# A STATISTICAL APPROACH TO MACHINE TRANSLATION

Chunwei Yan

superjom@sz.pku.edu.cn

October 31, 2012

## Overview

# A Brief History

- One of the first application envisioned for computers

# A Brief History

- One of the first application envisioned for computers
- First idea of SMT:Warren Weaver(沃伦・韦弗)
  - 1949
  - Claude Shannon's(香侬) information theory

# A Brief History

- One of the first application envisioned for computers
- First idea of SMT:Warren Weaver(沃伦・韦弗)
  - 1949
  - Claude Shannon's(香侬) information theory
- Re-introduced in 1991 by researchers at IBM's Thomas J. Watson Research Center

# Interest in MT

- Translation is a universal need
- MT is popular on the web
  - Google Translator
  - Bing Translator
  - 百度在线翻译
- (Semi-)automated translation could lead to huge savings

# Introduction to Translation

## Definition

Job of a translator: render in one language the meaning expressed by a passage of text in another language

# Introduction to Translation

**Assume**

Every sentence in one language is a possible translation of any sentence in the other.

| | |
|---|---|
| C1 | E1 |
| C2 | E2 |
| C3 | E3 |

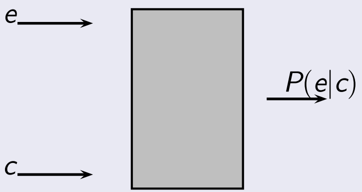
# Introduction to Translation

## Assume

Every sentence in one language is a possible translation of any sentence in the other.

| | |
|---|---|
| C1 | E1 |
| C2 | E2 |
| C3 | E3 |

## Program 1



# Introduction

**Assume**

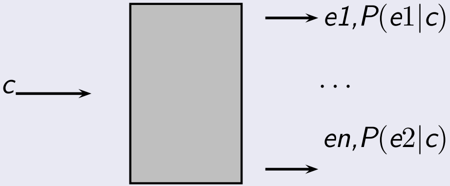
Given a Chinese sentence c, seek the English sentence e that maximizes $P(e|c)$

# Introduction

## Assume

Given a Chinese sentence c, seek the English sentence e that maximizes $P(e|c)$

## Program 2

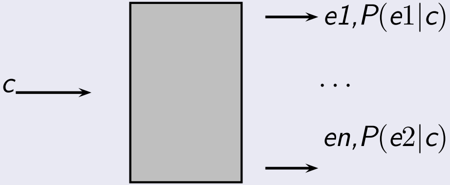


# Introduction

## Assume

Given a Chinese sentence c, seek the English sentence e that maximizes $P(e|c)$

## Program 2




$$\arg\max_e P(e|c)$$

Using Bayes Rule(贝叶斯法则):

$$P(e|c) = \frac{P(ec)}{P(c)} = \frac{P(c|e)P(e)}{P(c)}$$

$$\text{So: } P(e|c) \sim P(c|e)P(e)$$

Using Bayes Rule(贝叶斯法则):

$$P(e|c) = \frac{P(ec)}{P(c)} = \frac{P(c|e)P(e)}{P(c)}$$

$$\text{So: } P(e|c) \sim P(c|e)P(e)$$

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

# Noisy Channel Model

$$\arg\max_e P(e|f)\,P(e)$$

# Noisy Channel Model

$$\arg\max_e P(e|f) P(e)$$

## Bayesian Reasioning

Observe f and try to come up with the most likely translation e,

every e gets the score $P(e) * P(f|e)$

- Medical symptom

# Noisy Channel Model

$$\arg\max_e P(e|f) P(e)$$

**Bayesian Reasioning**

Observe f and try to come up with the most likely translation e,

every e gets the score $P(e) * P(f|e)$

- Medical symptom
  - The probability that symptoms f will arise from disease e **P(f|e)**

# Noisy Channel Model

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

## Bayesian Reasioning

Observe f and try to come up with the most likely translation e, every e gets the score $P(e) * P(f|e)$

- Medical symptom
  - The probability that symptoms f will arise from disease e **P(f|e)**
  - Is disease e a common disease? **P(e)**

# Noisy Model

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

# Noisy Model

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

1. Imagine that someone has e in his head **P(e)**

# Noisy Model

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

1. Imagine that someone has e in his head **P(e)**
2. By the time it gets on to the printed page it is corrupted by "noise" and becomes f **P(f|e)**

# SMT Model

## $P(e)$ Language Model

Is the disease symptom common?

Or is the English sentence common?

- good enough
- grammatically right

# SMT Model

## $P(e)$ Language Model

Is the disease symptom common?

Or is the English sentence common?

- good enough
- grammatically right

$P(e)$ worries about English word order and asign a high score to e if it is grammatical

# SMT Model

## $P(e)$ Language Model

Is the disease symptom common?

Or is the English sentence common?

- good enough
- grammatically right

$P(e)$ worries about English word order and asign a high score to e if it is grammatical

## $P(e|f)$ Translation Model

Will symptoms f arise from disease e?

Or can e sentence be transformed to f?

# SMT Model

## $P(e)$ Language Model

Is the disease symptom common?

Or is the English sentence common?

- good enough
- grammatically right

$P(e)$ worries about English word order and asign a high score to e if it is grammatical

## $P(e|f)$ Translation Model

Will symptoms f arise from disease e?

Or can e sentence be transformed to f?

$P(f|e)$ will ensure that a good e will have words that generally translate to words in f.

# SMT Model Demo

## $P(e)$ Language Model

1. a bites dog him
2. bites hime a dog
3. a dog bites him

# SMT Model Demo

## $P(e)$ Language Model

1. a bites dog him
2. bites hime a dog
3. a dog bites him

## $P(e|f)$ Translation Model

**Word bag**

Turn a bag of French words into a bag of English words

Translate " 一只狗咬了他" to

1. a bites dog him
2. bites hime a dog
3. a dog bites him

# Language Model

Language Model worries about English word order

# Language Model

Language Model worries about English word order

## N-grams

| | | |
|---|---|---|
| n=1 | unigram | $P(x)$ |
| n=2 | bigram | $P(y\|x)$ |
| n=3 | trigram | $P(yz\|x)$ |

# Language Model

Given a word string, $s_1 s_2 \cdots s_n$, we can write

# Language Model

Given a word string, $s_1 s_2 \cdots s_n$, we can write

$$P(s_1 s_2 \cdots s_n) = P(s_1)P(s_2|s_1)\cdots P(s_n|s_1 s_2 \cdots s_{n-1})$$

# Language Model

Given a word string, $s_1 s_2 \cdots s_n$, we can write

$$P(s_1 s_2 \cdots s_n) = P(s_1)P(s_2|s_1) \cdots P(s_n|s_1 s_2 \cdots s_{n-1})$$

Using Bigram Model:

# Language Model

Given a word string, $s_1 s_2 \cdots s_n$, we can write

$$P(s_1 s_2 \cdots s_n) = P(s_1) P(s_2|s_1) \cdots P(s_n|s_1 s_2 \cdots s_{n-1})$$

Using Bigram Model:

$$P(s_1 s_2 \cdots s_n) \sim P(s_1|NULL) P(s_2|s_1) \cdots P(s_n|s_{n-1})$$

# Language Model

Given a word string, $s_1 s_2 \cdots s_n$, we can write

$$P(s_1 s_2 \cdots s_n) = P(s_1)P(s_2|s_1) \cdots P(s_n|s_1 s_2 \cdots s_{n-1})$$

Using Bigram Model:

$$P(s_1 s_2 \cdots s_n) \sim P(s_1|NULL)P(s_2|s_1) \cdots P(s_n|s_{n-1})$$

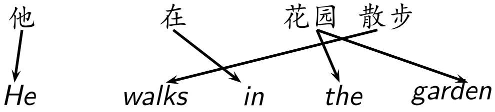
## Bigram Model Demo

$$b(y|x) = \frac{number - of - occurrences(xy)}{number - of - occurrences(x)}$$

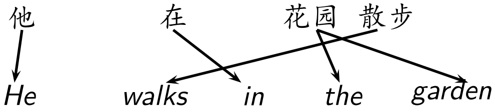
P(a dog bites him) $\sim$

b(a|NULL) b(dog|a) b(bites|dog) b(him|bites)

# Translation Model



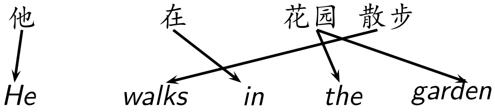
# Translation Model




**Fertility Parameters(生殖力系数)**

The number of the English words that a Chinese word produces in a given alignment n(1| 他) n(2| 花园)

# Translation Model




## Fertility Parameters(生殖力系数)

The number of the English words that a Chinese word produces in a given alignment n(1| 他) n(2| 花园)

## Distortion Parameters(畸变系数)

d(3|2)=1 : frequency of 在 ->in

d(3|2,4,5)=1: add length of the Chinese and English sentences

# Translation Model

## Translation Parameters

The frequency of a Chinese word that connects to a certain English word.

t(walk| 散步)=1

t(garden| 花园)=1

# Translation Model

## Translation Parameters

The frequency of a Chinese word that connects to a certain English word.

t(walk| 散步)=1

t(garden| 花园)=1

$$P(a, f|e) = \prod_{i=1}^{l} n(ph_i|e_i) * \prod_{j=1}^{m} t(f_j|ea_j) * \prod_{j=1}^{m} d(j|a_j, l, m)$$

# Translation Model

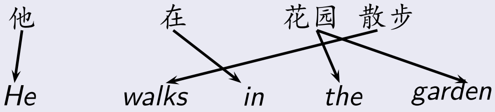
If we had a bunch of English strings and a bunch of step-by-step rewritings into French, then life would be easy.

# Translation Model

If we had a bunch of English strings and a bunch of step-by-step rewritings into French, then life would be easy.

## Translation Sequence

他在花园散步 => He walks in the garden



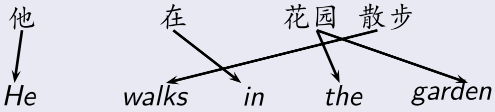

- input> 他在花园散步

# Translation Model

If we had a bunch of English strings and a bunch of step-by-step rewritings into French, then life would be easy.

## Translation Sequence

他在花园散步 => He walks in the garden



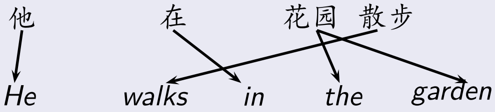

- input> 他在花园散步
- 他在花园花园散步< Fertility Parameter

# Translation Model

If we had a bunch of English strings and a bunch of step-by-step rewritings into French, then life would be easy.

## Translation Sequence

他在花园散步 => He walks in the garden



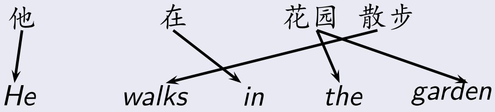

- input> 他在花园散步
- 他在花园花园散步< Fertility Parameter
- He in the garden walks < Translation Model

# Translation Model

If we had a bunch of English strings and a bunch of step-by-step rewritings into French, then life would be easy.

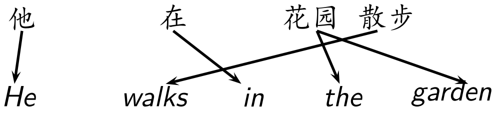
## Translation Sequence

他在花园散步 => He walks in the garden

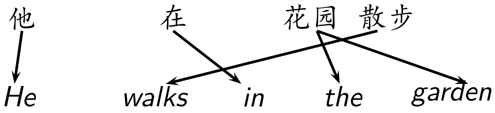



- input> 他在花园散步
- 他在花园花园散步< Fertility Parameter
- He in the garden walks < Translation Model
- He walks in the garden< Language Model

# Word-for-Word Alignments



# Word-for-Word Alignments




- n(2| 在):See how many times " 在" connected to two English words
- t(garden| 花园):Count up all the English words generated by all the occurrences of " 花园", and see how many of those words are "garden"

# Estimationg Parameter Values

$$d(p_e|p_c, l_c, l_e) = \frac{dc(p_e|p_c, l_c, l_e)}{\sum_{j=1}^{N} (j|p_c, l_c, l_e)}$$

# Estimationg Parameter Values

$$d(p_e|p_c, l_c, l_e) = \frac{dc(p_e|p_c, l_c, l_e)}{\sum_{j=1}^{N} (j|p_c, l_c, l_e)}$$

$$n(i|c) = \frac{Count(c - connect - i - words)}{Count(c - connections)}$$

# Estimationg Parameter Values

$$d(p_e|p_c, l_c, l_e) = \frac{dc(p_e|p_c, l_c, l_e)}{\sum_{j=1}^{N} (j|p_c, l_c, l_e)}$$

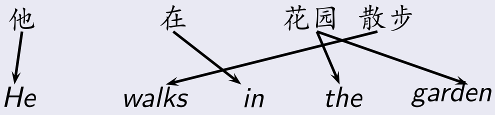
$$n(i|c) = \frac{Count(c - connect - i - words)}{Count(c - connections)}$$

$$t(e|c) = \frac{c - connect - e}{Count(c - connections)}$$

# EM (Estimation-Maximiztion) algorithm

## Compute parameter estimates

Given the alignment, we compute parameters




n(2| 花园) t(he| 他) $\cdots$

# EM (Estimation-Maximiztion) algorithm

## Compute Alignment Probabilities

$$P(a, f|e) = \prod_{i=1}^{l} n(ph_i|e_i) * \prod_{j=1}^{m} t(f_j|e_j) * \prod_{j=1}^{m} d(j|a_j, l, m)$$

$$P(a|e, f) = \frac{P(a, f|e)}{\sum_a P(a, f|e)}$$

# EM (Estimation-Maximiztion) algorithm

## Compute Alignment Probabilities

$$P(a, f|e) = \prod_{i=1}^{l} n(ph_i|e_i) * \prod_{j=1}^{m} t(f_j|e_j) * \prod_{j=1}^{m} d(j|a_j, l, m)$$

$$P(a|e, f) = \frac{P(a, f|e)}{\sum_a P(a, f|e)}$$

## Chicken and Egg

Uh oh, what went wrong?

# EM (Estimation-Maximiztion) algorithm

## Compute Alignment Probabilities

$$P(a, f|e) = \prod_{i=1}^{l} n(ph_i|e_i) * \prod_{j=1}^{m} t(f_j|e_j) * \prod_{j=1}^{m} d(j|a_j, l, m)$$

$$P(a|e, f) = \frac{P(a, f|e)}{\sum_a P(a, f|e)}$$

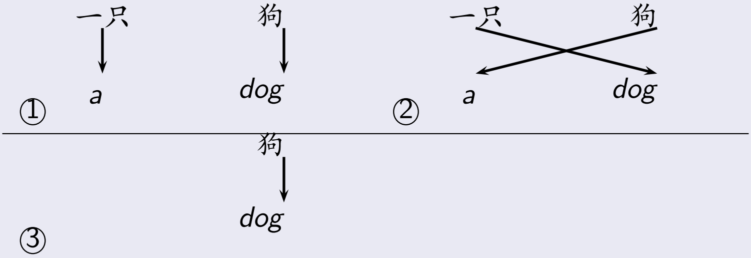
## Chicken and Egg

The EM algorithm can solve the problem.

# SMT Word-Alignment Demo

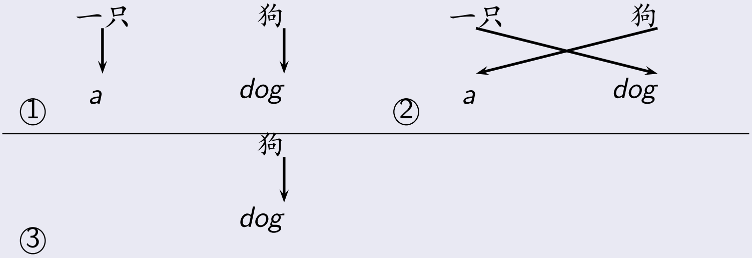
(一只狗,a dog), (狗,dog)

## Possible Alignments



# EM Process

## Possible Alignments

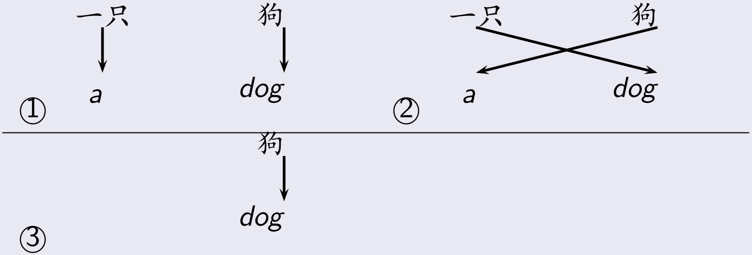



## Step 1. Set parameter values uniformly

- t(a| 一只)=1/2
- t(dog| 一只)=1/2
- t(a| 狗)=1/2
- t(dog| 狗)=1/2

# EM Process

## Possible Alignments

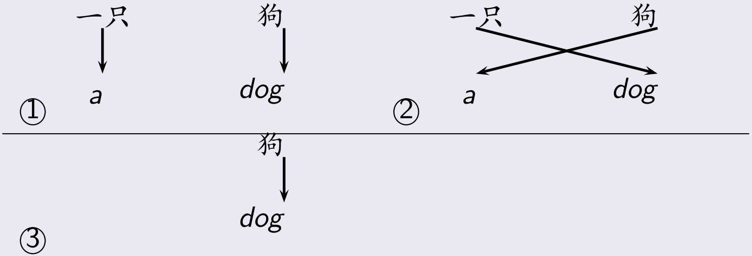



## Step 2. Compute $P(a, f|e)$ for all alignments

- ① $P(a, f|e) = 1/2 * 1/2 = 1/4$
- ② $P(a, f|e) = 1/2 * 1/2 = 1/4$
- ③ $P(a, f|e) = 1/2$

# EM Process

## Possible Alignments




## Step 3. Normalize $P(a, f|e)$ values to yield $P(a|e, f)$ values

- ① $P(a|e, f) = \frac{1/4}{2/4} = 1/2$
- ② $P(a|e, f) = \frac{1/4}{2/4} = 1/2$
- ③ $P(a|e, f) = \frac{1/2}{1/2} = 1$

# EM Process

**Step 4. Collect fractional counts**

- tc(a| 狗)=1/2
- tc(dog| 狗)=1 + 1/2 = 3/2
- tc(a| 一只)=1/2
- tc(dog| 一只)=1/2

# EM Process

**Step 5. Normalize fractional counts to get revised parameter values.**

- t(a| 狗)$=\frac{1/2}{4/2}=1/4$
- t(dog| 狗)$=\frac{3/2}{4/2}=3/4$
- t(a| 一只)$=\frac{1/2}{1}=1/2$
- t(dog| 一只)$=\frac{1/2}{1}=1/2$

# EM Process

## Repeating steps 2-5 many times

- t(a| 狗)=0.0001
- t(dog| 狗)=0.9999
- t(a| 一只)=0.9999
- t(dog| 一只)=0.0001

# EM Process

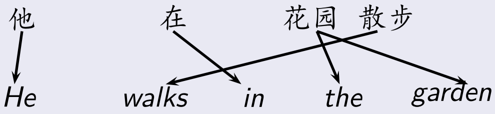
## Final Word Alignment

- 一只 ->a
- 狗 ->dog

# Summarize

## Translation Sequence

他在花园散步 => He walks in the garden

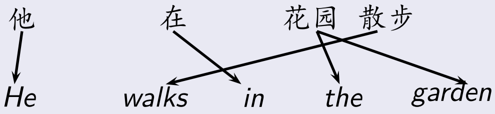



- input> 他在花园散步

# Summarize

## Translation Sequence

他在花园散步 => He walks in the garden

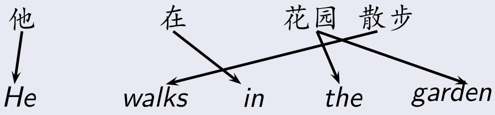



- input> 他在花园散步
- 他在花园花园散步< Fertility Parameter

# Summarize

## Translation Sequence

他在花园散步 => He walks in the garden

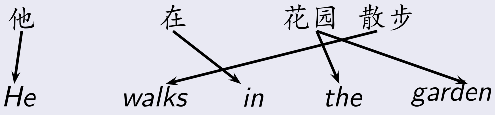



- input> 他在花园散步
- 他在花园花园散步< Fertility Parameter
- He in the garden walks < Translation Model

# Summarize

## Translation Sequence

他在花园散步 => He walks in the garden




- input> 他在花园散步
- 他在花园花园散步< Fertility Parameter
- He in the garden walks < Translation Model
- He walks in the garden< Language Model

# References

[Peter F.Brown, John Cocke, Stephen A. Della Pietra, · · · ]
A STATISTICAL APPROACH TO MACHINE TRANSLATION,June 1990

[Kevin Knight]
A Statistical MT Tutorial Workbook, April 30, 1999

# Thank You!

Thank you for your listenning!