

Assignment 2: Statistical Machine Translation

CSC401/2511 Tutorial 2 (Feb 27), Winter 2019
Raeid Saqr

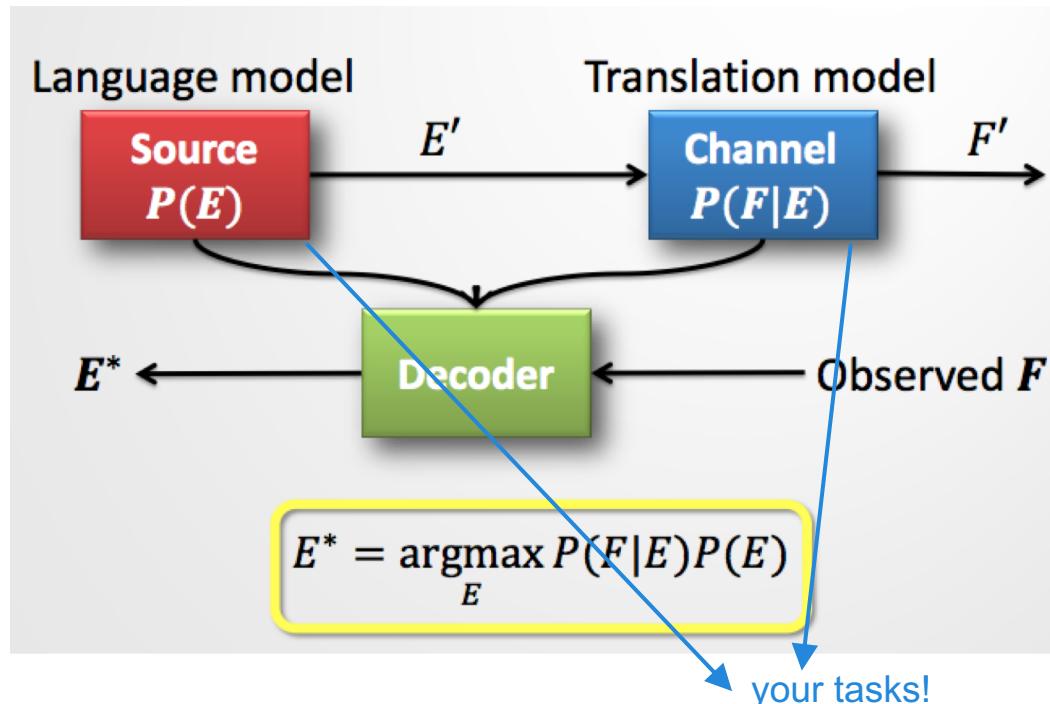
Based on slides by Mohamed Abdalla, Patricia Thaine, Jackie Cheung, Alex Fraser and Frank Rudzicz

Agenda

- Review and Q/A from previous tutorial
- Implementing IBM model 1
 - Initialization, EM algorithm, Pseudocode
- Marking: Task 4 auto-marking discussion
- Bonus Section Pointers
- Q/A

Review: Noisy Channel Model

We need a language model, a translation model, and a decoder.



A Translation Model

.Problem: calculating $P(F|E)$.

.Solution: Introducing word alignments.

.Possible Mappings between source words (English) and target words (French).

.Assumptions:

- .Translating from French F into English E.
- .Each French word comes from one English word.
- .No many-to-one or many-to-many mappings.

IBM Model 1

$$P(F|E) = \sum_A P(F, A|E)$$

Why?

$$P(F|E) = P(F, A_1|E)P(A_1) + \dots + P(F, A_n|E)P(A_n)$$

But since all alignments are equal in IBM1:

$$P(F|E) = P(F, A_1|E) + \dots + P(F, A_n|E)$$

Is equal to: $P(F|E) = \sum_A P(F, A|E)$

IBM Model 1

$$P(F|E) = \sum_A P(F, A|E)$$

Okay so how do we calculate $P(F, A|E)$?

