— INF4820 —
Algorithms for AI and NLP

Basic Probability Theory & Language Models

Murhaf Fares & Stephan Oepen

Language Technology Group (LTG)

October 20, 2016

So far...

- Vector space model
- Classification
  - Rocchio
  - kNN
- Clustering
  - K-means

Point-wise prediction; geometric models.

Today onwards

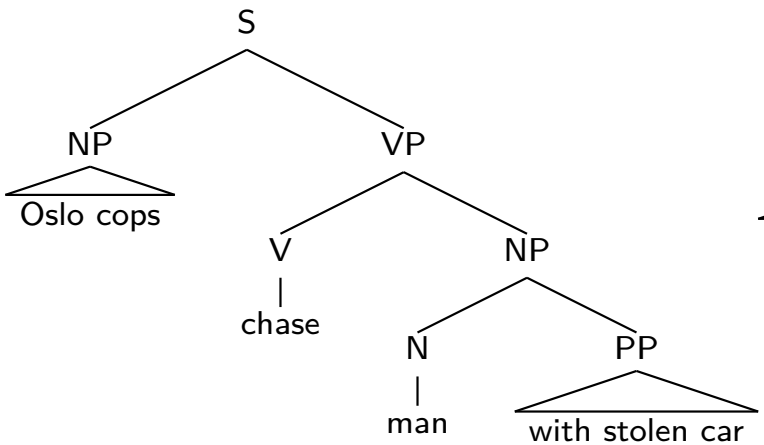
Structured prediction; probabilistic models.

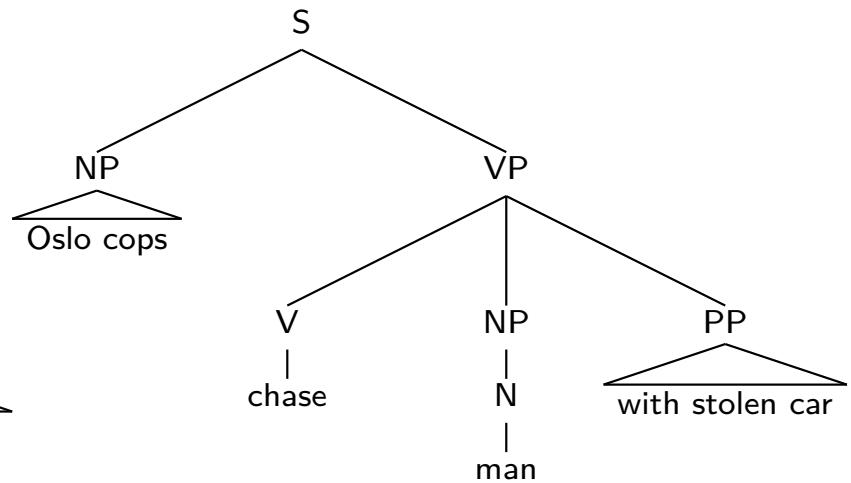
- Sequences
  - Language models
- Labelled sequences
  - Hidden Markov Models
- Trees
  - Statistical (Chart) Parsing

Most Likely Interpretation

Probabilistic models to determine the *most likely interpretation*.

- Which string is most likely?
  - *She studies morphosyntax* vs. *She studies more faux syntax*
- Which category sequence is most likely for *Time flies like an arrow*?
  - Time_N flies_N like_V an_D arrow_N
  - Time_N flies_V like_P an_D arrow_N
- Which syntactic analysis is most likely?





Probability Basics (1/4)

- Experiment (or trial)
  - the process we are observing
- Sample space (Ω)
  - the set of all possible outcomes of a random experiment
- Event(s)
  - the subset of Ω we are interested in
- Our goal is to assign probability to certain events
  - $P(A)$ is the probability of event A, a real number $\in [0, 1]$

Probability Basics (2/4)

- Experiment (or trial)
  - rolling a die
- Sample space (Ω)
  - $\Omega = \{1, 2, 3, 4, 5, 6\}$
- Event(s)
  - $A =$ rolling a six: $\{6\}$
  - $B =$ getting an even number: $\{2, 4, 6\}$
- Our goal is to assign probability to certain events
  - $P(A) =?$ $P(B) =?$

