Language Technologies Institute

Carnegie Mellon University

Algorithms for NLP

CS 11711, Fall 2019

Lecture 2: Language Models

Yulia Tsvetkov

Announcements

- Homework 1 released on 9/3
  - you need to attend next lecture to understand it
  - Chan will give an overview in the end of the next lecture
  - + recitation on 9/6

1-slide review of probability

Event space (e.g., $\mathcal{X}$, $\mathcal{Y}$)—in this class, usually discrete

Random variables (e.g., X, Y)

Slide credit: Noah Smith

1-slide review of probability

Event space (e.g., $\mathcal{X}$, $\mathcal{Y}$)—in this class, usually discrete

Random variables (e.g., X, Y)

Typical statement: "random variable X takes value $x \in \mathcal{X}$ with probability $p(X = x)$, or, in shorthand, $p(x)$"

Slide credit: Noah Smith

1-slide review of probability

Event space (e.g., $\mathcal{X}$, $\mathcal{Y}$)—in this class, usually discrete

Random variables (e.g., X, Y)

Typical statement: "random variable X takes value $x \in \mathcal{X}$ with probability $p(X = x)$, or, in shorthand, $p(x)$"

Joint probability: $p(X = x, Y = y)$

Slide credit: Noah Smith

1-slide review of probability

Event space (e.g., $\mathcal{X}$, $\mathcal{Y}$)—in this class, usually discrete

Random variables (e.g., X, Y)

Typical statement: "random variable X takes value $x \in \mathcal{X}$ with probability $p(X = x)$, or, in shorthand, $p(x)$"

Joint probability: $p(X = x, Y = y)$

Conditional probability: $p(X = x \mid Y = y) = \frac{p(X = x, Y = y)}{p(Y = y)}$

Slide credit: Noah Smith

1-slide review of probability

Event space (e.g., $\mathcal{X}$, $\mathcal{Y}$)—in this class, usually discrete

Random variables (e.g., X, Y)

Typical statement: "random variable X takes value $x \in \mathcal{X}$ with probability $p(X = x)$, or, in shorthand, $p(x)$"

Joint probability: $p(X = x, Y = y)$

Conditional probability: $p(X = x \mid Y = y) = \frac{p(X = x, Y = y)}{p(Y = y)}$

Always true:

$$p(X = x, Y = y) = p(X = x \mid Y = y) \cdot p(Y = y) = p(Y = y \mid X = x) \cdot p(X = x)$$

Slide credit: Noah Smith

1-slide review of probability

Event space (e.g., $\mathcal{X}$, $\mathcal{Y}$)—in this class, usually discrete

Random variables (e.g., X, Y)

Typical statement: "random variable X takes value $x \in \mathcal{X}$ with probability $p(X = x)$, or, in shorthand, $p(x)$"

Joint probability: $p(X = x, Y = y)$

Conditional probability: $p(X = x \mid Y = y) = \dfrac{p(X = x, Y = y)}{p(Y = y)}$

Always true:
$p(X = x, Y = y) = p(X = x \mid Y = y) \cdot p(Y = y) = p(Y = y \mid X = x) \cdot p(X = x)$

Sometimes true: $p(X = x, Y = y) = p(X = x) \cdot p(Y = y)$

Slide credit: Noah Smith


elijah wood
everything is illuminated

My legal name is Alexander Perchov.

My legal name is Alexander Perchov. But all of my many friends dub me Alex, because that is a more flaccid-to-utter version of my legal name.

My legal name is Alexander Perchov. But all of my many friends dub me Alex, because that is a more flaccid-to-utter version of my legal name. Mother dubs me Alexi-stop-spleening-me!, because I am always spleening her.

My legal name is Alexander Perchov. But all of my many friends dub me Alex, because that is a more flaccid-to-utter version of my legal name. Mother dubs me Alexi-stop-spleening-me!, because I am always
spleening her. If you want to know why I am always spleening her, it is because I am always elsewhere with friends, and disseminating so much currency, and performing so many things that can spleen a mother.

My legal name is Alexander Perchov. But all of my many friends dub me Alex, because that is a more flaccid-to-utter version of my legal name. Mother dubs me Alexi-stop-spleening-me!, because I am always
spleening her. If you want to know why I am always spleening her, it is because I am always elsewhere with friends, and disseminating so much currency, and performing so many things that can spleen a mother. Father used to dub me Shapka, for the fur hat I would don even in the summer month.

My legal name is Alexander Perchov. But all of my many friends dub me Alex, because that is a more flaccid-to-utter version of my legal name. Mother dubs me Alexi-stop-spleening-me!, because I am always
spleening her. If you want to know why I am always spleening her, it is because I am always elsewhere with friends, and disseminating so much currency, and performing so many things that can spleen a mother. Father used to dub me Shapka, for the fur hat I would don even in the summer month. He ceased dubbing me that because I ordered him to cease dubbing me that. It sounded boyish to me, and I have always thought of myself as very potent and generative.

Language models play the role of ...

- a judge of grammaticality
- a judge of semantic plausibility
- an enforcer of stylistic consistency
- a repository of knowledge (?)

The Language Modeling problem

- Assign a probability to every sentence (or any string of words)
  - finite vocabulary (e.g. words or characters) {*the, a, telescope, …*}
  - infinite set of sequences
    - *a telescope* STOP
    - a STOP
    - the the the STOP
    - I saw a woman with a telescope STOP
    - STOP
    - ...

