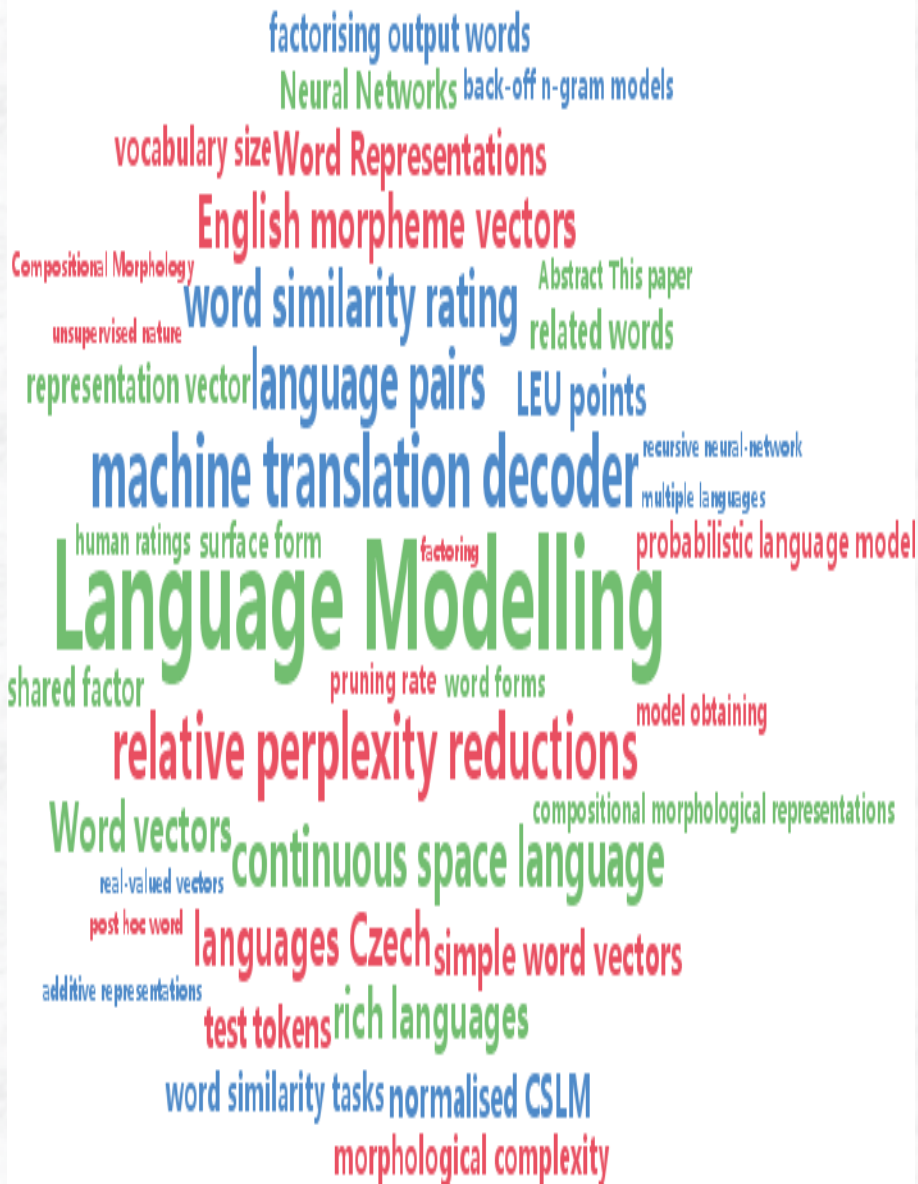


Language Modeling



N-Gram Chapter 6

Instructor : Dr. Hanaa Bayomi Ali
Mail : h.mobarz @ fci-cu.edu.eg

Probabilistic Language Models

- **The goal: assign a probability to a sentence**
 - Speech recognition was the original motivation.
(Related problems are optical character recognition, handwriting recognition.)
 - $P(\text{I saw a van}) \gg P(\text{eyes awe of an})$
 - Machine Translation:
 - $P(\text{high winds tonight}) > P(\text{large winds tonight})$
 - Spelling Correction
The office is about fifteen **minuets** from my house
 - $P(\text{about fifteen minutes from}) > P(\text{about fifteen minuets from})$
 - + Summarization, question-answering, augmentative communication for the disabled, etc., etc.!!

Probabilistic Language Models

- Goal: compute the probability of a sentence or sequence of words W :

$$P(W) = P(w_1, w_2, w_3, w_4, w_5 \dots w_n)$$

- Related task: probability of an upcoming word:

$$P(w_5 | w_1, w_2, w_3, w_4)$$

- A model that computes either of these:

$P(W)$ or $P(w_n | w_1, w_2 \dots w_{n-1})$ is called a **language model**.