Probability Basics (3/4)

- Experiment (or trial)
  - flipping two coins
- Sample space (Ω)
  - $\Omega = \{HH, HT, TH, TT\}$
- Event(s)
  - $A =$ the same outcome both times: $\{HH, TT\}$
  - $B =$ at least one head: $\{HH, HT, TH\}$
- Our goal is to assign probability to certain events
  - $P(A) =?$ $P(B) =?$

Probability Basics (4/4)

- Experiment (or trial)
  - rolling two dice
- Sample space (Ω)
  - $\Omega = \{11, 12, 13, 14, 15, 16, 21, 22, 23, 24, \ldots, 63, 64, 65, 66\}$
- Event(s)
  - $A =$ results sum to 6: $\{15, 24, 33, 42, 51\}$
  - $B =$ both results are even: $\{22, 24, 26, 42, 44, 46, 62, 64, 66\}$
- Our goal is to assign probability to certain events
  - $P(A) = \frac{|A|}{|\Omega|}$ $P(B) = \frac{|B|}{|\Omega|}$

Axioms

Probability Axioms

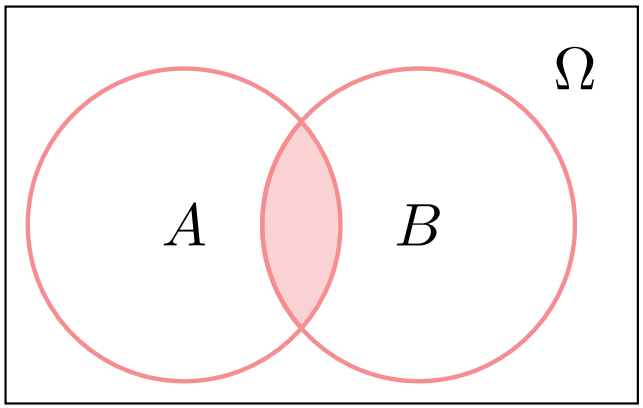
- $0 \leqslant P(A) \leqslant 1$
- $P(\Omega) = 1$
- $P(A \cup B) = P(A) + P(B)$ where A and B are mutually exclusive

More useful axioms

- $P(A) = 1 - P(\neg A)$
- $P(\emptyset) = 0$

Joint Probability

- $P(A, B)$: probability that both A and B happen
- also written: $P(A \cap B)$ or $P(A, B)$

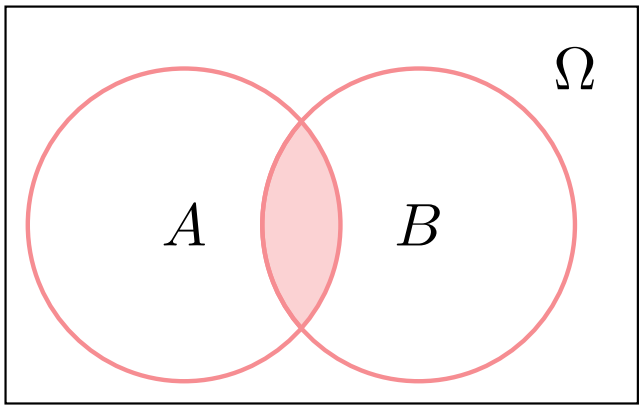


What is the probability, when throwing two fair dice, that

- A: the results sum to 6 and
- B: at least one result is a 1?

Joint Probability

- $P(A, B)$: probability that both A and B happen
- also written: $P(A \cap B)$ or $P(A, B)$

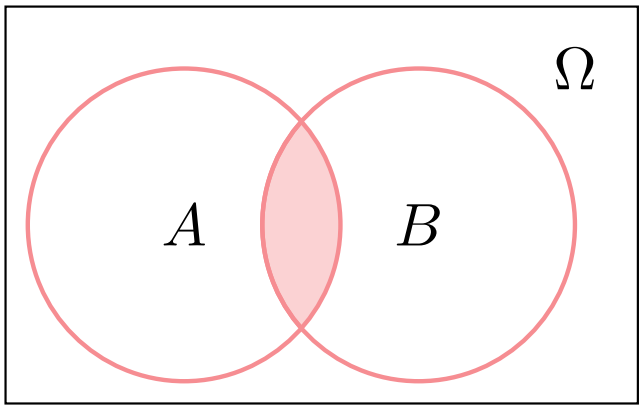


What is the probability, when throwing two fair dice, that

- A: the results sum to 6 and
- B: at least one result is a 1?

