Recap of the last lecture

Regular Expression (RE)

- A standard notation of characterizing a text sequence
- How can we search for any of the following:
  - woodchuck
  - woodchucks
  - Woodchuck
  - Woodchucks

- RE search requires a pattern and a corpus of texts to search through.

Morphology: Definition

The study of words, how they are formed, and their relationship to other words in the same language.

The Porter Stemmer (Porter, 1980)

- A simple rule-based algorithm for stemming
- An example of a HEURISTIC method
- Based on rules like:
  - ATIONAL -> ATE (e.g., *relational* -> *relate*)
- The algorithm consists of seven sets of rules, applied in order

The Edit Distance Table

5	L	5	6	7	6	**5**
4	A	4	5	6	5	6
3	I	3	4	5	4	5
2	R	2	3	4	5	6
1	T	1	2	3	4	5
0	#	0	1	2	3	4
		#	Z	E	I	L
i	j	0	1	2	3	4

$$D(i,j) = \min \begin{cases} D(i-1,j) + 1 \\ D(i,j-1) + 1 \\ D(i-1,j-1) + \begin{cases} 2; & \text{if } S_1(i) \neq S_2(j) \\ 0; & \text{if } S_1(i) = S_2(j) \end{cases} \end{cases}$$

Language Model
N-Gram

Assumption: we don’t know what is Deep Learning

Word Prediction

- Guess the next word...

  ... I notice three guys standing on the ???

- There are many sources of knowledge that could be helpful, including world knowledge.
- But we can do pretty well by simply looking at the preceding words and keeping track of simple counts.

Word Prediction

- Formalize this task using *N*-gram models.
- *N*-grams are token sequences of length *N*.
- Given N-grams counts, we can guess likely next words in a sequence.

Probabilistic Language Models

The goal: assign a probability to a sentence

- Machine Translation:
  - P(**high** winds tonite) > P(**large** winds tonite)
- Spelling Correction
  - The office is about fifteen **minuets** from my house
    - P(about fifteen **minutes** from) > P(about fifteen **minuets** from)
- Speech Recognition
  - P(I saw a van) >> P(eyes awe of an)
- + Summarization, question-answering, etc., etc.!!

Probabilistic Language Modeling

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

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

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

- Better: **the grammar** But **language model** or **LM** is standard

How to compute P(W)

- How to compute this joint probability:

P(its water is so transparent that the)=

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

P(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

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

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

- More variables:

  $P(A,B,C,D) = P(A)P(B|A)P(C|A,B)P(D|A,B,C)$

- The Chain Rule in General

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

The Chain Rule: joint probability 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(“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?

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

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

Markov Assumption

Andrei Markov

- Simplifying assumption:

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

- or maybe

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

Markov Assumption

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

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

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

Simplest case: Unigram model

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

Some automatically generated sentences from a unigram model

```
fifth, an, of, futures, the, an, incorporated, a, a,
the, inflation, most, dollars, quarter, in, is, mass

thrift, did, eighty, said, hard, 'm, july, bullish

that, or, limited, the
```

Quiz

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

- P(I like ice cream)
- P(the the the the)
- P(Go to class daily)
- P(class daily go to)

Bigram model

- Condition on the previous word:

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

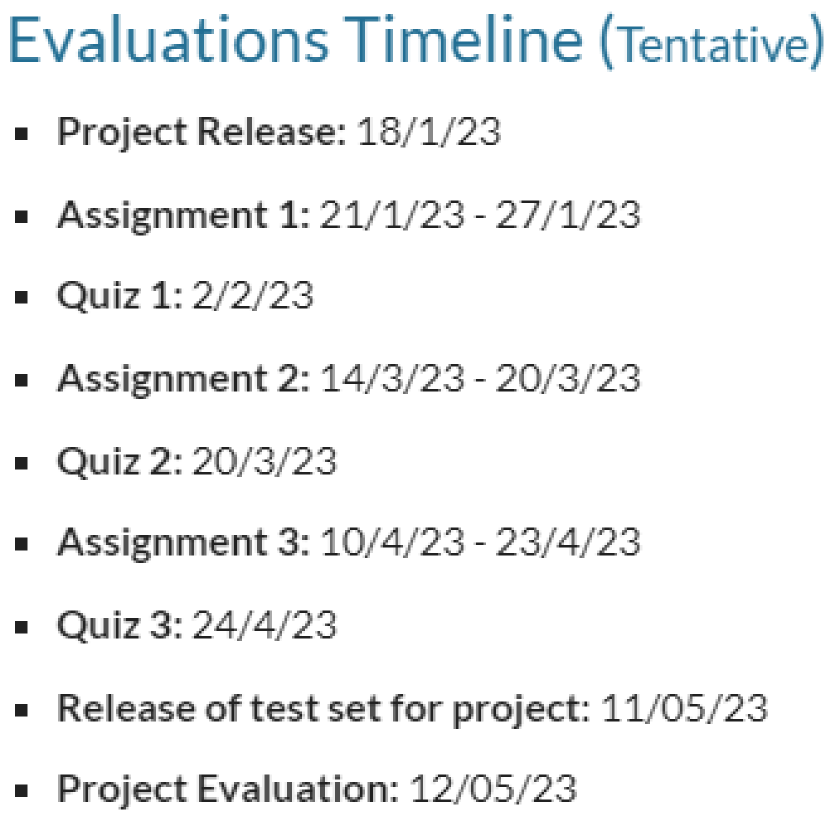
```
texaco, rose, one, in, this, issue, is, pursuing,
growth, in, a, boiler, house, said, mr., gurria, mexico,
's, motion, control, proposal, without, permission,
from, five, hundred, fifty, five, yen

outside, new, car, parking, lot, of, the, agreement,
reached

this, would, be, a, record, november
```

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



Evaluations Timeline (Tentative)

- **Project Release:** 18/1/23
- **Assignment 1:** 21/1/23 - 27/1/23
- **Quiz 1:** 2/2/23
- **Assignment 2:** 14/3/23 - 20/3/23
- **Quiz 2:** 20/3/23
- **Assignment 3:** 10/4/23 - 23/4/23
- **Quiz 3:** 24/4/23
- **Release of test set for project:** 11/05/23
- **Project Evaluation:** 12/05/23

Sign up!

Piazza: https://piazza.com/iitd.ac.in/spring2023/ell881

https://sites.google.com/view/ell881-iitd/home

Estimating N-gram Probabilities

Estimating bigram probabilities

- **The Maximum Likelihood Estimate (MLE)**

$$P(w_i \mid w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

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

An example

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

<s> I am Sam </s>

<s> Sam I am </s>

<s> I do not like green eggs and ham </s>

$$P(\texttt{I} \mid \texttt{<s>}) = \tfrac{2}{3} = .67 \qquad P(\texttt{Sam} \mid \texttt{<s>}) = \tfrac{1}{3} = .33 \qquad P(\texttt{am} \mid \texttt{I}) = \tfrac{2}{3} = .67$$

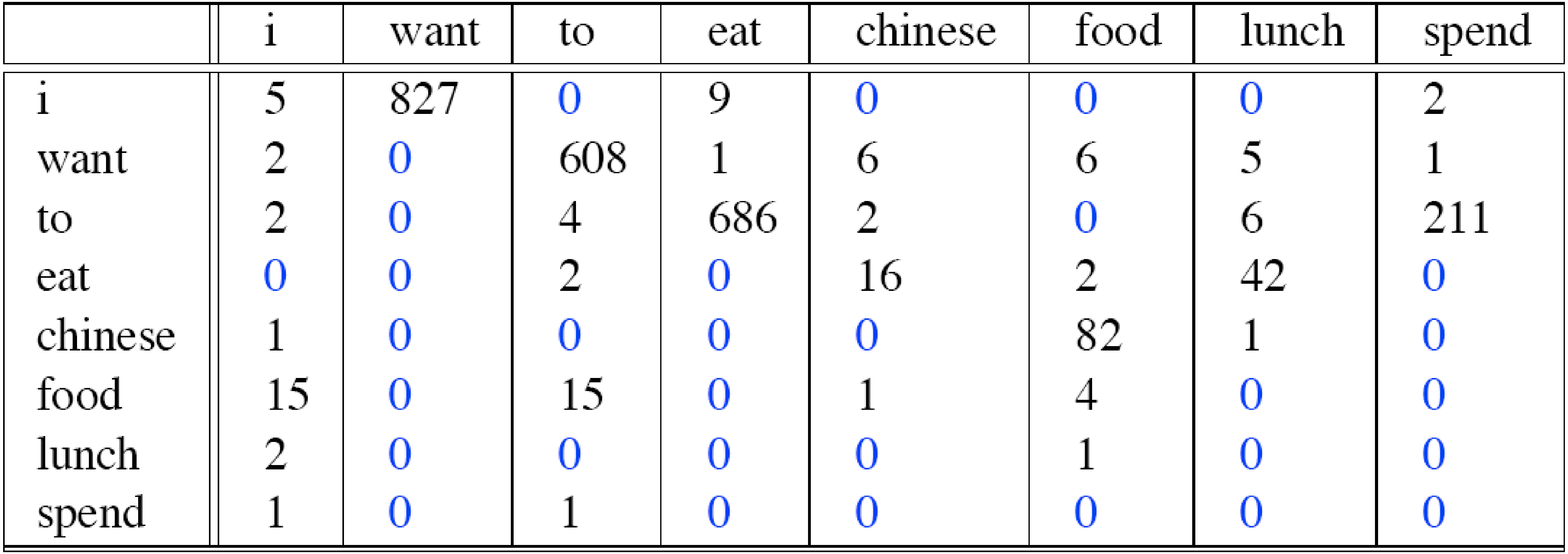
$$P(\texttt{</s>} \mid \texttt{Sam}) = \tfrac{1}{2} = 0.5 \qquad P(\texttt{Sam} \mid \texttt{am}) = \tfrac{1}{2} = .5 \qquad P(\texttt{do} \mid \texttt{I}) = \tfrac{1}{3} = .33$$

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)

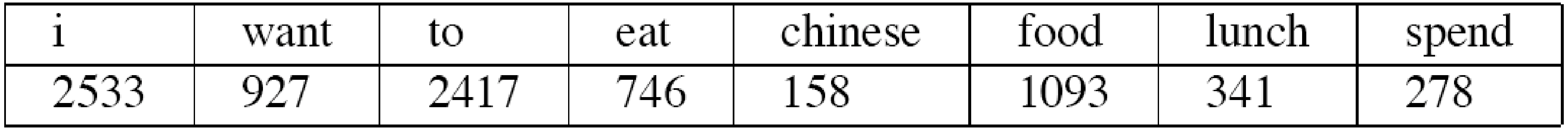
- Out of 9222 sentences



	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

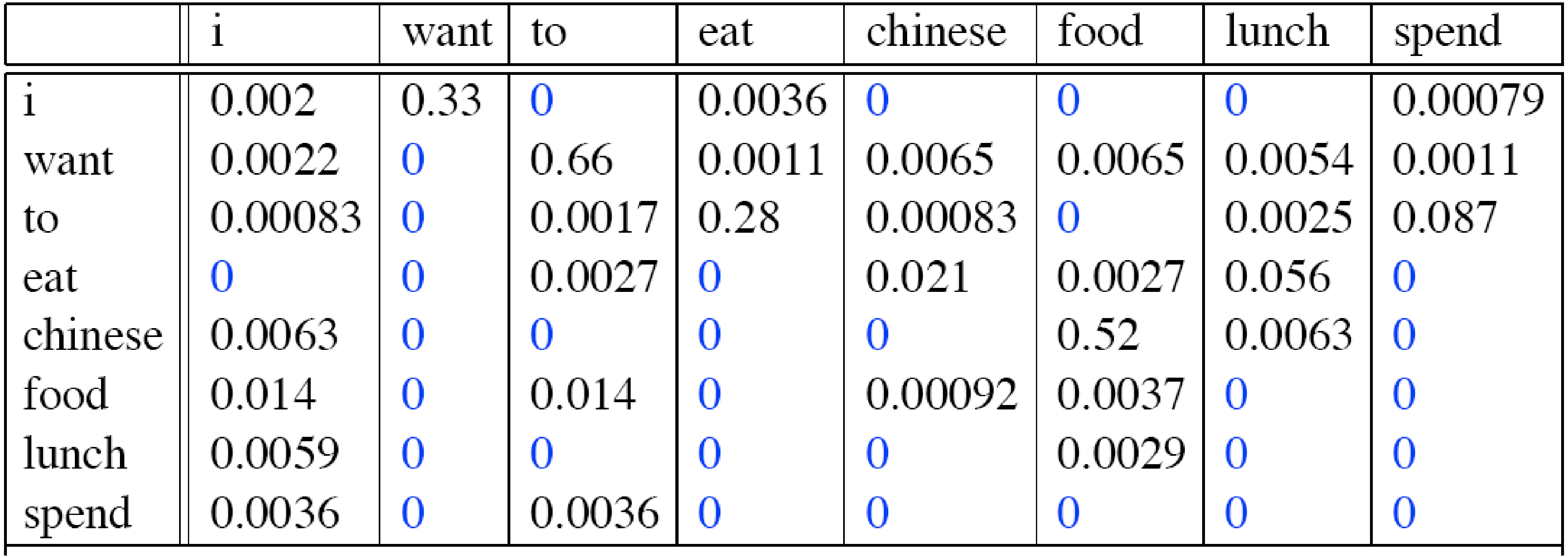
Raw bigram probabilities (relative measure)

- Normalize by unigrams:



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

- Result:



	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

$$
\begin{aligned}
&P(\text{<s> I want english food </s>}) = \\
&\quad P(\text{I}|\text{<s>}) \\
&\quad \times\ P(\text{want}|\text{I}) \\
&\quad \times\ P(\text{english}|\text{want}) \\
&\quad \times\ P(\text{food}|\text{english}) \\
&\quad \times\ P(\text{</s>}|\text{food}) \\
&\qquad = .000031
\end{aligned}
$$

What kinds of knowledge?

what kind of knowledge does this language model encode =>
things about the world (chinese foold is more likely than english foold)
facts about grammar => some sequence are grammatocally correct / incorrect

- P(english|want) = .0011
- P(chinese|want) = .0065
- P(to|want) = .66
- P(eat | to) = .28
- P(food | to) = 0
- P(want | spend) = 0
- P (i | <s>) = .25

world

grammar

contingent zero => zero by chance (didn't observe)
structurally => by structure of the language such sequences do not exist

contingent => subject to chance

grammar (contingent zero)

grammar (structural zero)

The **right to food**, and its non variations, is a human
right protecting the right for people to feed themselves in dignity

Practical Issues

- We do everything in log space
  - Avoid underflow: multiplying extremely small numbers
  - Adding is faster than multiplying

$$p_1 \times p_2 \times p_3 \times p_4 \Rightarrow \log p_1 + \log p_2 + \log p_3 + \log p_4$$

Evaluation of a Language Model

Evaluation: How good is our model?

- Does our language model prefer good sentences to bad ones?
    - Assign higher probability to “real” or “frequently observed” sentences
        - Than “ungrammatical” or “rarely observed” sentences?
- We train parameters of our model on a **training set**.
- We test the model’s performance on data we haven’t seen.
    - A **test set** is an unseen dataset that is different from our training set, totally unused.
    - An **evaluation metric** tells us how well our model does on the test set.

Extrinsic evaluation of N-gram models

- Best evaluation for comparing models A and B
  - Put each model in a task
    - spelling corrector, speech recognizer, machine translation system
  - Run the task, get an accuracy for A and for B
    - How many misspelled words corrected properly
    - How many words translated correctly
  - Compare accuracy for A and B

Difficulty of extrinsic (in-vivo) evaluation of N-gram models

- Extrinsic evaluation
  - Time-consuming; can take days or weeks
- So instead
  - Sometimes use **intrinsic** evaluation: **perplexity**
  - Bad approximation
    - unless the test data looks **just** like the training data
    - So **generally only useful in pilot experiments**
  - But is helpful to think about.

Intuition of Perplexity

- How well can we predict the next word?

  I always order pizza with cheese and ____

  The 33rd President of the US was ____

  I saw a ____

  - Unigrams are terrible at this game. (Why?)

mushrooms 0.1

pepperoni 0.1

….

fried rice 0.0001

….

and 1e-100

- A better model
  - is one which assigns a higher probability to the word that actually occurs

Perplexity

The best language model is one that best predicts an unseen test set

- Gives the highest P(sentence)

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

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

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

Chain rule:

$$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_1 \ldots w_{i-1})}}$$

For bigrams:

$$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_{i-1})}}$$

Minimizing perplexity is the same as maximizing probability

Example

- How hard is the task of recognizing digits ‘0,1,2,3,4,5,6,7,8,9’
  - Perplexity 10
- How hard is recognizing (30,000) names at Microsoft.
  - Perplexity = 30,000
- Perplexity is weighted equivalent branching factor (number of possible children)

Perplexity as branching factor

- Let’s suppose a sentence consisting of random digits
- What is the perplexity of this sentence according to a model that assign P=1/10 to each digit?

$$
\begin{aligned}
\mathrm{PP}(W) &= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \\
&= (\frac{1}{10}^{N})^{-\frac{1}{N}} \\
&= \frac{1}{10}^{-1} \\
&= 10
\end{aligned}
$$

Quiz

A traffic signal has three colors: green, yellow, and red, which appear with the following probabilities. Using a unigram model, **what is the perplexity of the sequence (green, yellow, red)?**

P(green) = 2/5

P(yellow) = 1/5

P(red) = 2/5

$$PP(green, yellow, red) = \left(\frac{2}{5} \times \frac{1}{5} \times \frac{2}{5}\right)^{-\frac{1}{3}}$$

Quiz

- The number zero is really frequent and occurs 10 times more often than other numbers.
- What is the perplexity?

Lower perplexity = better model

- Training 38 million words, test 1.5 million words, WSJ

N-gram Order	Unigram	Bigram	Trigram
Perplexity	962	170	109