

CS 6501 Natural Language Processing

Statistical Machine Translation

Yangfeng Ji

October 1, 2018

Department of Computer Science
University of Virginia



ENGINEERING

Overview

1. Noisy Channel Model
2. IBM Model 1
3. IBM Model 2
4. Parameter Estimation

Based on slides from [Collins, 2017]

Noisy Channel Model

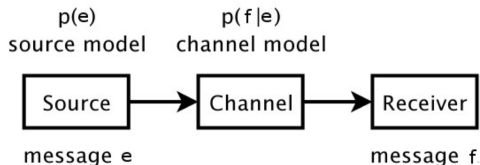
Problem

- ▶ Goal: translate French to English
- ▶ Mathematical formulation

$$P(\mathbf{e} \mid \mathbf{f}) \tag{1}$$

where $\mathbf{f} = (f_1, \dots, f_m)$ is a French sentence and $\mathbf{e} = (e_1, \dots, e_l)$ is an English translation.

Noisy Channel Model: Definition



► Bayes theorem

$$P(e | f) = \frac{P(e, f)}{P(f)} = \frac{P(e)P(f | e)}{\sum_e P(e)P(f | e)} \quad (2)$$

Two Components

$$P(e \mid f) = \frac{P(e, f)}{P(f)} = \frac{P(e)P(f \mid e)}{\sum_e P(e)P(f \mid e)} \quad (3)$$

- ▶ $P(e)$: the language model
- ▶ $P(f \mid e)$: the translation model

Two Components

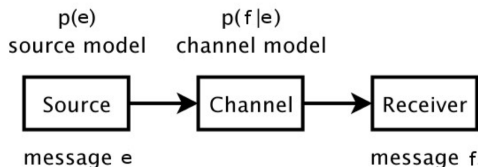
$$P(e \mid f) = \frac{P(e, f)}{P(f)} = \frac{P(e)P(f \mid e)}{\sum_e P(e)P(f \mid e)} \quad (3)$$

- ▶ $P(e)$: the language model
- ▶ $P(f \mid e)$: the translation model

Translation:

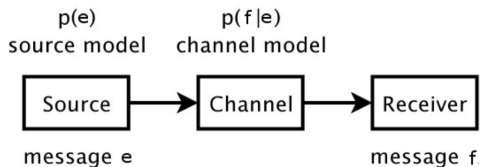
$$\hat{e} = \arg \max_e P(e \mid f) = \arg \max_e P(e)P(f \mid e) \quad (4)$$

Why Noisy Channel Model?



- ▶ Divide one big problem into two subproblems
- ▶ Solve them separately (with extra resources)

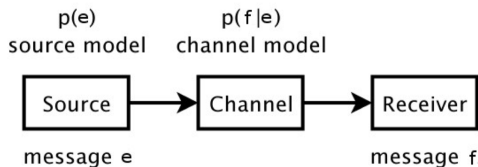
Question



Central question

How do we model $P(f | e)$?

Question



Central question

How do we model $P(f | e)$?

Examples:

- ▶ IBM Model 1
- ▶ IBM Model 2

What about $P(e)$?

IBM Model 1

Challenge of Probability Definition

Given

- ▶ an English sentence e with l words (e_1, \dots, e_l) and
- ▶ a French sentence f with m words (f_1, \dots, f_m) ,

directly modeling

$$P(f_1, \dots, f_m \mid e_1, \dots, e_l) \tag{5}$$

is a challenging task.

Example

e = And the program has been implemented

f = Le programme a ete mis en application


- ▶ $P(f \mid e)$ defines a probability on a 13-dimensional space
- ▶ There are alignments between French and English words

Example (Cont.)

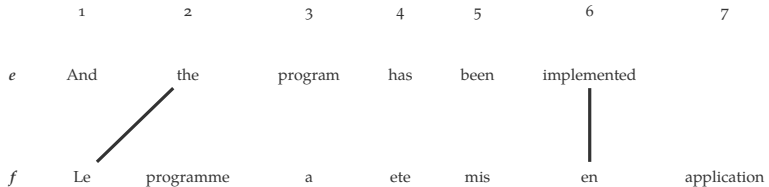
	1	2	3	4	5	6	7
<i>e</i>	And	the	program	has	been	implemented	
<i>f</i>	Le	programme	a	ete	mis	en	application

Example (Cont.)