Joint Probability

- $P(A, B)$: probability that both A and B happen
- also written: $P(A \cap B)$ or $P(A, B)$

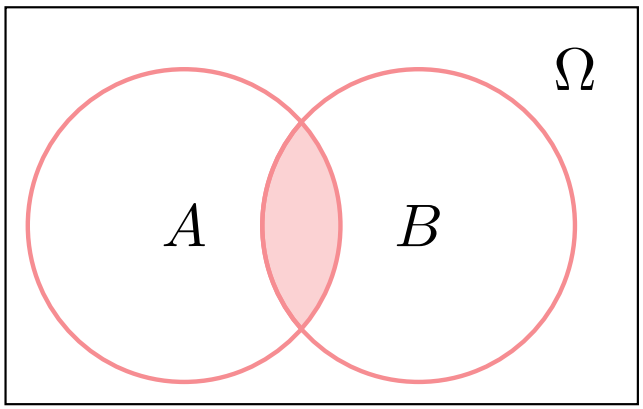


What is the probability, when throwing two fair dice, that

- A: the results sum to 6 and $\frac{5}{36}$
- B: at least one result is a 1?

Joint Probability

- $P(A, B)$: probability that both A and B happen
- also written: $P(A \cap B)$ or $P(A, B)$



What is the probability, when throwing two fair dice, that

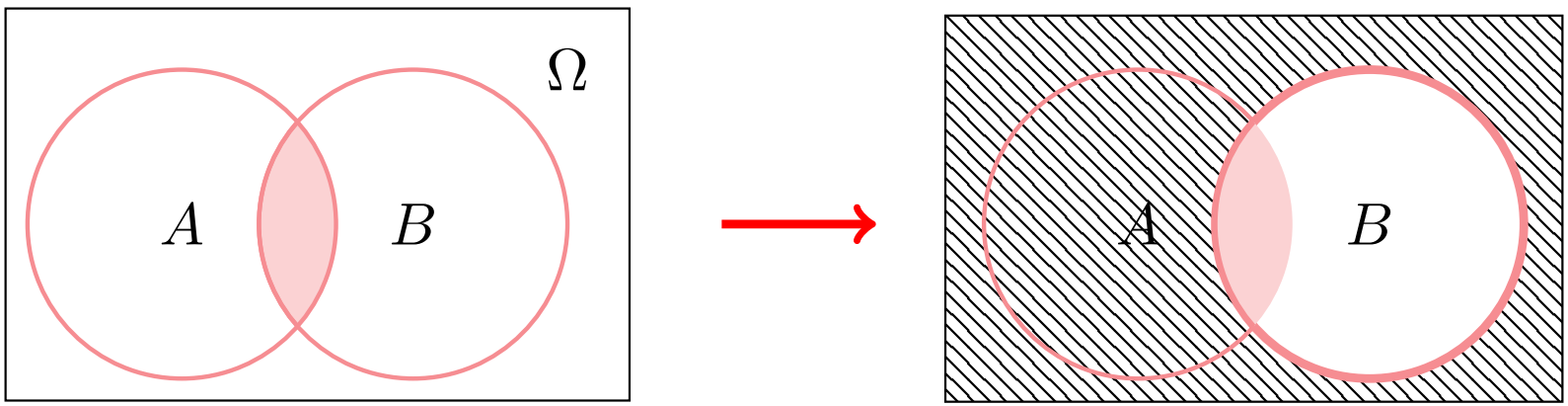
- A: the results sum to 6 and $\frac{5}{36}$
- B: at least one result is a 1? $\frac{11}{36}$

Conditional Probability

Often, we have partial knowledge about the outcome of an experiment.

What is the probability $P(A|B)$, when throwing two fair dice, that

- A: the results sum to 6 given
- B: at least one result is a 1?