The Language Modeling problem

- Assign a probability to every sentence (or any string of words)
  - finite vocabulary (e.g. words or characters)
  - infinite set of sequences

$$\sum_{\mathbf{e} \in \Sigma^*} p_{\mathrm{LM}}(\mathbf{e}) = 1$$

$$p_{\mathrm{LM}}(\mathbf{e}) \geq 0 \quad \forall \mathbf{e} \in \Sigma^*$$

p(***disseminating so much currency*** STOP) = 10^{-15}

p(***spending a lot of money*** STOP) = 10^{-9}

The Language Modeling problem

- Assign a probability to every sentence (or any string of words)
  - finite vocabulary (e.g. words or characters)
  - infinite set of sequences

$$\sum_{\mathbf{e} \in \Sigma^*} p_{\mathrm{LM}}(\mathbf{e}) = 1$$

$$p_{\mathrm{LM}}(\mathbf{e}) \geq 0 \quad \forall \mathbf{e} \in \Sigma^*$$

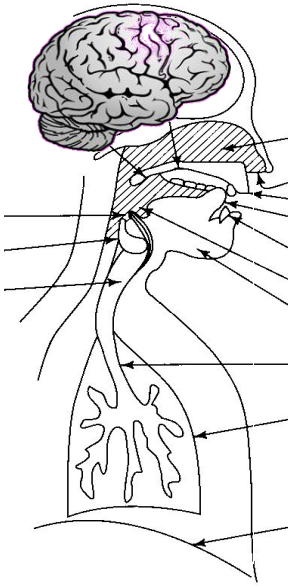
Objections?

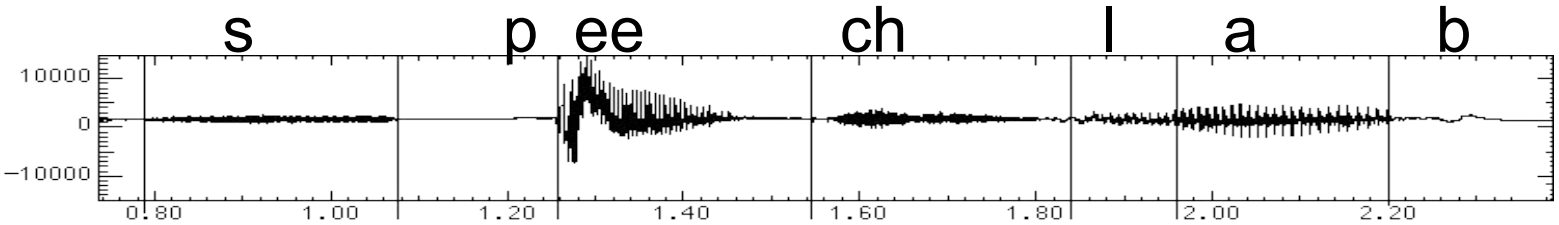
Motivation

- Machine translation
  - p(**strong** winds) > p(**large** winds)
- Spell 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, handwriting recognition, OCR, etc.

Motivation

- Speech recognition: we want to predict a sentence given acoustics





Motivation

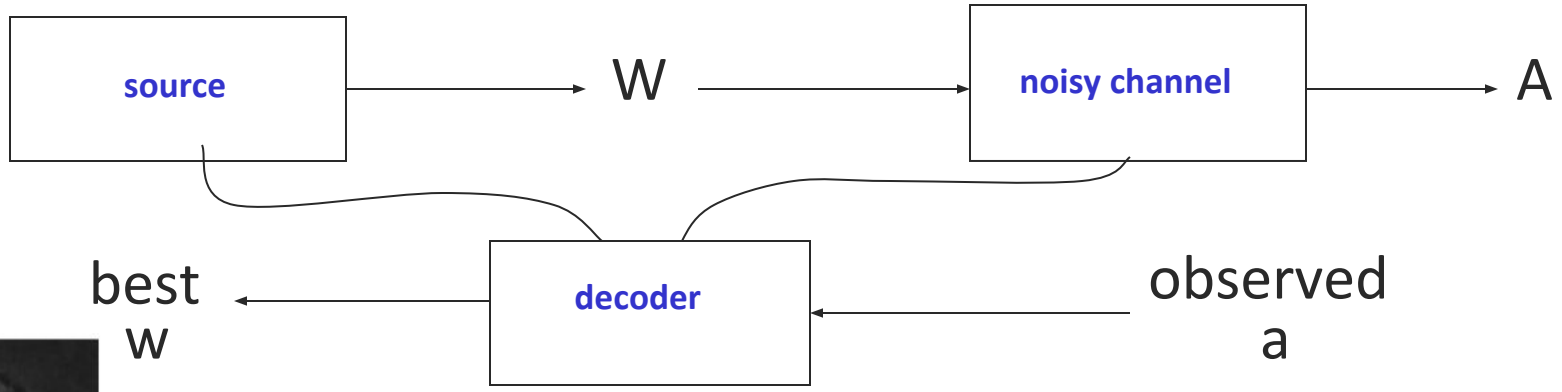
- Speech recognition: we want to predict a sentence given acoustics

the station signs are in deep in english	-14732
the stations signs are in deep in english	-14735
the station signs are in deep into english	-14739
the station 's signs are in deep in english	-14740
the station signs are in deep in the english	-14741
the station signs are indeed in english	-14757
the station 's signs are indeed in english	-14760
the station signs are indians in english	-14790
the station signs are indian in english	-14799
the stations signs are indians in english	-14807
the stations signs are indians and english	-14815

