ACS Statistical Machine Translation

Lecture 2: Introduction to SMT Models

(based on slides by Chris-Callison Burch and Philipp Koehn)

UNIVERSITY OF CAMBRIDGE

Stephen Clark

Natural Language and Information Processing (NLIP) Group

`sc609@cam.ac.uk`

Probabilities for SMT

- Find the most probable English sentence given a foreign language sentence (this is often how the problem is framed - of course can be generalised to any language pair in any direction)

$$\begin{aligned}\hat{e} &= \arg\max_e p(e|f) \\ &= \arg\max_e \frac{p(f|e)p(e)}{p(f)} \\ &= \arg\max_e p(f|e)p(e)\end{aligned}$$

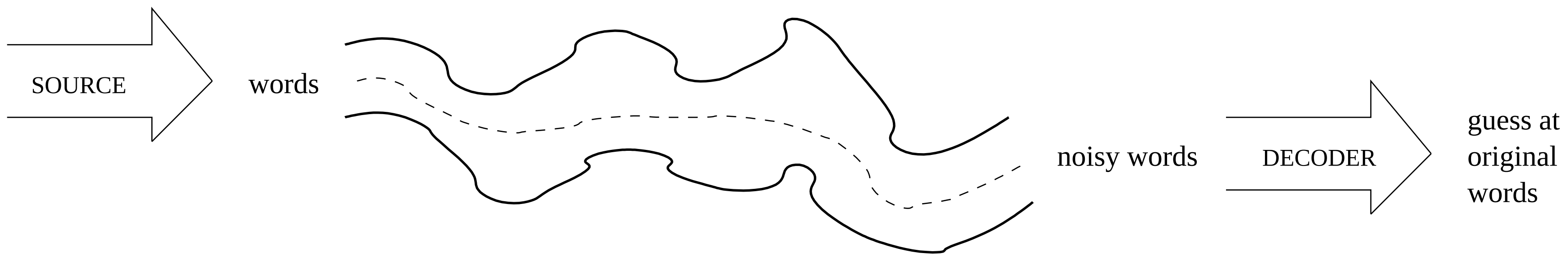
Individual Models

- $p(f|e)$ is the *translation model*
  (note the reverse ordering of f and e due to Bayes)
  - assigns a higher probability to English sentences that have the same meaning as the foreign sentence
  - needs a bilingual (parallel) corpus for estimation
- $p(e)$ is the *language model*
  - assigns a higher probability to fluent/grammatical sentences
  - only needs a monolingual corpus for estimation (which are plentiful)

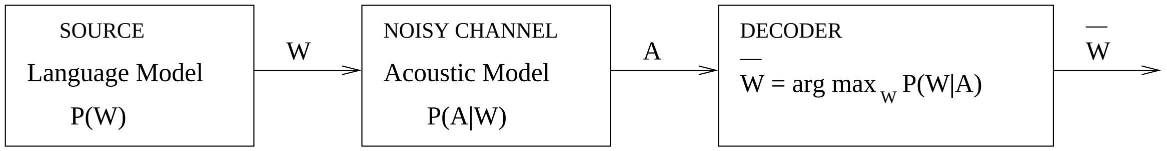
(picture of mt system: translation model, language model, search)

Model Interpretation as a Noisy Channel

- **Noisy channel** model has been applied to many language processing problems
- Based on the notion of a noisy channel from Shannon's information theory
- First applied to a language problem by the speech recognition group at IBM in the 70s



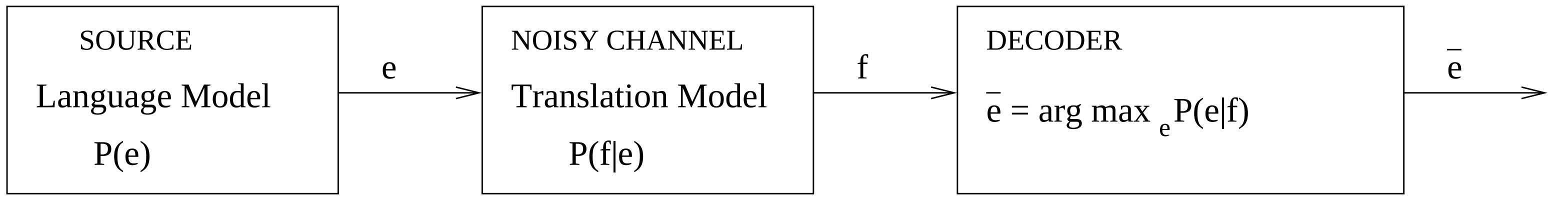
Speech Recognition as a Noisy Channel



- Speaker has word sequence W
- W is articulated as acoustic sequence A
- This process introduces noise:
  - variation in pronunciation
  - acoustic variation due to microphone etc.
- Bayes theorem gives us:

$$\begin{aligned}\overline{W} &= \arg\max_W P(W|A) \\ &= \arg\max_W \underbrace{P(A|W)}_{likelihood}\,\underbrace{P(W)}_{prior}\end{aligned}$$