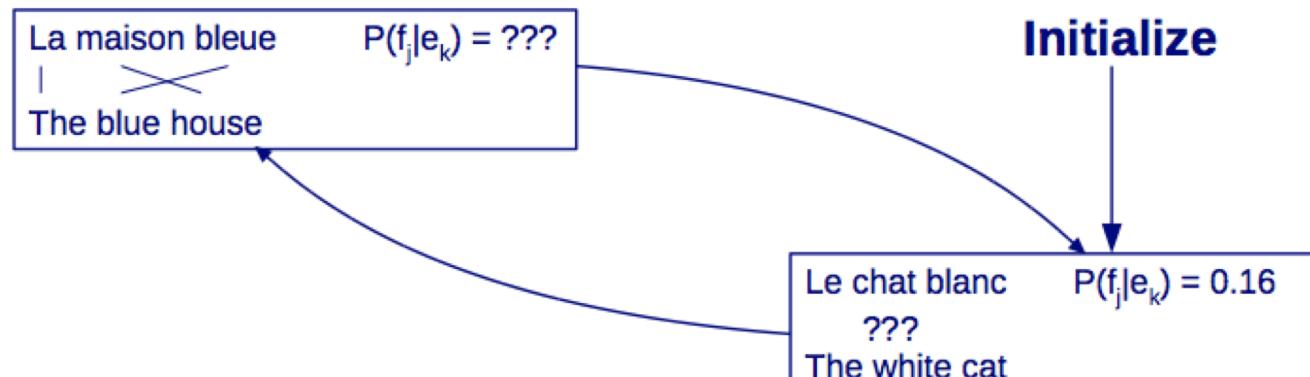
$$P(F, A|E) = \prod_j P(f_j | e_{a(j)})$$

Translated to English: The probability of a French sentence and an alignment is the probability of the first French word given aligned English word, AND (hence multiplication) the second French word given aligned English word, AND ...

EM Algorithm

.Parameters to learn: $P(f_j | e_{a(j)})$

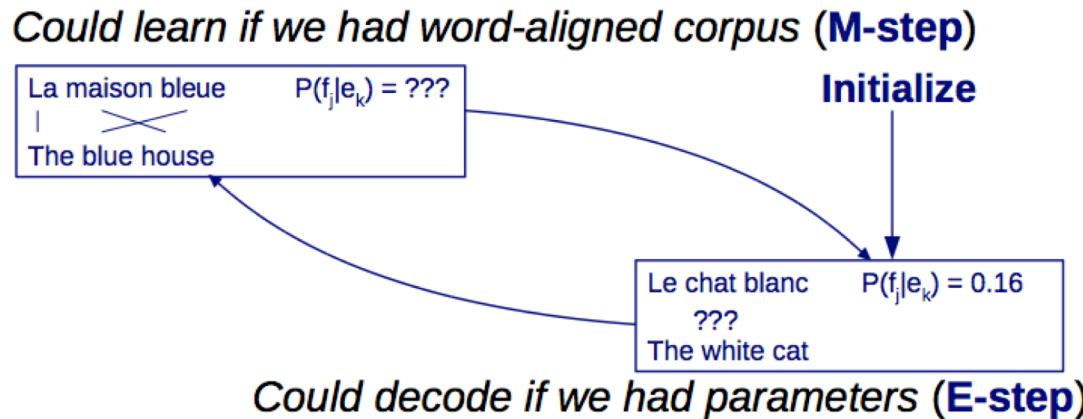
Could learn if we had word-aligned corpus (M-step)



Could decode if we had parameters (E-step)

EM Algorithm

- Initialize model parameters
- Iterate:
 - . Assign probabilities to missing data (alignments)
 - . Estimate model parameters from completed data



Initialization

- Need to start the cycle somewhere.
- Make up (reasonable) values for the parameters.
- Assume uniform distribution over all word pairs that occur together in some sentence.