Motivation: the Noisy-Channel Model

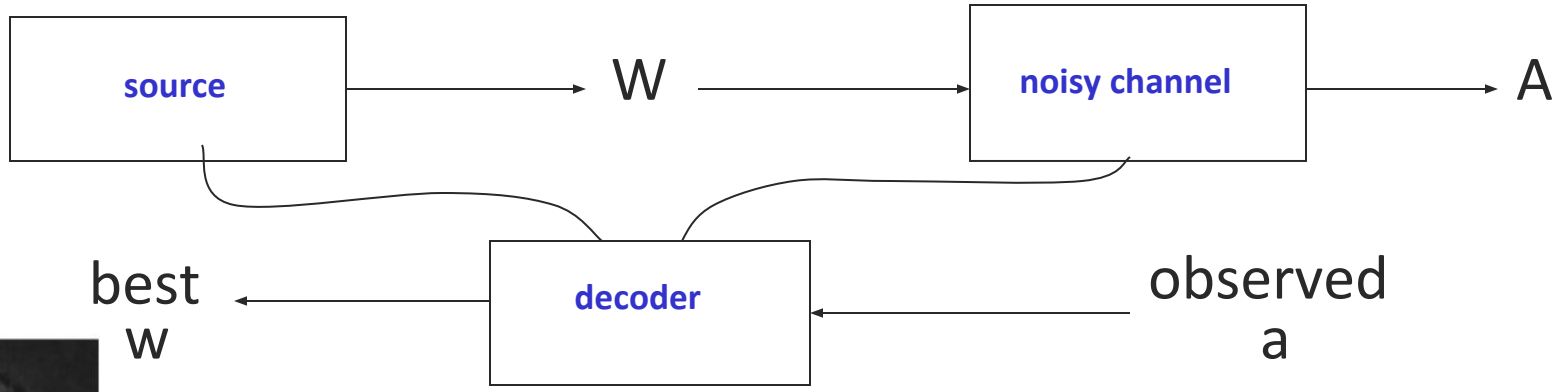
Claude Shannon. "A Mathematical Theory of Communication" 1948.

Motivation: the Noisy-Channel Model



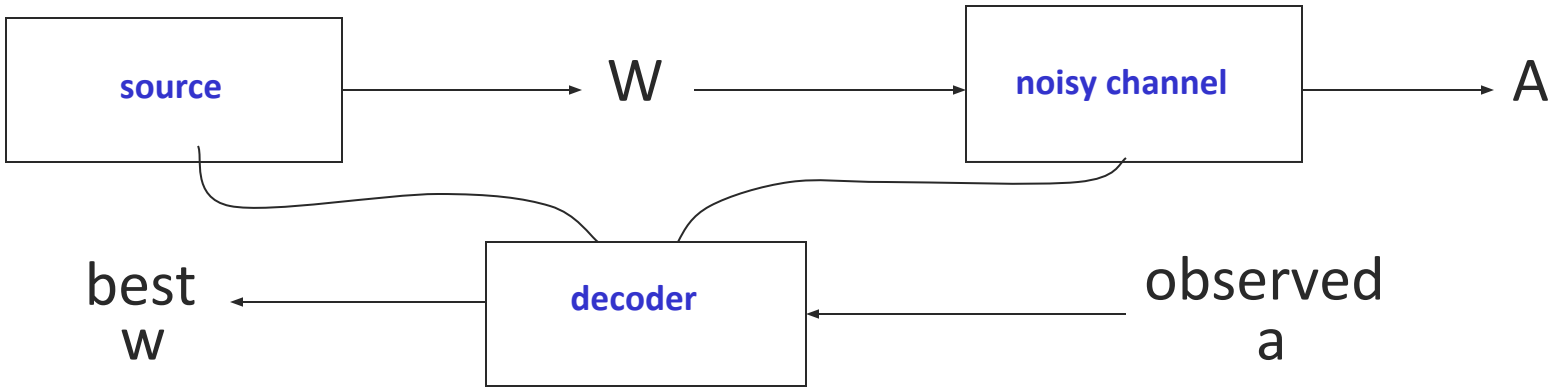
Claude Shannon. "A Mathematical Theory of Communication" 1948.

Motivation: the Noisy-Channel Model



Claude Shannon. "A Mathematical Theory of Communication" 1948.

Motivation: the Noisy-Channel Model



- We want to predict a sentence given acoustics:

$$w^* = \arg\max_w P(w|a)$$

Motivation: the Noisy-Channel Model

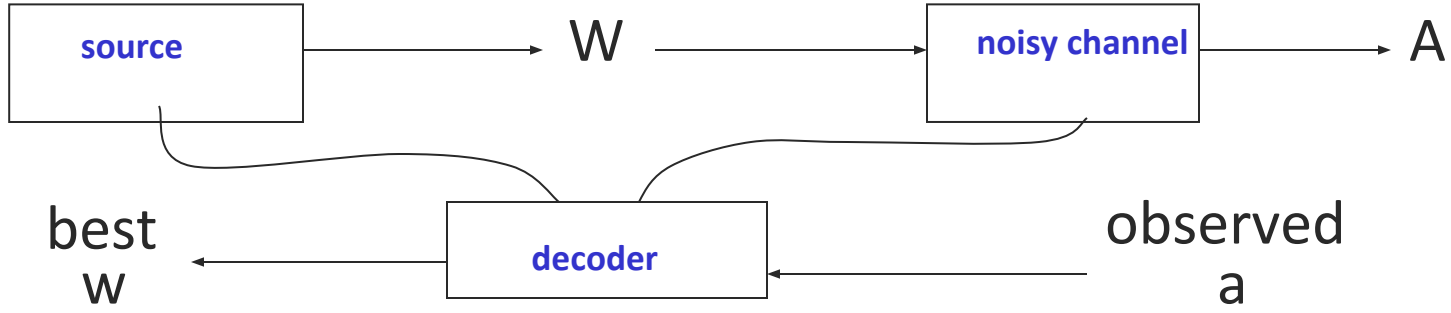
- We want to predict a sentence given acoustics:

$$w^* = \arg\max_w P(w|a)$$

- The noisy-channel approach:

$$w^* = \arg\max_w P(w|a)$$
$$= \arg\max_w P(a|w)P(w)/P(a)$$

Motivation: the Noisy-Channel Model



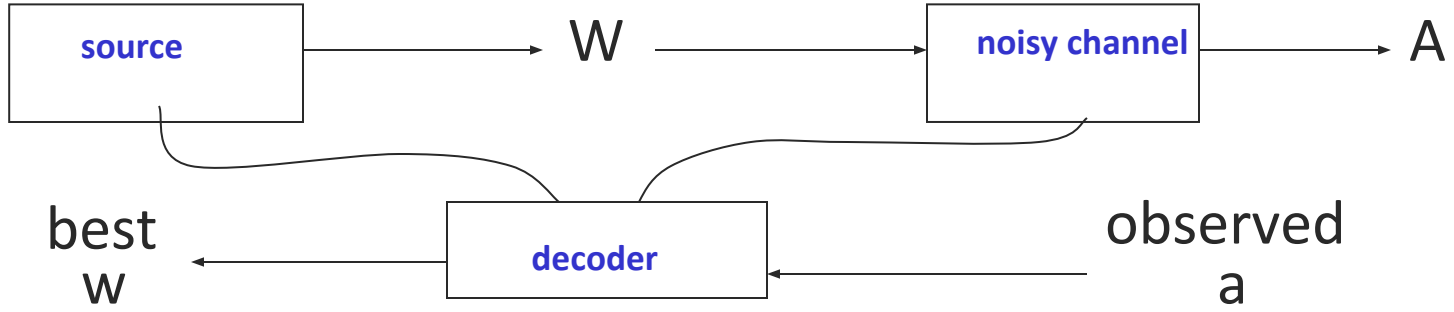
- The noisy-channel approach:

$$w^* = \arg\max_w P(w|a)$$
$$= \arg\max_w P(a|w)P(w)/P(a)$$
$$= \arg\max_w P(a|w)P(w)$$

channel model

source model

Motivation: the Noisy-Channel Model



- The noisy-channel approach:

$$w^* = \arg\max_w P(w|a)$$
$$= \arg\max_w P(a|w)P(w)/P(a)$$
$$= \arg\max_w P(a|w)P(w)$$

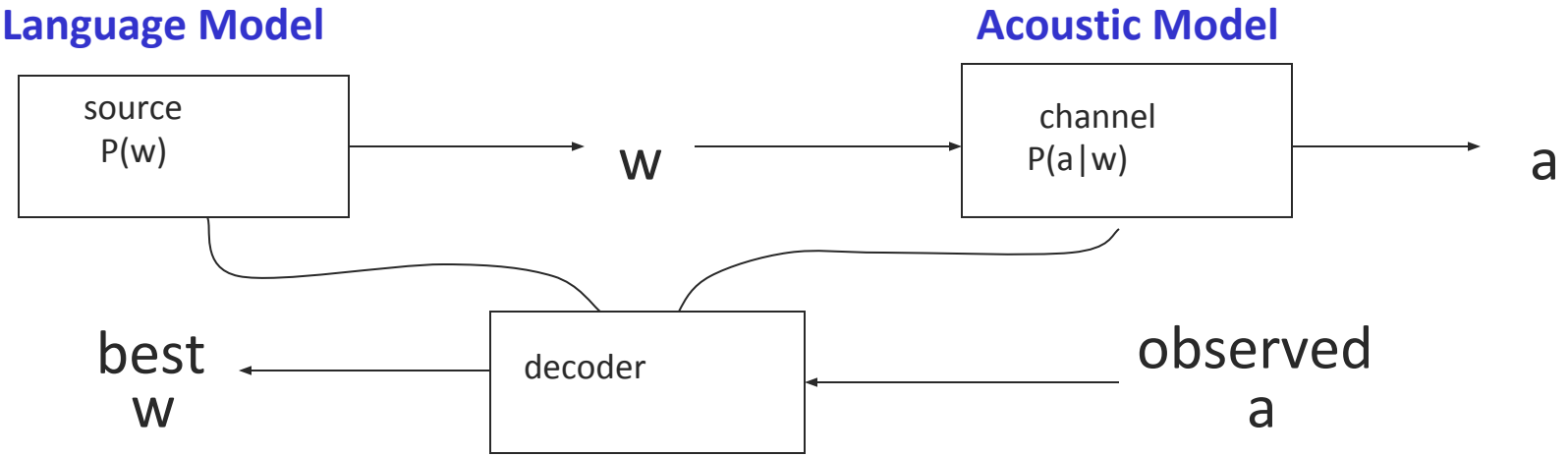
Likelihood

Prior

Acoustic model (HMMs)

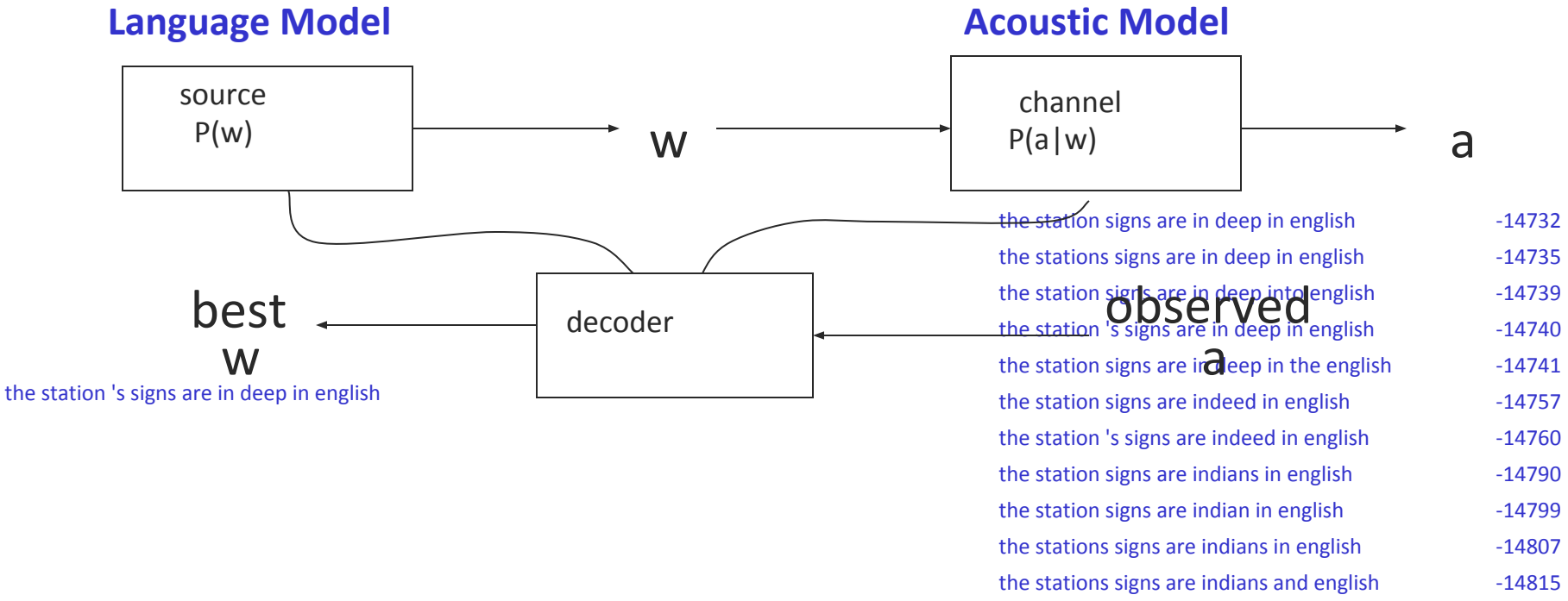
Language model: Distributions over sequences of words (sentences)

Noisy channel example: Automatic Speech Recognition

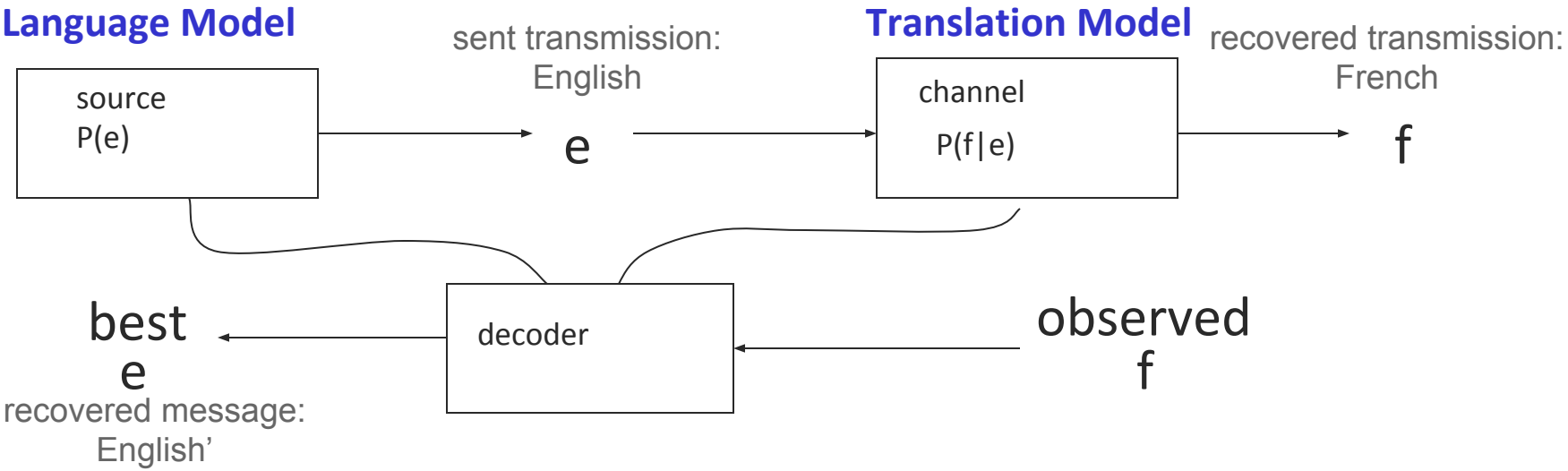


$$\underset{w}{\operatorname{argmax}}\, P(w|a) = \underset{w}{\operatorname{argmax}}\, P(a|w)P(w)$$

Noisy channel example: Automatic Speech Recognition



Noisy channel example: Machine Translation



$$\underset{e}{\text{argmax}}\, P(e|f) = \underset{e}{\text{argmax}}\, P(f|e)P(e)$$

One naturally wonders if the problem of translation could conceivably be treated as a problem in cryptography. When I look at an article in Russian, I say: '***This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.***'

Warren Weaver to Norbert Wiener, March, 1947

Noisy Channel Examples

- speech recognition
- machine translation
- optical character recognition
- spelling and grammar correction
- handwriting recognition
- document summarization
- dialog generation
- linguistic decipherment
- etc.