$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad \text{(where } P(B) > 0\text{)}$$

The Chain Rule

Joint probability is symmetric:

$$\begin{aligned} P(A \cap B) &= P(A)\,P(B|A) \\ &= P(B)\,P(A|B) \quad \text{(multiplication rule)} \end{aligned}$$

More generally, using the chain rule:

$$P(A_1 \cap \cdots \cap A_n) = P(A_1)P(A_2|A_1)P(A_3|A_1 \cap A_2)\ldots P(A_n|\cap_{i=1}^{n-1} A_i)$$

The chain rule will be very useful to us through the semester:

- it allows us to break a complicated situation into parts;
- we can choose the breakdown that suits our problem.

(Conditional) Independence

- Let A be the event that it rains tomorrow $P(A) = \frac{1}{3}$
- Let B be the event that flipping a coin results in heads $P(B) = \frac{1}{2}$
- What is $P(A|B)$?

If knowing event B is true has no effect on event A, we say

A and B are independent of each other.

If A and B are independent:

- $P(A \cap B) = P(A)\,P(B)$
- $P(A|B) = P(A)$
- $P(B|A) = P(B)$

Intuition? (1/3)

- Your friend, Yoda, wakes up in the morning feeling sick.
- He uses a website to diagnose his disease by entering the symptoms
- The website returns that 99% of the people who had a disease D had the same symptoms Yoda has.
- Yoda freaks out, comes to your place and tells you the story.
- You are more relaxed, you continue reading the web page Yoda started reading, and you find the following information:

- The prevalence of disease D: 1 in 1000 people
- The reliability of the symptoms:
  - False negative rate: 1%
  - False positive rate: 2%

What is the probability that he has the disease?

Intuition? (2/3)

Given:

- event A: has disease
- event B: has the symptoms

We know:

- $P(A) = 0.001$
- $P(B|A) = 0.99$
- $P(B|\neg A) = 0.02$

We want

- $P(A|B) = ?$

Intuition? (3/3)

	A	¬ A	
B	0.00099	0.01998	0.02097
¬ B	0.00001	0.97902	0.97903
	0.001	0.999	1

$$P(A) = 0.001; \quad P(B|A) = 0.99; \quad P(B|\neg A) = 0.02$$

$$P(A \cap B) = P(B|A)P(A)$$
$$P(\neg A \cap B) = P(B|\neg A)P(\neg A)$$

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{0.00099}{0.02097} = 0.0472$$

Bayes' Theorem

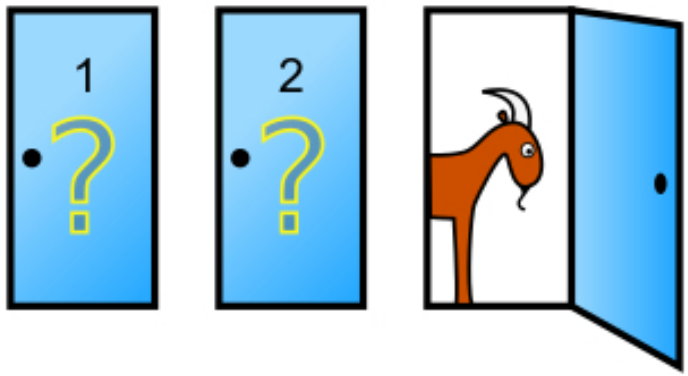
- From the two 'symmetric' sides of the joint probability equation:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- reverses the order of dependence (which can be useful)
- in conjunction with the chain rule, allows us to determine the probabilities we want from the probabilities we know

Bonus: The Monty Hall Problem

- On a gameshow, there are three doors.
- Behind 2 doors, there is a goat.
- Behind the 3rd door, there is a car.



- The contestant selects a door that she hopes has the car behind it.
- Before she opens that door, the gameshow host opens one of the other doors to reveal a goat.
- The contestant now has the choice of opening the door she originally chose, or switching to the other unopened door.

What should she do?

What's Next?

- Now that we have ____ the basics in probability theory we can move to a new topic.
- Det var en _____?
- Je ne parle pas ____?

Natural language contains redundancy, hence can be predictable.

Previous context can constrain the next word

- semantically;
- syntactically;

$\rightarrow$ by frequency – language models.

