Foundations of Natural Language Processing
Lecture 7c: The Noisy Channel Model

Alex Lascarides

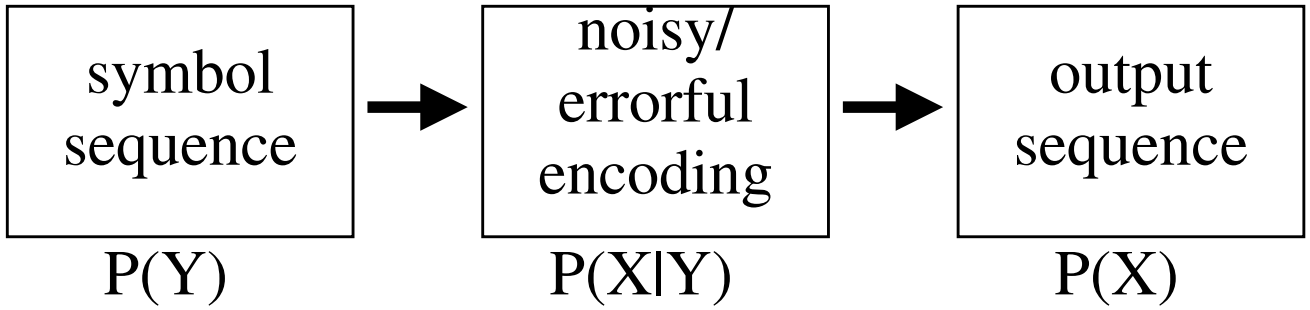
School of informatics

Back to the big picture

- However we train our LM, we will want to use it in some application.
- Now, a bit more detail about how that can work.
- We need another concept from information theory: the **Noisy Channel Model**.

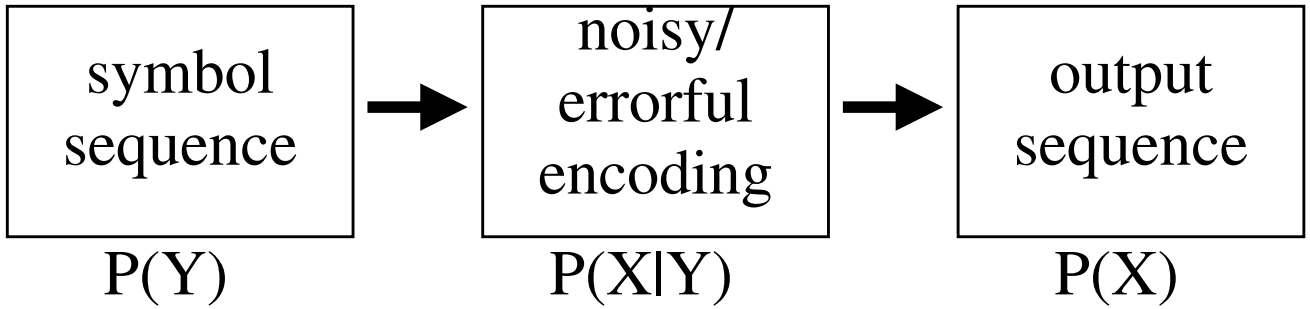
Noisy channel model

- We imagine that someone tries to communicate a sequence to us, but noise is introduced. We only see the output sequence.



Noisy channel model

- We imagine that someone tries to communicate a sequence to us, but noise is introduced. We only see the output sequence.



Application	Y	X
Speech recognition	spoken words	acoustic signal
Machine translation	words in L_1	words in L_2
Spelling correction	intended words	typed words

Example: spelling correction

- $P(Y)$: Distribution over the words the user intended to type. A language model!

- $P(X|Y)$: Distribution describing what user is **likely** to type, given what they **meant**. Could incorporate information about common spelling errors, key positions, etc. Call it a **noise model**.

- $P(X)$: Resulting distribution over what we actually see.

- Given some particular observation x (say, effert), we want to recover the most probable y that was intended.

Noisy channel as probabilistic inference

- Mathematically, what we want is $\text{argmax}_y P(y|x)$.
  - Read as "the y that maximizes $P(y|x)$"
- Rewrite using Bayes' Rule:

$$\begin{aligned} \underset{y}{\text{argmax}}\, P(y|x) &= \underset{y}{\text{argmax}}\, \frac{P(x|y)P(y)}{P(x)} \\ &= \underset{y}{\text{argmax}}\, P(x|y)P(y) \end{aligned}$$

Noisy channel as probabilistic inference

- So to recover the best y, we will need
  - a language model, which will be fairly similar for different applications
  - a noise model, which depends on the application: acoustic model, translation model, misspelling model, etc.

- Both are normally trained on corpus data.

You may be wondering

If we can train $P(X|Y)$, why can't we just train $P(Y|X)$? Who needs Bayes' Rule?

- Answer 1: sometimes we do train $P(Y|X)$ directly. Stay tuned...
- Answer 2: training $P(X|Y)$ or $P(Y|X)$ requires **input/output pairs**, which are often limited:
  - Misspelled words with their corrections; transcribed speech; translated text

  But LMs can be trained on huge unannotated corpora: a better model. Can help improve overall performance.

Model versus algorithm

- We defined a probabilistic model, which says **what** we should do.
  - E.g., for spelling correction: given trained LM and noise model (we haven't said yet how to acquire the noise model), find the intended word that is most probable given the observed word.
- We haven't considered **how** we would do that.
  - A **search problem**: there are (infinitely) many possible inputs that could have generated what we saw; which one is best?
  - We need to design an algorithm that can solve the problem.

Summary

- LMs are central to many NLP tasks:
  - Speech recognition, machine translation, spelling correction. . .
- They form a part of the Noisy Channel Model:
  - LM estimates likelihood of what speaker wanted to convey, given context.
  - Noise model estimates likely ways in which that (latent) message gets 'garbled':
    words to waves; French to English; effert to effort; . . .

Next Time: Algorithms for learning Noisy Channel Models from data.