Plan

- ~~what is language modeling~~
- ~~motivation~~
- how to build an *n*-gram LMs
- how to estimate parameters from training data (*n*-gram probabilities)
- how to evaluate (perplexity)
- how to select vocabulary, what to do with OOVs (smoothing)

The Language Modeling problem

- Assign a probability to every sentence (or any string of words)
  - finite vocabulary (e.g. words or characters)
  - infinite set of sequences

$$\sum_{\mathbf{e} \in \Sigma^*} p_{\mathrm{LM}}(\mathbf{e}) = 1$$

$$p_{\mathrm{LM}}(\mathbf{e}) \geq 0 \quad \forall \mathbf{e} \in \Sigma^*$$

A trivial model

- Assume we have *n* training sentences
- Let $x_1, x_2, \ldots, x_n$ be a sentence, and $c(x_1, x_2, \ldots, x_n)$ be the number of times it appeared in the training data.
- Define a language model:

$$p(x_1, \ldots, x_n) = \frac{c(x_1, \ldots, x_n)}{N}$$

A trivial model

- Assume we have n training sentences
- Let $x_1, x_2, \ldots, x_n$ be a sentence, and $c(x_1, x_2, \ldots, x_n)$ be the number of times it appeared in the training data.
- Define a language model:

$$p(x_1, \ldots, x_n) = \frac{c(x_1, \ldots, x_n)}{N}$$

- No generalization!

Markov processes

- Markov processes:
  - Given a sequence of *n* random variables:
  - We want a sequence probability model

$$X_1, X_2, \ldots, X_n, \; n = 100, \; X_i \in \mathcal{V}$$
$$p(X_1 = x_1, X_2 = x_2, \cdots X_n = x_n)$$

Markov processes

- Markov processes:
  - Given a sequence of *n* random variables:
  - We want a sequence probability model

$$X_1, X_2, \ldots, X_n,\ n = 100,\ X_i \in \mathcal{V}$$
$$p(X_1 = x_1, X_2 = x_2, \cdots X_n = x_n)$$

  - There are $|V|^n$ possible sequences

First-order Markov process

Chain rule

$$p(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) =$$

$$p(X_1 = x_1) \prod_{i=2}^{n} p(X_i = x_i \mid X_1 = x_1, \ldots, X_{i-1} = x_{i-1})$$

First-order Markov process

Chain rule

$$p(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) =$$

$$p(X_1 = x_1) \prod_{i=2}^{n} p(X_i = x_i \mid X_1 = x_1, \ldots, X_{i-1} = x_{i-1})$$

Markov assumption

$$= \quad P(X_1 = x_1) \prod_{i=2}^{n} P(X_i = x_i | X_{i-1} = x_{i-1})$$

Second-order Markov process:

- Relax independence assumption:

$$p(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) =$$
$$p(X_1 = x_1) \times p(X_2 = x_2 \mid X_1 = x_1)$$
$$\times \prod_{i=3}^{n} p(X_i = x_i \mid X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1})$$

Second-order Markov process:

- Relax independence assumption:

$$p(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) =$$
$$p(X_1 = x_1) \times p(X_2 = x_2 \mid X_1 = x_1)$$
$$\times \prod_{i=3}^{n} p(X_i = x_i \mid X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1})$$

- Simplify notation:

$$x_0 = *, x_{-1} = *$$

Detail: variable length

- We want probability distribution over sequences of any length

Detail: variable length

- Probability distribution over sequences of any length
- Define always X_n=STOP, where STOP is a special symbol

Detail: variable length

- Probability distribution over sequences of any length
- Define always X_n=STOP, where STOP is a special symbol
- Then use a Markov process as before:

$$P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n) = \prod_{i=1}^{n} P(X_i = x_i | X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1})$$

- We now have probability distribution over all sequences
  - Intuition: at every step you have probability α_h to stop (conditioned on history) and (1-α_h) to keep going

1. Initialize $i = 1$, and $x_0 = x_{-1} = *$

2. Generate x_i from the distribution

$$P(X_i = x_i | X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1})$$

3. If x_i = STOP then return the sequence $x_1 \ldots x_i$. Otherwise, set $i = i + 1$ and return to step 2.

3-gram LMs

- A trigram language model contains
  - a vocabulary V
  - a non negative parameters q(*w*|*u,v*) for every trigram, such that

$$w \in \mathcal{V} \cup \{\text{STOP}\},\ u, v \in \mathcal{V} \cup \{*\}$$

  - the probability of a sentence x_1, …, x_n, where x_n=STOP is

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} q(x_i \mid x_{i-1}, x_{i-2})$$

Example

$$p(\text{the dog barks STOP}) =$$

Example

$$p(\text{the dog barks STOP}) = q(the \mid *, *) \times$$

Example

$$
\begin{aligned}
p(\text{the dog barks STOP}) = & q(the \mid *, *) \times \\
& q(dog \mid *, the) \times \\
& q(barks \mid the, dog) \times \\
& q(STOP \mid dog, barks) \times
\end{aligned}
$$

Limitations?

Limitation

- Markovian assumption is false

  He is from France, so it makes sense that his first language is…

- We would want to model longer dependencies

Plan

- ~~what is language modeling~~
- ~~motivation~~
- ~~how to build *n*-gram LMs~~
- how to estimate parameters from training data (*n*-gram probabilities)
- how to evaluate (perplexity)
- how to select vocabulary, what to do with OOVs (smoothing)

Empirical N-Grams

- How do we know P(w | history)?
  - Use statistics from data (examples using Google N-Grams)
  - E.g. what is P(door | the)?

Training Counts

198015222 the first
194623024 the same
168504105 the following
158562063 the world
…
14112454 the door

23135851162 the *

$$\hat{P}(\text{door}|\text{the}) = \frac{14112454}{23135851162}$$

$$= 0.0006$$

Increasing N-Gram Order

- Higher orders capture more dependencies

Bigram Model

198015222 the first
194623024 the same
168504105 the following
158562063 the world
…
14112454 the door

23135851162 the *

P(door | the) = 0.0006

Trigram Model

197302 close the window 191125
close the door 152500 close the
gap 116451 close the thread
87298 close the deal
…

3785230 close the *

P(door | close the) = 0.05

Berkeley restaurant project sentences

- can you tell me about any good cantonese restaurants close by
- mid priced that food is what i'm looking for
- tell me about chez pansies
- 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

Bigram counts (~10K 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

Bigram probabilities

	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 did we learn

- p(English | want) < p(Chinese | want) - people like Chinese stuff more when it comes to this corpus
- p(to | want) = 0.66 - English behaves in a certain way
- p(eat | to) = 0.28 - English behaves in a certain way

Sparseness

- Maximum likelihood for estimating q
  - Let c(w_1, …, w_n) be the number of times that *n*-gram appears in a corpus

$$q(w_i \mid w_{i-2}, w_{i-1}) = \frac{c(w_{i-2}, w_{i-1}, w_i)}{c(w_{i-2}, w_{i-1})}$$

  - If vocabulary has 20,000 words ⇒ Number of parameters is 8 x 10^{12}!

Sparseness

- Maximum likelihood for estimating q
  - Let $c(w_1, \ldots, w_n)$ be the number of times that *n*-gram appears in a corpus

$$q(w_i \mid w_{i-2}, w_{i-1}) = \frac{c(w_{i-2}, w_{i-1}, w_i)}{c(w_{i-2}, w_{i-1})}$$

  - If vocabulary has 20,000 words ⇒ Number of parameters is 8×10^{12}!
  - Most n-grams will never be observed, even if they are linguistically plausible (Zipf law)
  - ⇒ Most sentences will have zero or undefined probabilities

Plan

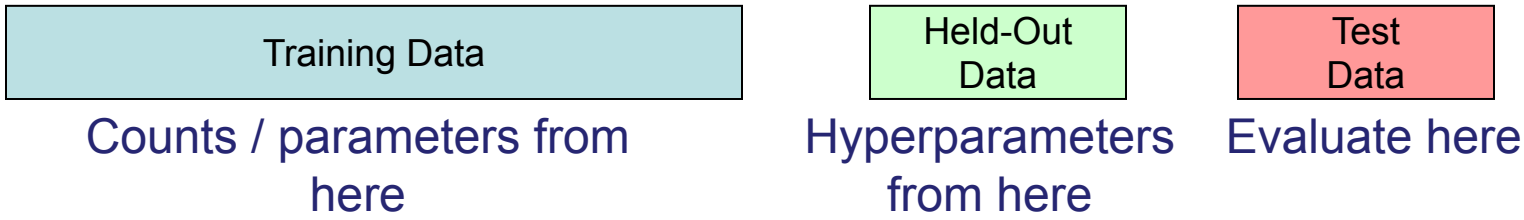
- ~~what is language modeling~~
- ~~motivation~~
- ~~how to build *n*-gram LMs~~
- ~~how to estimate parameters from training data (*n*-gram probabilities)~~
- how to evaluate (perplexity)
- how to select vocabulary, what to do with OOVs (smoothing)

Evaluation

- Extrinsic evaluation: build a new language model, use it for some task (MT, ASR, etc.)
- Intrinsic: measure how good we are at modeling language

Intrinsic evaluation

- Intuitively, language models should assign high probability to real language they have not seen before
  - Want to maximize likelihood on test, not training data
  - Models derived from counts / sufficient statistics require generalization parameters to be tuned on held-out data to simulate test generalization
  - Set hyperparameters to maximize the likelihood of the held-out data (usually with grid search or EM)



Evaluation: perplexity

- Test data: S = {s_1, s_2, …, s_{sent}}
  - Parameters are not estimated from S
  - Perplexity is the normalized inverse probability of S

$$p(\mathcal{S}) = \prod_{i=1}^{sent} p(s_i)$$

$$\log_2 p(\mathcal{S}) = \sum_{i=1}^{sent} \log_2 p(s_i)$$

$$\text{perplexity} = 2^{-l},\ l = \frac{1}{M} \sum_{i=1}^{sent} \log_2 p(s_i)$$

Evaluation: perplexity

- Test data: S = {s_1, s_2, …, s_{sent}}
  - parameters are estimated on training data

$$p(\mathcal{S}) = \prod_{i=1}^{sent} p(s_i)$$

$$\log_2 p(\mathcal{S}) = \sum_{i=1}^{sent} \log_2 p(s_i)$$

$$\text{perplexity} = 2^{-l},\ l = \frac{1}{M} \sum_{i=1}^{sent} \log_2 p(s_i)$$

  - *sent* is the number of sentences in the test data
  - M is the number of words in the test corpus
  - A good language model has high p(S) and low perplexity