Initialization Example

the blue cat

le chat bleu



$P(\text{le}|\text{the}),$
 $P(\text{chat}|\text{the}),$
 $P(\text{bleu}|\text{the}),$
 $P(\text{chien}|\text{the}),$
 $P(\text{rouge}|\text{the})$
 $= 1/5$



$P(\text{le}|\text{cat}),$
 $P(\text{chat}|\text{cat}),$
 $P(\text{bleu}|\text{cat})$
 $= 1/3$

the red dog

le chein rouge



$P(\text{le}|\text{red}),$
 $P(\text{rouge}|\text{red}),$
 $P(\text{chien}|\text{red})$
 $= 1/3$

But,
 $P(\text{rouge}|\text{cat}) = 0$

$P(\text{le}|\text{blue}),$
 $P(\text{chat}|\text{blue}),$
 $P(\text{bleu}|\text{blue})$
 $= 1/3$

$P(\text{le}|\text{dog}),$
 $P(\text{chien}|\text{dog}),$
 $P(\text{rouge}|\text{dog})$
 $= 1/3$

Expectation Step

Calculating $P(A|F, E)$ for all word alignments

$$P(A|E, F) = \frac{P(F, A|E)}{P(F|E)}$$

$$P(F|E) = \sum_A P(F, A|E)$$


IBM Model 1

Expectation Step

$$P(A|E,F) = \frac{\prod_j P(f_j | e_{a(j)})}{\sum_{A'} \prod_j P(f_j | e_{a'(j)})}$$
$$P(A|E,F) = \prod_j \frac{P(f_j | e_{a(j)})}{\sum_i P(f_j | e_i)}$$

Expectation Step

- Given ‘t’ parameters
- For each sentence pair:
 - For every possible alignment of this sentence pair, simply work out the equation of Model 1
 - Sum the Model 1 alignment scores, over all alignments of a sentence pair
 - Divide the alignment score of each alignment by this sum to obtain a normalized score
 - The resulting normalized score is the posterior probability of the alignment

Maximization Step

Collect counts to estimate new parameter values:

$$P(f|e) = \frac{tcount(f,e)}{total(e)}$$

→ Number of cases in training corpus where f is aligned to e, weighted by the probability of that alignment.

$$tcount(f,e) = \sum_{(E,F)} c(f|e;F,E)$$

aligned sentence-pairs (E,F) in training corpus

number of times f aligned to e in a

$$c(f|e;F,E) = \sum_a P(a|E,F) \text{ count}(f,e,a)$$

Maximization Step

$$c(f|e;F,E) = \sum_a P(a|E,F) \text{ count}(f,e,a)$$

parameters come from
previous iteration

$$c(f|e;F,E) = \frac{P(f|e)}{\sum_i P(f|e_i)} \text{count}(f,e)$$

ways to align f to e

Pseudocode

```
initialize P(f|e)
for a number of iterations:
    set tcount(f, e) to 0 for all f, e
    set total(e) to 0 for all e
    for each sentence pair (F, E) in training corpus:
        for each unique word f in F:
            denom_c = 0
            for each unique word e in E:
                denom_c += P(f|e) * F.count(f)
            for each unique word e in E:
                tcount(f, e) += P(f|e) * F.count(f) * E.count(e) / denom_c
                total(e) += P(f|e) * F.count(f) * E.count(e) / denom_c
        for each e in domain(total()):
            for each f in domain(tcount(:,e)):
                P(f|e) = tcount(f, e) / total(e)
```



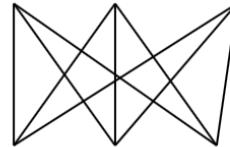
Model after iter i



Visual Example

Iteration 1

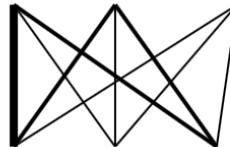
... la maison ... la maison blue ... la fleur ...



... the house ... the blue house ... the flower ...

Iteration 2

... la maison ... la maison blue ... la fleur ...



... the house ... the blue house ... the flower ...

Iteration 3

... la maison ... la maison bleu ... la fleur ...



... the house ... the blue house ... the flower ...