Machine Translation as a Noisy Channel



- Translating French sentence (f) to English sentence (e)
- French speaker has English sentence in mind (P(e))
- English sentence comes out as French via the noisy channel (P(f|e))

n-gram Models

- In language modelling for ASR and MT, sequence information is important
  - e.g. $W =$ *The dogs were barking loudly*
  - **trigram** model captures some dependencies:

$$P(W) = P(\textit{The})P(\textit{dogs}|\textit{The})P(\textit{were}|\textit{The, dogs})P(\textit{barking}|\textit{dogs, were})P(\textit{loudly}|\textit{were, barking})$$

Calculating Language Model Probabilities

- Unigram probabilities

$$p(w_1) = \frac{f(w_1)}{N}$$

where $f(w_1)$ is the number of times w_1 is seen in some corpus and N is the total number of words seen in the corpus (by token)

- In this case the relative frequency estimation can be shown to be an instance of *maximum likelihood estimation*

Calculating Language Model Probabilities

- Bigram probabilities

$$p(w_2|w_1) = \frac{f(w_1, w_2)}{f(w_1)}$$

where $f(w_1, w_2)$ is the number of times w_2 is seen following w_1 in some corpus

Calculating Language Model Probabilities

- Trigram probabilities

$$p(w_3|w_1, w_2) = \frac{f(w_1, w_2, w_3)}{f(w_1, w_2)}$$

where $f(w_1, w_2, w_3)$ is the number of times w_3 is seen following w_2 and w_1 in some corpus

Sparse Data

- As we move to trigram counts (and perhaps beyond) **sparse data** becomes a problem
- Language is extremely productive, meaning that we're likely to encounter n-grams not seen in the training data
- Zero counts are particularly problematic, leading to zero relative frequency estimates (or undefined if the denominator is zero)
- Zero probabilities propogate through the product leading to a zero probability for the whole string

Smoothing

- Linear interpolation:

$$\tilde{p}(w_3|w_1,w_2) = \lambda_1 \hat{p}(w_3|w_1,w_2) + \lambda_2 \hat{p}(w_3|w_2) + \lambda_3 \hat{p}(w_3) + \epsilon$$

$\lambda_1 + \lambda_2 + \lambda_3 + \epsilon = 1.0$

Can we go beyond trigrams?

- yes - commercial systems (speech recognisers, Google SMT) will use 5 or 6-grams
- see “All Our N-gram are Belong to You”
  - Google have prepared a *1 trillion* word n-gram corpus and made it freely available
- But will still need smoothing, however much data we use (because language is so productive)

Examples of 4-grams from Google

```
serve as the incoming 92
serve as the incubator 99
serve as the independent 794
serve as the index 223
serve as the indication 72
serve as the indicator 120
serve as the indicators 45
serve as the indispensable 111
serve as the indispensible 40
serve as the individual 234
serve as the industrial 52
serve as the industry 607
serve as the info 42
serve as the informal 102
serve as the information 838
serve as the informational 41
serve as the infrastructure 500
```

serve as the occurs once in a 1-million word corpus

Translation Model

- $p(f|e)$ - the probability of some foreign language string given a hypothesis English translation
- $f =$ Ces gens ont grandi, vecu et oeuvre des dizaines d'annees dans le domaine agricole.
- $e =$ *Those people have grown up, lived and worked many years in a farming district.*
- $e =$ *I like bungee jumping off high bridges.*
- Allowing highly improbable translations (but assigning them small probabilities) was a radical change in how to think about the MT problem

Translation Model

- How do we estimate $p(f|e)$?
- $p(f|e) = \text{count}(f, e)/\text{count}(e)$
- We’ve seen enough language modelling now to know this isn’t going to work

Translation Model

- Introduce alignment variable a which represents alignments between the individual words in the sentence pair
- $p(f|e) = \Sigma_a \, p(a, f|e)$

(word alignment diagram)

Alignment Probabilities

- Now break the sentences up into manageable chunks (initially just the words)
- $p(a, f|e) = \Pi_{j=1}^{m} t(f_j|e_i)$

where e_i is the English word(s) corresponding to the French word f_j and $t(f_j|e_i)$ is the (conditional) probability of the words being aligned

Alignment Probabilities

- Relative frequency estimates can be used to estimate $t(f_j|e_i)$
- Problem is that we don't have *word*-aligned data, only sentence-aligned
- There is an elegant mathematical solution to this problem - the EM algorithm (more on this later)

Example Translations

See Koehn and Callison-Burch slides