Language Models in NLP

Language model: a probabilistic model that assigns *approximate* probability to an arbitrary sequence of words.

- Machine translation
  - She is going home vs. She is going house
- Speech recognition
  - She studies morphosyntax vs. She studies more faux syntax
- Spell checkers
  - *Their* are many NLP applications that use language models.
- Input prediction (predictive keyboards)

Language Models

- A probabilistic language model M assigns probabilities $P_M(x)$ to all strings x in language L.
  - L is the sample space
  - $0 \leq P_M(x) \leq 1$
  - $\sum_{x \in L} P_M(x) = 1$

Language Models

- Given a sentence $S = w_1 \dots w_n$, we want to estimate $P(S)$
- $P(S)$ is the joint probability over the words in S: $P(w_1, w_2 \dots, w_n)$
- We can calculate the probability of S using the chain rule:

$$P(w_1 \dots w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1 \cap w_2) \dots P(w_n| \cap_{i=1}^{n-1} w_i)$$

- Example:

$P(\textit{I want to go to the beach}) =$
$P(\textit{I})\, P(\textit{want}|\textit{I})\, P(\textit{to}|\textit{I want})\, P(\textit{go}|\textit{I want to})\, P(\textit{to}|\textit{I want to go}) \dots$

Language Models

- Given a sentence $S = w_1 \dots w_n$, we want to estimate $P(S)$
- $P(S)$ is the joint probability over the words in S: $P(w_1, w_2 \dots, w_n)$
- We can calculate the probability of S using the chain rule:

$$P(w_1 \dots w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1 \cap w_2) \dots P(w_n| \cap_{i=1}^{n-1} w_i)$$

- Example:

$P(\textit{I want to go to the beach}) =$
$P(\textit{I})\, P(\textit{want}|\textit{I})\, P(\textit{to}|\textit{I want})\, P(\textit{go}|\textit{I want to})\, P(\textit{to}|\textit{I want to go}) \dots$

Language Models

- Given a sentence $S = w_1 \dots w_n$, we want to estimate $P(S)$
- $P(S)$ is the joint probability over the words in S: $P(w_1, w_2 \dots, w_n)$

Recall The Chain Rule

$$P(A_1 \cap \dots \cap A_n) = P(A_1)P(A_2|A_1)P(A_3|A_1 \cap A_2) \dots P(A_n| \cap_{i=1}^{n-1} A_i)$$

- We can calculate the probability of S using the chain rule:

$$P(w_1 \dots w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1 \cap w_2) \dots P(w_n| \cap_{i=1}^{n-1} w_i)$$

- Example:

$P(\textit{I want to go to the beach}) =$
$P(\textit{I})\, P(\textit{want}|\textit{I})\, P(\textit{to}|\textit{I want})\, P(\textit{go}|\textit{I want to})\, P(\textit{to}|\textit{I want to go}) \dots$

Markov Assumption

$P(I\ want\ to\ go\ to\ the\ beach\ to\ read\ about\ Markov\ assumption) =$
$P(I)\ P(want|I)\ P(to|I\ want)\ P(go|I\ want\ to)$
$\ldots P(assumption|I\ want\ to\ go\ to\ the\ beach\ to\ read\ about\ Markov)$

- Simplifying assumption (limited history):

$$P(assumption|I\ want\ to\ go\ to\ the\ beach\ to\ read\ about\ Markov) \approx P(assumption|Markov)$$

- Or:

$$P(assumption|I\ want\ to\ go\ to\ the\ beach\ to\ read\ about\ Markov) \approx P(assumption|about\ Markov)$$

N-Grams

We simplify using the Markov assumption (limited history):

the last $n-1$ elements can approximate the effect of the full sequence.