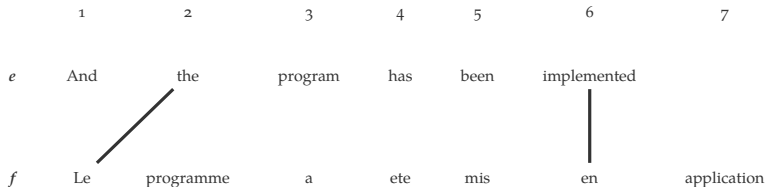
	1	2	3	4	5	6	7
<i>e</i>	And	the	program	has	been	implemented	
<i>f</i>	Le	programme	a	ete	mis	en	application



Example (Cont.)



Example (Cont.)



$$a_j = i$$

the j -th French word is aligned with the i -th word in English.

Examples

$$a_1 = 2, \quad a_6 = 6$$

Alignments

$$P(f_1, \dots, f_m, a_1, \dots, a_m \mid e_1, \dots, e_l) \quad (6)$$

where $a_j \in \{0, 1, \dots, l\}$

Alignments

$$P(f_1, \dots, f_m, a_1, \dots, a_m \mid e_1, \dots, e_l) \quad (6)$$

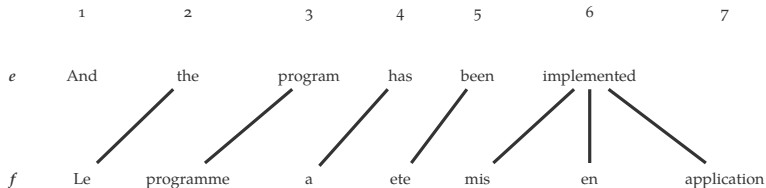
where $a_j \in \{0, 1, \dots, l\}$

Marginalizing over (a_1, \dots, a_m) gives the translation probability

$$P(f_1, \dots, f_m \mid e_1, \dots, e_l) = \sum_{a_1, \dots, a_m} P(f_1, \dots, f_m, a_1, \dots, a_m \mid e_1, \dots, e_l) \quad (7)$$

Example (Cont.)

An example of alignment



$$a_2 = 3$$

$$a_3 = 4$$

$$a_4 = 5$$

$$a_5 = 6$$

$$a_6 = 6$$

$$a_7 = 6$$

Further Factorization

Factorization

$$P(f, a \mid e) = P(a \mid e)P(f \mid a, e) \quad (8)$$

- ▶ Alignment $P(a \mid e)$
- ▶ Translation with a given alignment $a, P(f \mid a, e)$

IBM Model 1: $P(a \mid e)$

In IBM Model 1, all alignments are equally likely

$$P(a \mid e) = \frac{1}{(l+1)^m} \quad (9)$$

- ▶ Major simplification, great starting point
- ▶ Independent of words in f and e

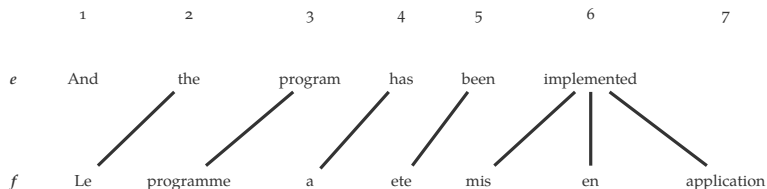
IBM Model 1: $P(f \mid a, e)$

Translation probabilities

$$P(f \mid a, e) = \prod_{j=1}^m t(f_j \mid e_{a_j}) \quad (10)$$

- ▶ f_j : the j -th French word
- ▶ $a_j = i$: the alignment of the j -th French word
- ▶ e_{a_j} : the aligned English word of the j -th French word
- ▶ $t(f_j \mid e_{a_j})$: the translation probability from the a_j -th English word to the j -th French word

Example (Cont.)



$$\begin{aligned} P(f \mid a, e) = & t(\text{Le} \mid \text{the}) \cdot t(\text{programme} \mid \text{program}) \cdot \\ & t(a \mid \text{has}) \cdot t(\text{ete} \mid \text{been}) \cdot \\ & t(\text{mis} \mid \text{implemented}) \cdot t(\text{en} \mid \text{implemented}) \cdot \\ & t(\text{application} \mid \text{implemented}) \end{aligned}$$

IBM Model 1: Final result

$$\begin{aligned}P(f \mid e) &= \sum_a P(f, a \mid e) \\&= \sum_a P(a \mid e) P(f \mid a, e) \\&= \sum_a \frac{1}{(l+1)^m} \prod_{j=1}^m t(f_j \mid e_{a_j})\end{aligned}\tag{11}$$

Break a big conditional probability into small pieces on word pairs

IBM Model 2

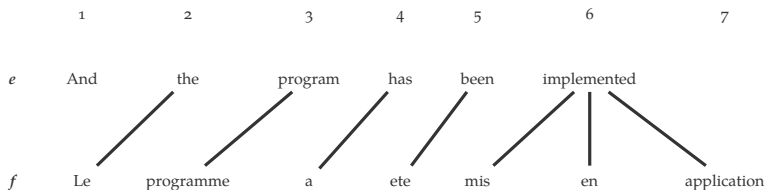
Alignments: Non-uniform distribution

Instead of uniform distribution, IBM 2 defines

$$P(a \mid e) = \prod_{j=1}^m q(a_j = i \mid j, l, m), \quad (12)$$

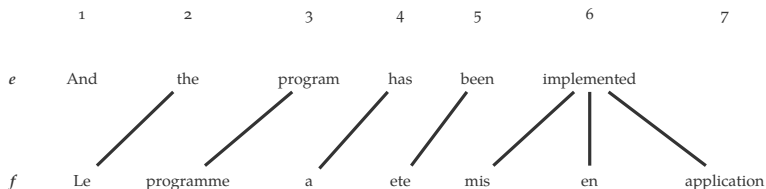
the probability that the j -th French word is connected to the i -th English word, given sentence lengths as l and m in English and French respectively

An Example



$$\begin{aligned} P(a \mid e) = & q(2 \mid 1, 6, 7) \cdot q(3 \mid 2, 6, 7) \cdot \\ & q(4 \mid 3, 6, 7) \cdot q(5 \mid 4, 6, 7) \cdot \\ & q(6 \mid 5, 6, 7) \cdot q(6 \mid 6, 6, 7) \cdot \\ & q(6 \mid 7, 6, 7) \end{aligned}$$

An Example (Cont.)



$$\begin{aligned} P(f \mid a, e) = & t(\text{Le} \mid \text{the}) \cdot t(\text{programme} \mid \text{program}) \cdot \\ & t(a \mid \text{has}) \cdot t(\text{ete} \mid \text{been}) \cdot \\ & t(\text{mis} \mid \text{implemented}) \cdot t(\text{en} \mid \text{implemented}) \cdot \\ & t(\text{application} \mid \text{implemented}) \end{aligned}$$

IBM Model 2: Final result

$$\begin{aligned}P(f \mid e) &= \sum_a P(f, a \mid e) \\&= \sum_a P(a \mid e) P(f \mid a, e) \\&= \sum_a \prod_{j=1}^m \{q(i \mid j, l, m) t(f_j \mid e_{a_j})\}\end{aligned}\tag{13}$$

Break a big conditional probability into small pieces on word pairs

Parameter Estimation

Problem Definition

- ▶ Input: $\{(e^{(k)}, f^{(k)}, a^{(k)})\}$
- ▶ Output:

$$t(f | e) = ?$$

$$q(i | j, l, m) = ?$$

- ▶ Method: Maximum-likelihood estimation, parameter estimation used in
 - ▶ HMMs with fully observed sentence-tag pairs
 - ▶ PCFGs with fully observed sentence-parse pairs

Estimation with Alignments Observed

- ▶ An example sentence pair with alignment

$e^{(100)}$ = And the program has been implemented

$f^{(100)}$ = Le programme a ete mis en application

$a^{(100)}$ = (2, 3, 4, 5, 6, 6, 6)

- ▶ Maximum-likelihood parameter estimates

$$t_{ML}(f_j | e_i) = \frac{c(e_i, f_j)}{c(e_i)}$$

$$q_{ML}(i | j, l, m) = \frac{c(i, j, l, m)}{c(j, l, m)}$$

Input: A training corpus $(f^{(k)}, e^{(k)}, a^{(k)})$ for $k = 1 \dots n$, where $f^{(k)} = f_1^{(k)} \dots f_{m_k}^{(k)}$, $e^{(k)} = e_1^{(k)} \dots e_{l_k}^{(k)}$, $a^{(k)} = a_1^{(k)} \dots a_{m_k}^{(k)}$.

Algorithm:

- Set all counts $c(\dots) = 0$
- For $k = 1 \dots n$
 - For $i = 1 \dots m_k$
 - * For $j = 0 \dots l_k$

$$c(e_j^{(k)}, f_i^{(k)}) \leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j)$$

$$c(e_j^{(k)}) \leftarrow c(e_j^{(k)}) + \delta(k, i, j)$$

$$c(j|i, l, m) \leftarrow c(j|i, l, m) + \delta(k, i, j)$$

$$c(i, l, m) \leftarrow c(i, l, m) + \delta(k, i, j)$$

where $\delta(k, i, j) = 1$ if $a_i^{(k)} = j$, 0 otherwise.

Output:

$$t_{ML}(f|e) = \frac{c(e, f)}{c(e)} \quad q_{ML}(j|i, l, m) = \frac{c(j|i, l, m)}{c(i, l, m)}$$

Comments: (1) i and j are exchanged; (2) the equation of q_{ML} is wrong.

Estimation without Alignments Observed

- ▶ Input: $\{(e^{(k)}, f^{(k)})\}$
- ▶ Output:

$$q(i \mid j, l, m) = ?$$

$$t(f_j \mid e_{a_j}) = ?$$

- ▶ Challenge:

We do not have alignments $\delta(k, i, j)$ on our training examples

Basic Idea

- ▶ Initialize $\{P(f_i | e_j)\}$ and $\{P(i | j, l, m)\}$
- ▶ Iterate between the following
 - ▶ $\forall k$, compute the expected alignment $\delta(k, i, j)$ where $\sum_j \delta(k, i, j) = 1$, and use it to update $c(\dots)$,

$$c(\dots) \leftarrow c(\dots) + \delta$$

- ▶ Update $\{P(f_i | e_j)\}$ and $\{P(i | j, l, m)\}$ with $c(\dots)$

Comments: i and j are exchanged.

- For $s = 1 \dots S$
 - Set all counts $c(\dots) = 0$
 - For $k = 1 \dots n$
 - * For $i = 1 \dots m_k$
 - For $j = 0 \dots l_k$

$$\begin{aligned}
 c(e_j^{(k)}, f_i^{(k)}) &\leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j) \\
 c(e_j^{(k)}) &\leftarrow c(e_j^{(k)}) + \delta(k, i, j) \\
 c(j|i, l, m) &\leftarrow c(j|i, l, m) + \delta(k, i, j) \\
 c(i, l, m) &\leftarrow c(i, l, m) + \delta(k, i, j)
 \end{aligned}$$

where

$$\delta(k, i, j) = \frac{q(j|i, l_k, m_k) t(f_i^{(k)} | e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i, l_k, m_k) t(f_i^{(k)} | e_j^{(k)})}$$

- Set

$$t(f|e) = \frac{c(e, f)}{c(e)} \quad q(j|i, l, m) = \frac{c(j|i, l, m)}{c(i, l, m)}$$

Comments: (1) i and j are exchanged; (2) the equation of q_{ML} is wrong; (3) the only difference is how to compute δ .

EM Algorithm

E-step $\forall k$, compute its **expected** alignment δ to $c(\cdots)$

$$c(\cdots) \leftarrow c(\cdots) + \delta$$

M-step **Maximize** the likelihood with the total expected count $c(\cdots)$

[Eisenstein, 2018, Sec. 5.1]

Summary

1. Noisy Channel Model
2. IBM Model 1
3. IBM Model 2
4. Parameter Estimation

Reference



Collins, M. (2017).
Natural language processing: Lecture notes.



Eisenstein, J. (2018).
Natural Language Processing.
MIT Press.