How to compute $P(W)$

- How to compute this joint probability:
 - $P(\text{its, water, is, so, transparent, that})$
- Intuition: let's rely on the **Chain Rule of Probability**

The Chain Rule: General

- The definition of conditional probabilities

$$P(B \mid A) = P(A, B) / P(A)$$

Rewriting: $P(A, B) = P(B \mid A) P(A)$

- More variables:

$$P(A,B,C,D) = P(A)P(B \mid A)P(C \mid A,B)P(D \mid A,B,C)$$

- The Chain Rule in General

$$P(x_1, x_2, x_3, \dots, x_n) = P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1, x_2) \dots P(x_n \mid x_1, \dots, x_{n-1})$$

The Chain Rule: joint probability in sentence

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

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)

How to estimate these probabilities

- Could we just count and divide?

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

- No! Too many possible sentences!
- We'll never see enough data for estimating these

Markov Assumption

- Instead of count all possible sentences , it is enough to count last few words.

- Simplifying assumption:

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

- Or maybe

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

Markov Assumption

$$P(w_1 w_2 \dots w_n) \approx \prod_i P(w_i \mid w_{i-N+1} \dots w_{i-1})$$

- In other words, we approximate each component in the product

$$P(w_i \mid w_1 w_2 \dots w_{i-1}) \approx P(w_i \mid w_{i-N+1} \dots w_{i-1})$$

Simplest case: Unigram model

$$P(w_1 w_2 \dots w_n) \approx \prod_i P(w_i)$$

- An example

I am Sam
Sam I am
I do not like green apples
and bananas

No. Words= 14

$P(I) = 3$	$3/14 = 0.21$
$P(sam) = 2$	$2/14 = 0.14$



THINK



Question

Which is assigned higher probability by a unigram language model for English?

- $P(\text{I like ice cream})$
- $P(\text{the the the the})$
- $P(\text{Go to class daily})$
- $P(\text{class daily go to})$

Answer

Which is assigned higher probability by a unigram language model for English?

- $P(\text{I like ice cream})$
- $P(\text{the the the the})$
- $P(\text{Go to class daily})$
- $P(\text{class daily go to})$

The word "the" is very frequent in English. A unigram language model does not depend on surrounding words, so "the the the the" gets a high probability even though it isn't regularly used.

Bigram model

- Condition on the previous word:

$$P(w_i | w_1 w_2 \dots w_{i-1}) \approx P(w_i | w_{i-1})$$

$$P(w_i | w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

- An example

I am Sam
Sam I am
I do not like green apples
and bananas

No. Words= 14

$P(\text{do} | \text{I}) =$

$P(\text{am} | \text{I}) =$

N-gram models

- We can extend to trigrams, 4-grams, 5-grams
- In general this is an insufficient model of language
 - because language has **long-distance dependencies**

“The computer which I had just put into the machine room on the fifth floor crashed.”

- But we can often get away with N-gram models



THINK



QUESTION 2

If we estimate a bigram language model from the following corpus, **what is $P(\text{not}|\text{do})$** ?

I am Sam

Sam I am

I do not like green eggs and ham

Answer

$$P(\text{not}|\text{do}) = C(\text{do}, \text{not}) / C(\text{do}) = 1/1 = 1$$

More examples: Berkeley Restaurant Project sentences

- can you tell me about any good cantonese restaurants close by
- mid priced thai food is what i'm looking for
- tell me about chez panisse
- can you give me a listing of the kinds of food that are available
- i'm looking for a good place to eat breakfast
- when is caffe venezia open during the day

Raw bigram counts (absolute measure)

- Out of 9222 sentences

B\A	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Raw bigram probabilities (relative measure)

- Normalize by unigrams:

i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

- Result: $P(\text{want} | i) = C(i, \text{want}) / C(i) = 927 / 2533 = 0.33$

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

Bigram estimates of sentence probabilities

P(I want Chinese food) =

$$P(\text{want}|\text{I}) \times P(\text{Chinese}|\text{want}) \times P(\text{food}|\text{Chinese})$$

$$= 0.33 \times 0.0065 \times 0.52 = \mathbf{0.00112}$$

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

What kinds of knowledge?

- $P(\text{english} | \text{want}) = .0011$
 - $P(\text{chinese} | \text{want}) = .0065$
 - $P(\text{to} | \text{want}) = .66$
 - $P(\text{eat} | \text{to}) = .28$
 - $P(\text{food} | \text{to}) = 0$
 - $P(\text{want} | \text{spend}) = 0$
- world
- grammar
- grammar (structural zero)

Practical Issues

- We do everything in log space
 - Avoid underflow
 - (also adding is faster than multiplying)

$$p_1 \cdot p_2 \cdot p_3 \cdot p_4 = \log p_1 + \log p_2 + \log p_3 + \log p_4$$

