Statistical NLP

CSE 256, Spring 2019

Lecture 2: Language Modeling (part 1)

Ndapa Nakashole, UCSD

3 April 2019

Announcements

- Reading for today
  - Michael Collins: language modeling notes

- Programming assignment one will be on TritonEd today

- Updated TA list
  - Farheen Ahluwalia [ddfahluwal@eng.ucsd.edu]
  - Jimmy Ye [jiy162@eng.ucsd.edu]
  - Sharathabhinav Dharmaji sdharmaj@ucsd.edu]
  - Tayal, Piyush [ptayal@ucsd.edu]

A Quick Review of Probability (1/2)

- **Sample Space:** (e.g., $\mathcal{X}$, $\mathcal{Y}$)
- **Random Variables:** (e.g., X, Y)
- **Typical Statement:**
  "random variable X takes value $x \in \mathcal{X}$ with probability $p(X = x)$"
  or simply, $p(x)$

A Quick Review of Probability (2/2)

- **Joint Probability:** $p(X = x, Y = y)$

- **Conditional Probability:** $p(X = x | Y = y)$
  $= \frac{p(X=x, Y=y)}{p(Y=y)}$

- **Always true:**
  $p(X = x, Y = y)$
  $= p(X = x | Y = y).p(Y = y)$
  $= p(Y = y | X = x).p(X = x)$

- **Sometimes true:**
  $p(X = x, Y = y)$
  $= p(X = x).p(Y = y)$

The language modeling problem

- $\mathcal{V}$ is a finite vocabulary

E.g., $\mathcal{V} = \{the, man, a, girl, telescope, park, \ldots\}$

- $\mathcal{V}^\dagger$ is a set of sequences

the man STOP

the man saw STOP

the the the STOP

the man saw the girl with the telescope STOP

The language modeling problem (continued)

- A language model is a probability distribution over sequences of words (sentences)

$$p : (\texttt{sequence of words}) \rightarrow \mathbb{R}$$

- $p(x) \geq 0$ for all $x \in \mathcal{V}^\dagger$
- $\sum_{x \in \mathcal{V}^\dagger} p(x) = 1$

- Language modeling:

  Estimate p from example sequences

The language modeling problem (continued)

- Input: $x_{1:n}$ ("training data")
- Output: $p : \mathcal{V}^\dagger \to \mathbb{R}^+$

- p should be a measure of plausibility (not grammaticality).

$$
\begin{aligned}
p(\text{the STOP}) &= 10^{-12} \\
p(\text{the man STOP}) &= 10^{-8} \\
p(\text{the man saw the STOP}) &= 2 \times 10^{-8} \\
p(\text{the girl saw saw STOP}) &= 10^{-15} \\
&\ldots \\
p(\text{the girl walked to the park STOP}) &= 2 \times 10^{-9} \\
&\ldots
\end{aligned}
$$

- <u>Why is this a useful thing to do?</u>

Noisy channel model

$$\boxed{\text{Source}} \longrightarrow \mathbf{D} \longrightarrow \boxed{\text{Channel}} \longrightarrow \mathbf{O}$$

$p(\boldsymbol{d}) \leftarrow$ source model

$p(\boldsymbol{o}|\boldsymbol{d}) \leftarrow$ channel model

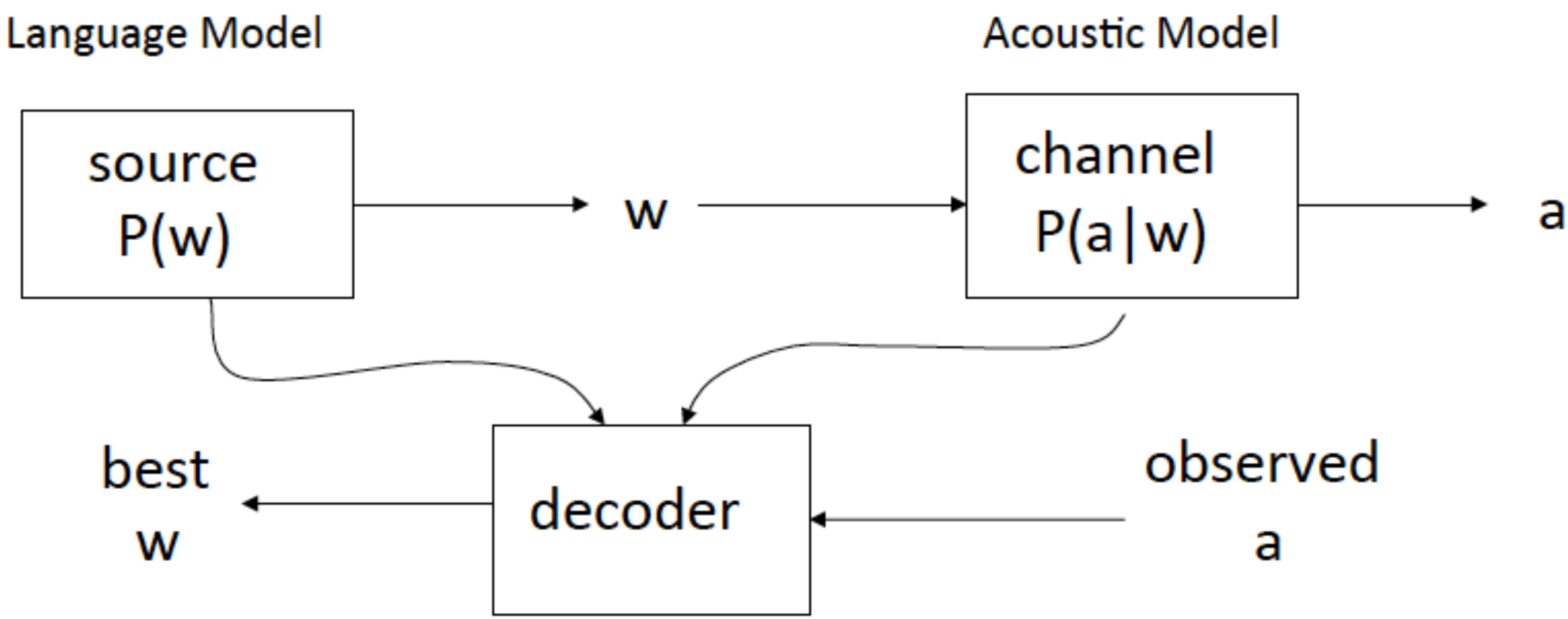
$$\begin{aligned}
\boldsymbol{d}^* &= \arg\max_{\boldsymbol{d}} p(\boldsymbol{d}|\boldsymbol{o}) \\
&= \arg\max_{\boldsymbol{d}} \frac{p(\boldsymbol{o}|\boldsymbol{d}) \cdot p(\boldsymbol{d})}{p(\boldsymbol{o})} \\
&= \arg\max_{\boldsymbol{d}} p(\boldsymbol{o}|\boldsymbol{d}) \cdot p(\boldsymbol{d})
\end{aligned}$$

When the source is language ….

- When the source is language
  - the source model is a **language model**

Noisy Channel Example: Speech Recognition

- Source generates sequence of words; channel produces sound waves.



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

Acoustic confusions: from the channel model p(a|w)

word sequence	$\log p(\text{acoustics} \mid \text{word sequence})$
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

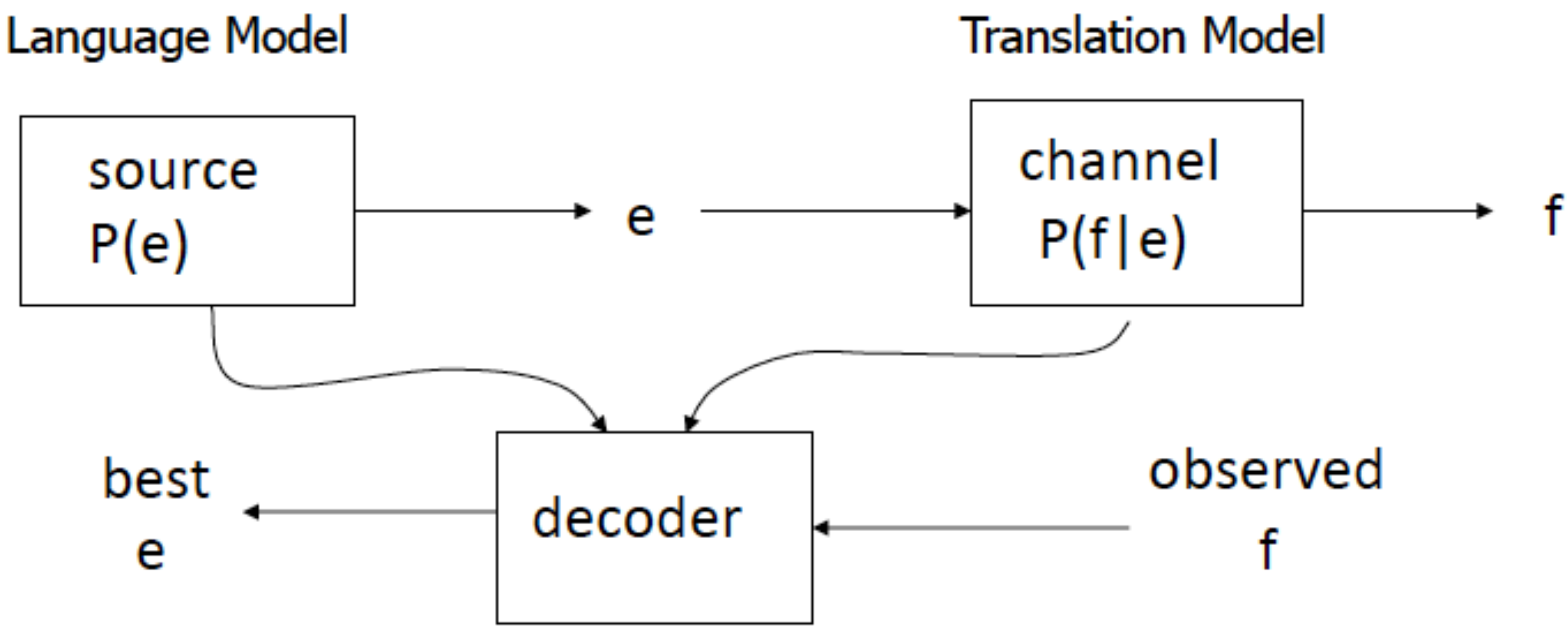
Translation: codebreaking?