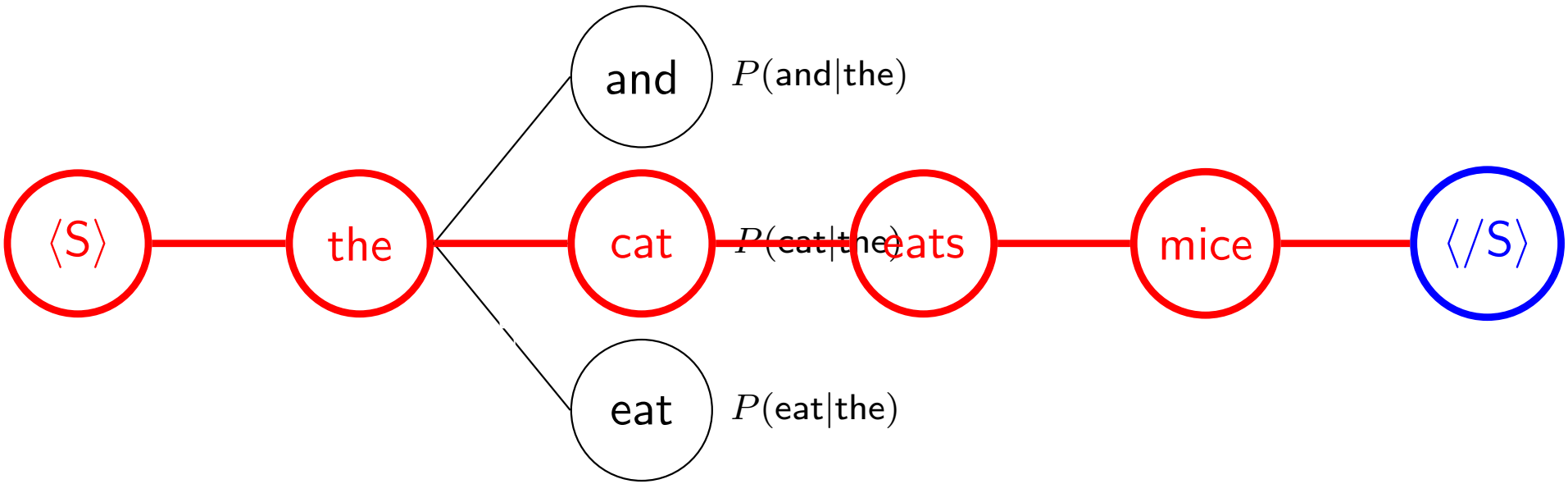
$$P(w_1 \dots w_n) \approx \prod_i P(w_i | w_{i-n+1} \dots w_{i-1})$$

We call these short sequences of words n-grams:

- bigrams ($n = 2$): *I want*, *want to*, *to go*, *go to*, *to the*, *the beach*
- trigrams ($n = 3$): *I want to*, *want to go*, *to go to*, *go to the*
- 4-grams ($n = 4$): *I want to go*, *want to go to*, *to go to the*

N-Gram Models - The Generative Story

- We can think of an n-gram model as a probabilistic automaton that generates sentences.



$$P(S) = P(\text{the}|\langle S\rangle)\ P(\text{cat}|\text{the})\ P(\text{eats}|\text{cat})\ P(\text{mice}|\text{eats})\ P(\langle/S\rangle|\text{mice})$$

N-Gram Models

An n-gram language model records the n-gram conditional probabilities:

$$\begin{aligned} P(I|\langle S\rangle) &= 0.0429 & P(to|go) &= 0.1540 \\ P(want|I) &= 0.0111 & P(the|to) &= 0.1219 \\ P(to|want) &= 0.4810 & P(beach|the) &= 0.0006 \\ P(go|to) &= 0.0131 \end{aligned}$$

We calculate the probability of a sentence as (assuming bi-grams):

$$\begin{aligned} P\left(w_1^n\right) &\approx \prod_{k=i}^{n} P\left(w_k|w_{k-1}\right) \\ &\approx P\left(I|\langle S\rangle\right) \times P\left(want|I\right) \times P\left(to|want\right) \times P\left(go|to\right) \times P\left(to|go\right) \times \\ &\quad P\left(the|to\right) \times P\left(beach|the\right) \\ &\approx 0.0429 \times 0.0111 \times 0.4810 \times 0.0131 \times 0.1540 \times \\ &\quad 0.1219 \times 0.0006 = 3.38 \times 10^{-11} \end{aligned}$$

Training an N-Gram Model – MLE

How to estimate the probabilities of n-grams?