Decoding

$$E_{\text{best}} = \operatorname{argmax} P(F|E)P(E)$$

Finding the sentence that maximize the translation and language model probabilities:
A Search Problem

Decoder

-Greedy transformation:

- Find most likely word e for word f by $P(f|e)$
- Greedily reorder words to maximize $P(E)$
- Take the highest probability output at each step

-Stack decoding: (25.8 in Jurafsky & Martin)

- A* search
- Maintain a priority queue of partial translations
- Each partial translation has a score calculated based on translation and language model probabilities

Marking

(A large) Portion of it will be auto-marked.

- Your code will be tested individually (by file).
- Your code must adhere to the specifications for function calls to work.
- Do **NOT** hardcode any paths.
- It must work on CDF (test your code).

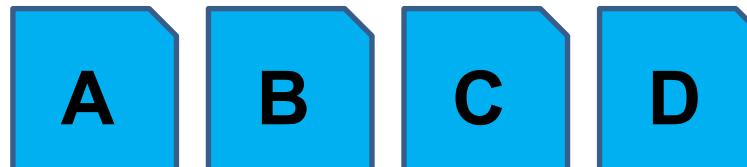
Marking (Cont)

Initially:

Correct Code:



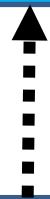
Your Code:



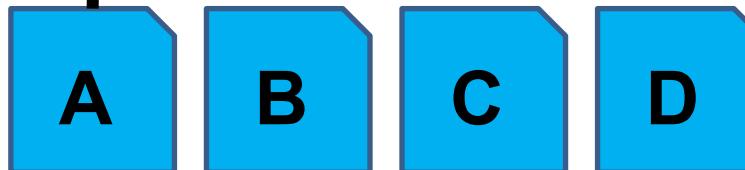
Marking (Cont)

If we are testing “A”:

Correct Code:



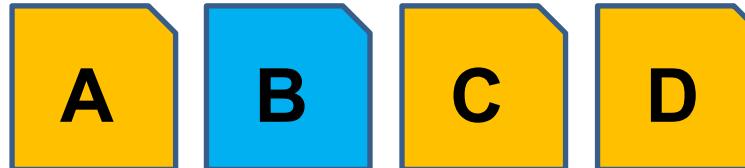
Your Code:



Marking (Cont)

If we are testing “B”:

Correct Code:



Your Code:

Bonus Marks

- Decoder
- Good-Turing smoothing
- IBM Model 2
- Null word
- Error analysis

Good-Turing Smoothing

- Assign the probability mass of n-grams with frequency of $r+1$ to the ones with frequency of r .
- Probability mass of n-grams that were never seen comes from the ones that occurred once

Good-Turing Smoothing

Fixed probability for unseen n-grams:

- . $P_0 = N_1 / N$
- . P_0 : total probability of all unseen events
- . N_1 : number of events that occurred once
- . N : number of all events

Probability of other events are adjusted to fit
inside probability space (M&S, Section 6.2.5)

IBM Model 2

$$P(F, a | E) = \prod_j P(f_j | e_{a(j)}) P(a(j) | j, \text{len}(F), \text{len}(E))$$

The probability that the i th English word is aligned to the j th French word, given lengths of the sentences.

Null Word

- Not all words may have an alignment:

Canada's program

Le programme du Canada

- Add NULL word

NULL Canada's program

Le programme du Canada

- Fixed probability of aligning with NULL.

Punctuation and Preprocessing

Q: *The preprocessing rules in the handout don't handle/mention this particular case. What do I do? Can I fix it?*

A: Yes, as long as it is reasonable and doesn't interfere with anything which is explicitly specified in the handout.

No bonus marks for extra work on this part.

You are not required to do anything beyond what is described in the handout.

Do not spend too much time on this part!

Accuracy

Q: How accurate should the final translations be?

A: The accuracy will vary wildly depending on the decoder and model you use. Marks for task 5 will not be based on the accuracy result you get, but rather the correctness of the evaluation code and your written report.

A large, bold, red 3D text "Questions" is positioned in the center of the image. The text has a slight shadow and a metallic texture. Surrounding the central text are numerous gray question marks of various sizes, some in sharp focus and others blurred, creating a sense of depth and repetition. The background is a solid blue color.

Questions