"Also knowing nothing official about, but having guessed and inferred considerable about, the powerful new mechanized methods in cryptography—methods which I believe succeed even when one does not know what language has been coded—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 (1947)

Noisy Channel Example: Machine Translation

- Source: generates sequence of English words



$$\arg\max_{e} P(e|f) = \arg\max_{e} P(f|e)P(e)$$

Other noisy channel models

- Handwriting recognition, OCR
- Grammar/spelling correction
- Document summarization
- Dialog generation
- …

A Naive Language Model

- We have N training sentences
- For any sentence $x_1, \ldots, x_n$,

  - $c(x_1, \ldots x_n)$: $\leftarrow$ number of times the sentence is seen in our training data

- A naive estimate:

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

A Naive Language Model: is it a well-formed LM?

- It is a well-formed language model but …
- Assigns probability zero to any sentence not in the training data
- Need models that **generalize** to new test sentences
  - (i.e., sentences we have not seen before)

- The language modeling problem
- N-gram models

Markov Processes

- We have sequence of random variables $X_1, X_2, \ldots, X_n$.
  - -each random variable can take any value in a finite set $\mathcal{V}$
  - -for now we assume the length n is fixed (e.g., $n = 100$).

- Our goal is to model the joint probability:

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

$|\mathcal{V}|^n$ possible sequences of of length n!

Chain Rule

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

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

$$P(X_1 = x_1) \cdot P(X_2 = x_2 | X_1 = x_1) \cdot P(PX_3 = x_3 | X_2 = x_2, X_1 = x_1) \ldots$$

First-order Markov Processes

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

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

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

- The first-order Markov assumption:
  $P(X_i = x_i) | X_1 = x_1 \ldots X_{i-1} = x_{i-1})$
  $\stackrel{\text{assumption}}{=} P(X_i = x_i | X_{i-1} = x_{i-1})$

Second-order Markov Process

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

$$\overset{\text{assumption}}{=} P(X_1 = x_1) \times P(X_2 = x_2 | X_1 = x_1)$$

$$\times \prod_{i=3}^{n} P(X_i = x_i | X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1})$$

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

(For convenience we assume $x_0 = x_{-1} = *$, where $*$ is a special 'start' symbol)

Modeling variable length sequences

- We would like the length of the sequence, n to also be a random variable

- Simple solution: always define $X_n = STOP$ where STOP is a special symbol

- Then use a Markov process as before:

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

$$\stackrel{\text{assumption}}{=} \prod_{i=1}^{n} P(X_i = x_i | X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1})$$

(assume $x_0 = x_{-1} = *$)

Markov Models == n-gram Models

$(n-1)$ th-order Markov assumption $\equiv$ n-gram model

- Unigram model is $n = 1$ case
  $\rightarrow$ no conditioning on previous words

- For a long time, trigram models ($n = 3$) were widely used.
- 5-gram models (n=5) not uncommon in phrase-based MT

- The language modeling problem
- N-gram models

Acknowledgements

- Includes content from
  - Michael Collins
  - Noah Smith
  - Dan Klein
  - and many others indirectly …