By counting:

$$P\left(w_i|w_{i-n+1},\ldots,w_{i-1}\right) = \frac{Count\left(w_{i-n+1},\ldots,w_i\right)}{Count(w_{i-n+1},\ldots,w_{i-1})}$$

E.g. for trigrams:

$$P\left(\text{go}|\text{want to}\right) = \frac{C\left(\text{want to go}\right)}{C\left(\text{want to}\right)}$$

The probabilities are estimated using the relative frequencies of observed outcomes. This process is called Maximum Likelihood Estimation (MLE).

- Suppose "Chinese" occurs 400 times in a corpus of 1M words
- MLE of "Chinese" is $\frac{400}{1000000} = 0.0004$
- Is this a good estimate for all corpora?

Bigram MLE Example

w_1	w_2	$C(w_1 w_2)$	$C(w_1)$	$P(w_2 \mid w_1)$
$\langle S \rangle$	I	1039	24243	0.0429
I	want	46	4131	0.0111
want	to	101	210	0.4810
to	go	128	9778	0.0131
go	to	59	383	0.1540
to	the	1192	9778	0.1219
the	beach	14	22244	0.0006

What's the probability of *Others want to go to the beach* ?

$$P(Others \mid \langle S \rangle) = ?$$

Problems with MLE of N-Grams

- Like many statistical models, n-grams is dependent on the training corpus
- Data sparseness: many perfectly acceptable n-grams will not be observed in the training data
- Zero counts will result in an estimated probability of 0

- But we still want to have good intuition about more likely sentences:
  - Others want to go to the beach.
  - Others the beach go to want to.

Remedy—Laplace Smoothing

- Reassign some of the probability mass of frequent events to less frequent (or unseen) events.
- Known as smoothing or discounting
- The simplest approach is Laplace ('add-one') smoothing:

$$P_{\mathrm{L}}\left(w_n|w_{n-1}\right) = \frac{C\left(w_{n-1}w_n\right)+1}{C\left(w_{n-1}\right)+V}$$

Bigram MLE Example with Laplace Smoothing

"Others want to go to the beach"

w_1	w_2	$C(w_1 w_2)$	$C(w_1)$	$P(w_2\|w_1)$	$P_L(w_2\|w_1)$
$\langle S \rangle$	I	1039	24243	0.0429	0.01934
$\langle S \rangle$	Others	17	24243	0.0007	0.00033
I	want	46	4131	0.0111	0.00140
Others	want	0	4131	0	0.00003
want	to	101	210	0.4810	0.00343
to	go	128	9778	0.0131	0.00328
go	to	59	383	0.1540	0.00201
to	the	1192	9778	0.1219	0.03035
the	beach	14	22244	0.0006	0.00029

$$P_{\mathrm{L}}(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n) + 1}{C(w_{n-1}) + 29534}$$

Practical Issues

S = I want to got the beach

$$
\begin{aligned}
P(S) &= P(I|\langle S\rangle) \times P(want|I) \times P(to|want) \times P(go|to) \times P(to|go) \times \\
&\quad P(the|to) \times P(beach|the) \\
&= 0.0429 \times 0.0111 \times 0.4810 \times 0.0131 \times 0.1540 \times \\
&\quad 0.1219 \times 0.0006 = 3.38 \times 10^{-11}
\end{aligned}
$$

- Multiplying many small probabilities $\rightarrow$ Risk underflow
- Solution: work in log(arithmic) space:
  - $\log(AB) = \log(A) + \log(B)$
  - hence $P(A)P(B) = \exp(\log(A) + \log(B))$
  - $\log(P(S)) = -1.368 + -1.954 + -0.317 + -1.882 \ldots$

N-Gram Summary

- The likelihood of the next word depends on its context.
- We can calculate this using the chain rule:

$$P\left(w_1^N\right) = \prod_{i=1}^{N} P\left(w_i | w_1^{i-1}\right)$$

- In an n-gram model, we approximate this with a Markov chain:

$$P\left(w_1^N\right) \approx \prod_{i=1}^{N} P\left(w_i | w_{i-n+1}^{i-1}\right)$$

- We use Maximum Likelihood Estimation to estimate the conditional probabilities.
- Smoothing techniques are used to avoid zero probabilities.