Understanding perplexity

$$\text{perplexity} = 2^{-\frac{1}{M}\sum_{i=1}^{\text{sent}} \log_2 p(s_i)}$$

- It's a branching factor
  - assign probability of 1 to the test data ⇒ perplexity = 1
  - assign probability of 1/|V| to every word ⇒ perplexity = |V|
  - assign probability of 0 to anything ⇒ perplexity = ∞
    - this motivates the proper probability constraint $\sum_{\mathbf{e}\in\Sigma^*} p_{\text{LM}}(\mathbf{e}) = 1$
    - $p_{\text{LM}}(\mathbf{e}) \geq 0 \quad \forall \mathbf{e} \in \Sigma^*$
- cannot compare perplexities of LMs trained on different corpora

Typical values of perplexity

- When |V| = 50,000
- trigram model perplexity: 74 (<< 50,000)
- bigram model: 137
- unigram model: 955

Plan

- ~~what is language modeling~~
- ~~motivation~~
- ~~how to build *n*-gram LMs~~
- ~~how to estimate parameters from training data (*n*-gram probabilities)~~
- ~~how to evaluate (perplexity)~~
- how to select vocabulary, what to do with OOVs (smoothing)
  - better parameter estimation methods

Dealing with Out-of-Vocabulary terms

- Define a special OOV or “unknown” symbol `unk`. Transform some (or all) rare words in the training data to `unk`
  - You cannot fairly compare two language models that apply different `unk` treatments
- Build a language model at the character level

Dealing with sparsity: Smoothing

- For most N-grams, we have few observations
- General approach: modify observed counts to improve estimates
  - Discounting: allocate probability mass for unobserved events by discounting counts for observed events
  - Interpolation: approximate counts of N-gram using combination of estimates from related denser histories
  - Back-off: approximate counts of unobserved N-gram based on the proportion of back-off events (e.g., N-1 gram)

Bias-variance tradeoff

- Given a corpus of length M

Trigram model:

$$q(w_i \mid w_{i-2}, w_{i-1}) = \frac{c(w_{i-2}, w_{i-1}, w_i)}{c(w_{i-1}, w_i)}$$

Bigram model:

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

Unigram model:

$$q(w_i) = \frac{c(w_i)}{M}$$

Unigram

To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have
Every enter now severally so, let
Hill he late speaks; or! a more to leg less first you enter
Are where exeunt and sighs have rise excellency took of.. Sleep knave we. near; vile like

Bigram

What means, sir. I confess she? then all sorts, he is trim, captain.
Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow.
What we, hath got so she that I rest and sent to scold and nature bankrupt, nor the first gentleman?

Trigram

Sweet prince, Falstaff shall die. Harry of Monmouth's grave.
This shall forbid it should be branded, if renown made it empty.
Indeed the duke; and had a very good friend.
Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.

Quadrigram

King Henry.What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in;
Will you not tell me who I am?
It cannot be but so.
Indeed the short and the long. Marry, 'tis a noble Lepidus.

Linear interpolation

- Combine the three models to get all benefits

$$
\begin{aligned}
q_{LI}(w_i \mid w_{i-2}, w_{i-1}) &= \lambda_1 \times q(w_i \mid w_{i-2}, w_{i-1}) \\
&\quad + \lambda_2 \times q(w_i \mid w_{i-1}) \\
&\quad + \lambda_3 \times q(w_i) \\
\lambda_i \geq 0, \;\; \lambda_1 + \lambda_2 + \lambda_3 &= 1
\end{aligned}
$$

Linear interpolation

- Need to verify the parameters define a probability distribution

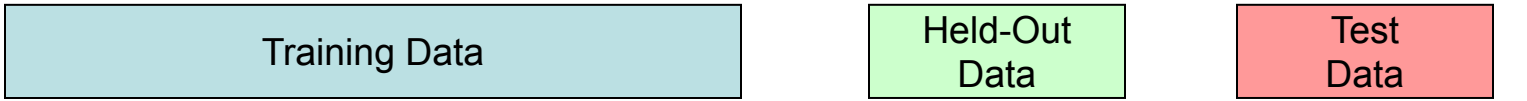
$$\sum_{w \in \mathcal{V}} q_{LI}(w \mid u, v)$$

$$= \sum_{w \in \mathcal{V}} \lambda_1 \times q(w \mid u, v) + \lambda_2 \times q(w \mid v) + \lambda_3 \times q(w)$$

$$= \lambda_1 \sum_{w \in \mathcal{V}} q(w \mid u, v) + \lambda_2 \sum_{w \in \mathcal{V}} q(w \mid v) + \lambda_3 \sum_{w \in \mathcal{V}} \times q(w)$$

$$= \lambda_1 + \lambda_2 + \lambda_3 = 1$$

Estimating coefficients



Counts / parameters from here

Hyperparameters from here

Evaluate here

Discounting methods

- Low count bigrams have high estimates

x	c(x)	$q(w_i \mid w_{i-1})$
the	48	
the, dog	15	15/48
the, woman	11	11/48
the, man	10	10/48
the, park	5	5/48
the, job	2	2/48
the, telescope	1	1/48
the, manual	1	1/48
the, afternoon	1	1/48
the, country	1	1/48
the, street	1	1/48

Discounting methods

$$c^*(x) = c(x) - 0.5$$

x	c(x)	c*(x)	$q(w_i \mid w_{i-1})$
the	48		
the, dog	15	14.5	14.5/48
the, woman	11	10.5	10.5/48
the, man	10	9.5	9.5/48
the, park	5	4.5	4.5/48
the, job	2	1.5	1.5/48
the, telescope	1	0.5	0.5/48
the, manual	1	0.5	0.5/48
the, afternoon	1	0.5	0.5/48
the, country	1	0.5	0.5/48
the, street	1	0.5	0.5/48

Discounting + Backoff

- next time: Kneser-Ney Smoothing

Next class

- KN smoothing
- Efficient LMs
  - relevant to your homework 1