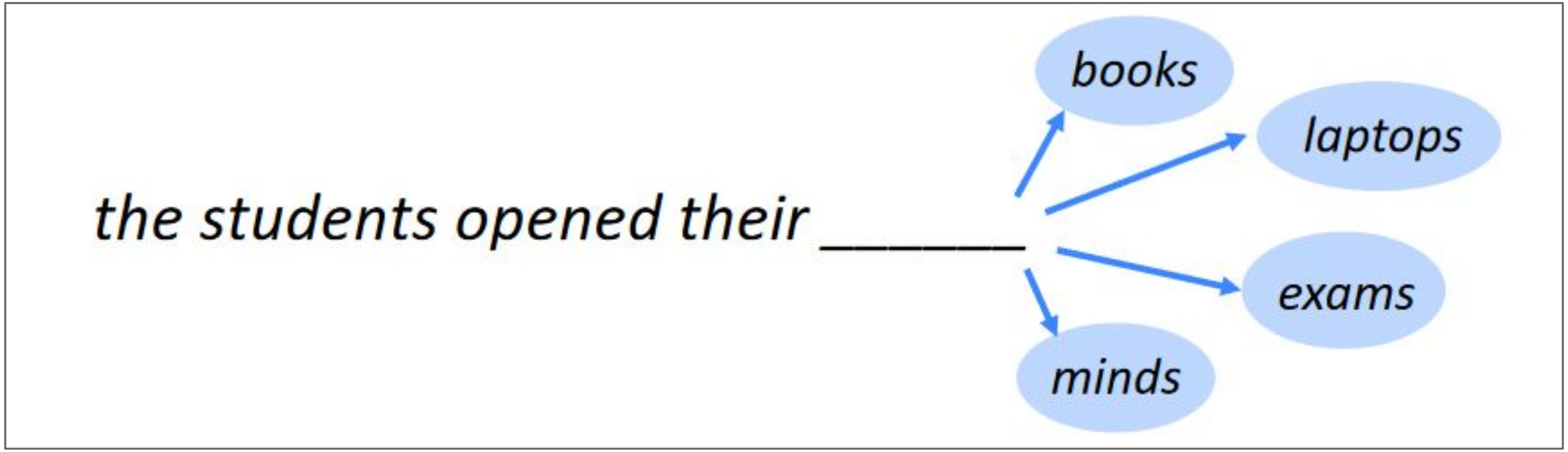
Abhishek Koirala

M.Sc. in Informatics and Intelligent Systems Engineering

THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING

Introduction

- Use of various statistical and probabilistic techniques to determine the probability of a given sequence of words occurring in a sentence
- Analyze bodies of text data to provide a base for word predictions



https://medium.com/@antonio.lopardo/the-basics-of-language-modeling-1c8832f21079

N-gram

The cow jumps over the moon

Unigram/ 1-gram

The
cow
jumps
over
the
moon

Bigram/2-gram

The cow
cow jumps
jumps over
over the
the moon

3-gram

The cow jumps
cow jumps over
jumps over the
over the moon

4-gram

The cow jumps over
cow jumps over the
jumps over the moon

If X=Num of words in a given sentence K, the number of n-grams for sentence K would be:

$$Ngrams_K = X - (N - 1)$$

THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING

Its water is so transparent that ....

$$P(the|its\ water\ is\ so\ transparent\ that).$$

One approach to calculate this using frequency approach

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

Will this give us a good estimate in all possible scenarios ??

THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING

Another way to do this is using **chain rule of probability**

p(w1...ws) = p(w1) . p(w2 | w1) . p(w3 | w1 w2) . p(w4 | w1 w2 w3) ..... p(wn | w1...wn-1)

But this is again computationally expensive

We make this more simpler with an assumption:

- We approximate the context of the word wk by looking at the last word of the context. (**Markov Assumption**)

Eg. for bigram

$$p(w) = \prod_{i=1}^{k+1} p(w_i | w_{i-1})$$

THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING

N-gram language models

<s> I am a human </s>
<s> I am not a stone </s>
<s> I live in Lahore </s>

P(I|<S>)=C(<s>|I)/C(<s>)=3/3=1

P(am|I)=C(I|am)/C(I)=⅔

P(a|am)=C(am|a)/C(a)=½

P(human|a)=C(a|human)/C(a)=½

P(</s>|human)=C(human|</s>)/C(human)=1

P(not|am)=C(am|not)/C(am)=½

P(a|not)=C(not|a)/C(not)=1

P(stone|a)=C(a|stone)/C(a)=½

P(</s>|stone)=C(stone|</s>)/C(stone)=1

P(live|I)=C(I|live)/C(I)=⅓

P(in|live)=C(live|in)/C(live)=1

P(Lahore|in)=C(in|Lahore)/C(in)=1

P(</s>|Lahore)=C(Lahore|</s>)/C(Lahore)=1

P(I am a human)
= P(I|<s>) P(am|I) P(a|am) P(human|a) P(</s>|human)
= 1 * ⅔ * ½ *½ * 1
= ⅙

P(I am human)
= P(I|<s>) P(am|I) P(human|am) P(</s>|human)
= 1 * ⅔ * 0 * 1
= 0 => Does this seem correct?

THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING

Laplace Smoothing

<s> I am a human </s>
<s> I am not a stone </s>
<s> I live in Lahore </s>

P(I|<S>)=C(<s>|I)/C(<s>)=3/3=1

P(am|I)=C(I|am)/C(I)=⅔

P(a|am)=C(am|a)/C(a)=½

P(human|a)=C(a|human)/C(a)=½

P(</s>|human)=C(human|</s>)/C(human)=1

P(not|am)=C(am|not)/C(am)=½

P(a|not)=C(not|a)/C(not)=1

P(stone|a)=C(a|stone)/C(a)=½

P(</s>|stone)=C(stone|</s>)/C(stone)=1

P(live|I)=C(I|live)/C(I)=⅓

P(in|live)=C(live|in)/C(live)=1

P(Lahore|in)=C(in|Lahore)/C(in)=1

P(</s>|Lahore)=C(Lahore|</s>)/C(Lahore)=1

The solution to the problem of unseen N-grams is to re-distribute some of the probability mass from the observed frequencies to unseen N-grams. This is a general problem in probabilistic modeling called **smoothing**.

$$P_{\text{Laplace}}(w_i) = \frac{c_i + 1}{N + V}$$

Using laplace smoothing (Vocab = 11)

P(I am human)
= P(I|<s>) P(am|I) P(human|am) P(</s>|human)
= (3+1)/(3+11) * (2+1)/(3+11) * (0+1)/(2+11) * (1+1)/(1+11)
= 4/14 * 3/14 * 1/13 * 2/12
= 0.00078

THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING

Good Turing Discounting

- Re-estimate the amount of probability mass to assign N-gram with zero or low counts by looking at the number of N-grams with higher counts
- Use the count of things which are seen once to help estimates the count of things never seen.
- Let Nc be number of N-grams that occur c times
  - For bigrams, No, is the number of bigrams of count 0, N1, is the number of bigrams with count 1, etc
- Revised count

$$\boxed{c^* = (c+1)\frac{N_{c+1}}{N_c}}$$

THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING

Kneser Ney Smoothing

- Let the count assigned to each unigram be the number of different words that it follows. Define:

$$N_{1+}(\bullet\ w_i) = |\{w_{i-1} : c(w_{i-1}w_i) > 0\}|$$

$$N_{1+}(\bullet\ \bullet) = \sum_{w_i} N_{1+}(\bullet\ w_i)$$

- Let lower-order distribution be:

$$p_{KN}(w_i) = \frac{N_{1+}(\bullet\ w_i)}{N_{1+}(\bullet\ \bullet)}$$

- Put it all together:

$$p_{KN}(w_i|w_{i-n+1}^{i-1}) = \frac{\max\{c(w_{i-n+1}^{i}) - \delta, 0\}}{\sum_{w_i} c(w_{i-n+1}^{i})} + \frac{\delta}{\sum_{w_i} c(w_{i-n+1}^{i})} N_{1+}(w_{i-n+1}^{i-1}\ \bullet) p_{KN}(w_i|w_{i-n+2}^{i-1})$$

THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING

Evaluating Language Models

2 types of evaluation

- Extrinsic evaluation
- Intrinsic evaluation

Extrinsic Evaluation

- Model metrics compared with respect to applications implemented in

Intrinsic Evaluation

- Single model evaluation
    - Perplexity

THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING

Perplexity

I always order burger with ....

fries
drinks
sausage
....
....
burgers
with burgers

A good language models with add words like fries, drinks or sausage to the above sentence while a bad language model could add completely random words

A better model of text is the one that assigns a higher probability to the words that actually occurs.

Perplexity is the probability of test set normalized by the number of words

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

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

THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING

Perplexity as average branching factor

How hard is the task of recognizing digits ‘0’, ‘1’,‘2’,‘3’,‘4’,‘5’,‘6’,‘8’,‘8’,‘9’ ?

There were …….. people in the room.

Assuming that the above space can be filled with any digit from 0-9 and probability of all these digits are equally likely

P(any digit) = 1/10

PP(any digit)= (1/10)^-1

= 10

Lower the perplexity, better the model

Perplexity as average branching factor

How hard is the task of recognizing digits ‘0’, ‘1’,‘2’,‘3’,‘4’,‘5’,‘6’,‘8’,‘8’,‘9’ ?

There were …….. people in the room.

Assuming that the above space can be filled with any digit from 0-9 and probability of all these digits are equally likely

P(any digit) = 1/10

PP(any digit)= (1/10)^-1

= 10

Lower the perplexity, better the model

Backoff

- Non linear method
- Estimate for an n-gram is allowed to back off through progressively shorter histories
- Trigram version

$$\hat{P}(w_i \mid w_{i-2}w_{i-1}) = \begin{cases} P(w_i \mid w_{i-2}w_{i-1}), & \text{if } C(w_{i-2}w_{i-1}w_i) > 0 \\ \alpha_1 P(w_i \mid w_{i-1}), & \text{if } C(w_{i-2}w_{i-1}w_i) = 0 \\ & \text{and } C(w_{i-1}w_i) > 0 \\ \alpha_2 P(w_i), & \text{otherwise.} \end{cases}$$

THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING

THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING