

第 4 章

詞彙為本模型

教科書網站：www.statmt.org/book/

參考課程網站：mt-class.org/jhu/syllabus.html

Sept. 25, 2018

詞對詞的翻譯 Lexical Translation

- 如何翻譯單字呢？ → 通常我們查人工編輯的雙語辭典 (如 dictionary.cambridge.org/us/dictionary/english-chinese-traditional/house)

guide word

house —

HOME

房屋，住宅；全家人；籠舍

PUBLIC BUILDING

大樓，大廈；

BUSINESS

公司，機構，商行

MUSIC

電子音樂

SCHOOL GROUP

組

FAMILY

家族，皇室

POLITICS

議會，議院，議員；辯論的發起方

PEOPLE AT THEATRE

觀眾，劇院觀眾

HOME *She lives in a little house in (us on) Cross Street.*

她住在十字街上的一間小房子裡。

*Try not to wake the whole **house** when you come in!*

你進來的時候不要把全家人吵醒！

BUILDING *the Sydney Opera **House*** 雪梨歌劇院

*Broadcasting **House*** 廣播電台大樓

BUSINESS *a publishing **house*** 出版社 *a fashion **house*** 時裝屋

MUSIC ***House** music first appeared in the late 1980s.*

電子音樂最早出現於20世紀80年代晚期。

SCHOOL GROUP *an inter-**house** hockey match* 校內小組間足球賽

FAMILY *The British Royal Family belong to the **House** of Windsor.*

英國王室屬於溫莎家族。

POLITICS *The **House** began debating the proposal at 3 p.m.*

議員們于下午三點開始就提案進行辯論。

PEOPLE AT THEATRE *The opera played to a full/packed **house**.*

該歌劇演出時觀眾爆滿。

- 多義詞或同音異議詞造成「一詞多譯」 multiple translations
 - 有些翻譯比較常見，例如房屋、衆議院
 - 有些翻譯非常獨特少見：例如 電子音樂、皇室
 - 有些是多義詞衍生的不同翻譯，有些是同音異議所衍生的不同翻譯
 - HOME, BUILDING, BUSINESS, PEOPLE 屬於「多義」
 - 要分那麼細嗎？是否可能有系統、一致地描述多義詞（空間轉喻人）
- 反之，「一義多詞」會不會影響到機器翻譯（機器學習如何翻譯）呢？
 - 定義：房屋，住宅
 - 例句：a little house in Cross Street 十字街上的一間小 房子
 - 同義詞：房屋、住宅、房子
- 附註：以後舉例常由「外文：德、華語」（來源）翻譯到英語（目標）
- 但是，描述詞與詞的對齊函數時，我們反向為之，由英語到外語（德語）

SMT 蒐集樣本、統計翻譯資訊（不用人工辭典）

- 分析平行語料庫（德語文本＋英語翻譯） 句子對句子 → 使用統計方法找出對應的翻譯
- 自動找到詞對詞的對應
- 計算互譯次數（不只是一起出現，而是隱藏，不易知道的互譯關係）

<i>Haus</i> 的翻譯	Count
house	8,000
building	1,600
home	200
household	150
shell	50

- 如果不確知互譯關係，可以猜測互譯次數嗎？（EM 演算法的 E 步驟）

估算翻譯機率 Estimate Translation Probabilities

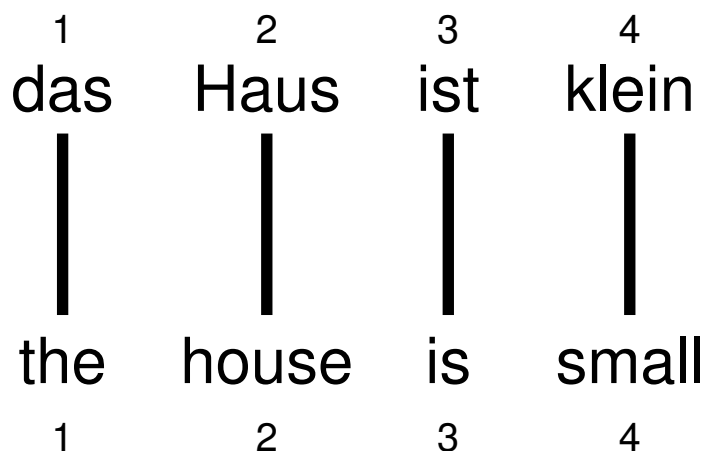
- 如果有次數，就可用最大似然估算 Maximum likelihood estimation (MLE)
- 用樣本內的實際（或推測）次數，來估算機率
- (必須處理 0 次數問題)

$$p_f(e) = \begin{cases} 0.8 & \text{if } e = \text{house,} \\ 0.16 & \text{if } e = \text{building,} \\ 0.02 & \text{if } e = \text{home,} \\ 0.015 & \text{if } e = \text{household,} \\ 0.005 & \text{if } e = \text{shell.} \end{cases}$$

詞彙對應 Word Alignment

- 在平行語料庫中，把目標語（英語）詞彙「對齊到」原始語（德語）詞彙（有時還需要反向）用 1-4 來表示詞的位置

先對齊



- 在互動式翻譯系統（如 TransType）中 [Langlais, Philippe, George Foster, and Guy Lapalme, 2000]
 - 一面翻譯一面做對齊，分析半完成翻譯和原句間的對應關係
 - 提示接下來（未對到部分）的翻譯

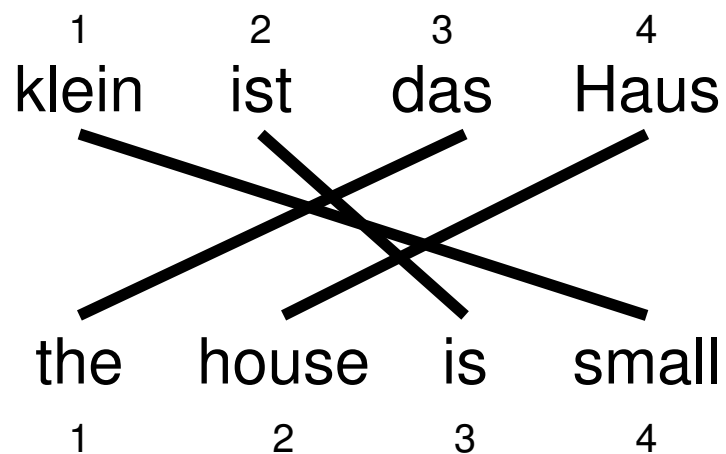
對齊函數 Alignment Function

- 用數學、形式化的方式描述目標語到來源語的對應：對齊函數
- 把（目標語）的英文詞位置，用函數對應到相關（來源語）德文詞位置
- 位置 i 英文詞 $e_i \rightarrow f_j$ 位置 j 的德文詞，函數 $a : i \rightarrow j$
- 例如

$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4\}$$

詞序重組 Reordering

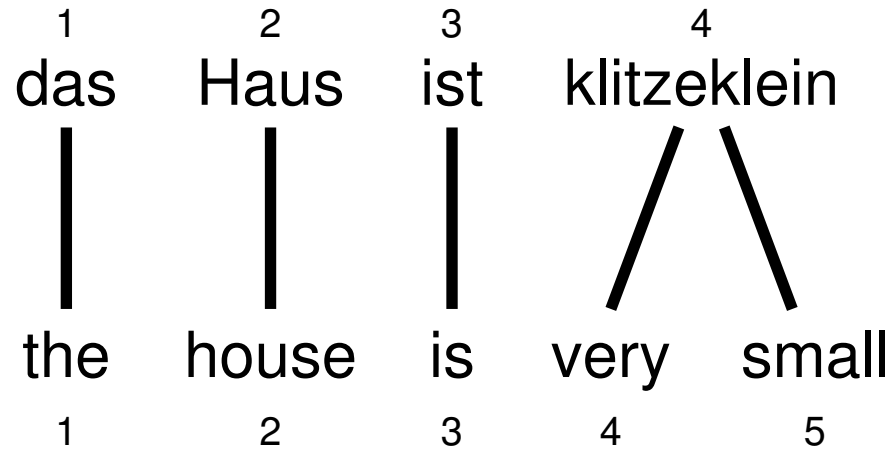
翻譯過程詞的順序可能重組



$$a : \{1 \rightarrow 3, 2 \rightarrow 4, 3 \rightarrow 2, 4 \rightarrow 1\}$$

一到多的翻譯 = 多到一的對齊

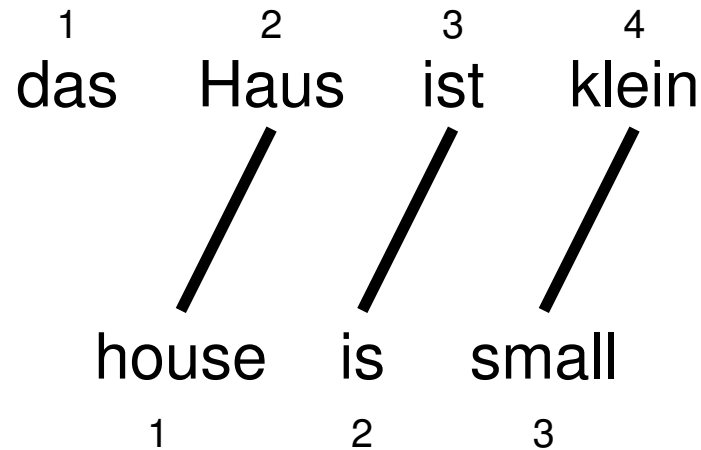
(多對一的) 函數可以表達 (來源語到目標語) 的「一到多對應」



$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4, 5 \rightarrow 4\}$$

刪除來源語的詞彙 Dropping Words

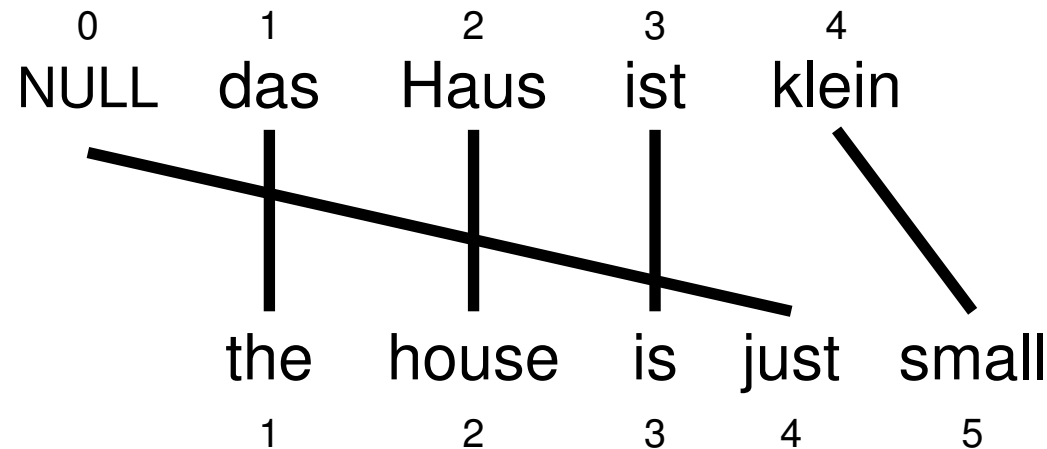
- 對應函數 a 可以表達「翻譯過程中，有些詞可以不翻譯」
- 例如，德文冠詞 das 丟掉 dropped 不翻出



$$a : \{1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 4\}$$

插入目標語詞彙 Inserting Words

- 翻譯過程中，可能會插入詞彙
 - 在例子中，英語詞 **just** 並沒有華語對應
 - 為此，我們假設插入詞對應到外文的第 0 詞 $f_0 = \text{NULL}$



$$a : \{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 0, 5 \rightarrow 4\}$$

IBM 模型 1

- 用生成性模型 generative model 決定機率最佳的詞對詞小步驟
 - IBM 模型 1 只考慮詞彙（不考慮片語、詞序重組）
 - 給予長度 l_f 的外文句 $\mathbf{f} = (f_1, \dots, f_{l_f})$ 和長度 l_e 的英語翻譯 $\mathbf{e} = (e_1, \dots, e_{l_e})$
 - 對應函數 $a : j \rightarrow i$ 把英語詞 e_j 對應到外語詞 $f_{a(j)}$
 - 定義整句的「翻譯機率」translation probability

$$p(\mathbf{e}, a | \mathbf{f}) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | \underline{f_{a(j)}}_{fi})$$

- 其中的 $(l_f + 1)^{l_e}$ 是排列組合 (l_e 個英文每個對 $(l_f + 1)$ 個外文或 NULL)
- 其中的 ϵ 為正規化參數（讓機率加總得到 1.0）

例子：整句的「翻譯機率」

das		Haus		ist		klein	
e	$t(e f)$	e	$t(e f)$	e	$t(e f)$	e	$t(e f)$
the	0.7	house	0.8	is	0.8	small	0.4
that	0.15	building	0.16	's	0.16	little	0.4
which	0.075	home	0.02	exists	0.02	short	0.1
who	0.05	household	0.015	has	0.015	minor	0.06
this	0.025	shell	0.005	are	0.005	petty	0.04

$$\begin{aligned} p(e, a|f) &= \frac{\epsilon}{5^4} \times t(\text{the}|\text{das}) \times t(\text{house}|\text{Haus}) \times t(\text{is}|\text{ist}) \times t(\text{small}|\text{klein}) \\ &= \frac{\epsilon}{5^4} \times 0.7 \times 0.8 \times 0.8 \times 0.4 \\ &= 0.0028\epsilon \end{aligned}$$

如何習得詞彙機率模型？

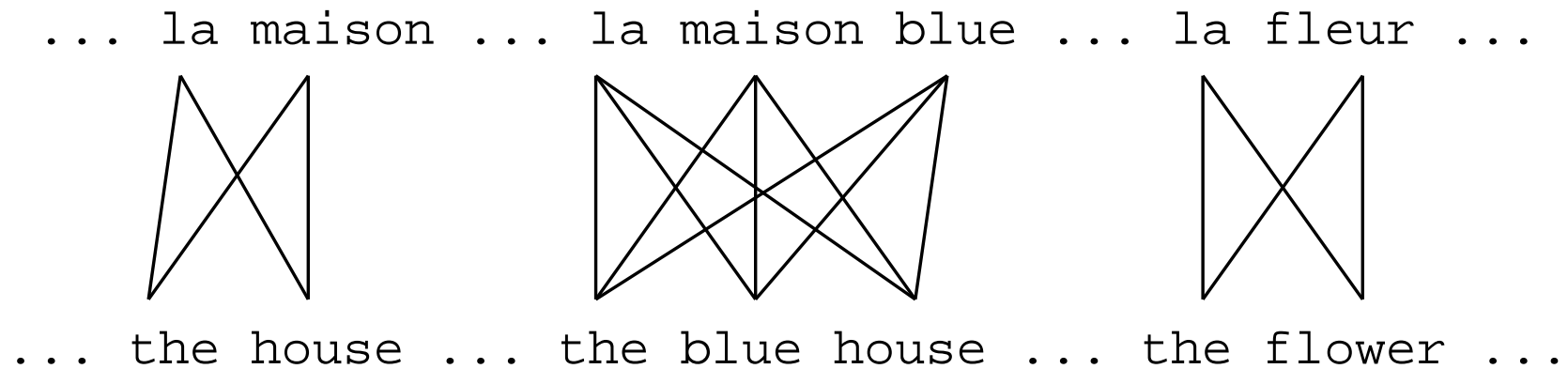
- 從平行語料庫中，估算詞彙翻譯機率 lexical translation probabilities $t(e|f)$
- 但是，我們沒有詞彙對應 *alignments* 的資料（不知道哪個詞對哪個詞）
- 雞生蛋，蛋生雞的問題
 - 假如，我們有對應 *alignments* → 我們就可用次數估算模型的參數 *prob.*
 - 假如，我們有機率模型參數 *prob.* → 我們就可推測詞彙對應 *alignments*

EM 演算法

用生成性模型補齊不完整的資料

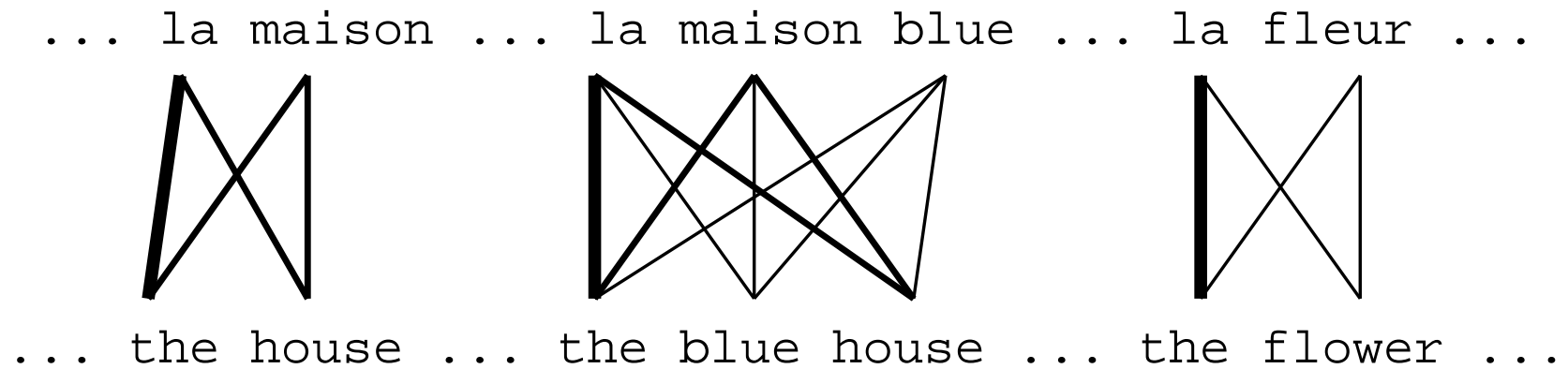
- 在不完整資訊資料 incomplete data 中推測補足資訊（最佳化生成性模型）
 - 如果我們有完整資料（如詞彙對應的註記），我們就可以估算機率模型
 - 如果我們有機率模型（詞彙的可能翻譯），我們就可以填補資料的缺口
- 簡介「期望值與最大似然演算法」Expectation Maximization (EM)
 1. 初始化模型參數 (平均分布：任意英語能翻譯任何德語，機率相同)
(非常不正確，但是資料的資訊可以限制之，補模型之不足)
 2. 計算每一筆樣本中的不足資料（詞與詞的對應）的期望值（次數）
越高越好
 3. 用所有樣本的（推測）次數來估算參數
 4. 反覆執行步驟 2-3 直到模型參數收斂 convergence（幾乎沒有變化）

演示 EM 演算法



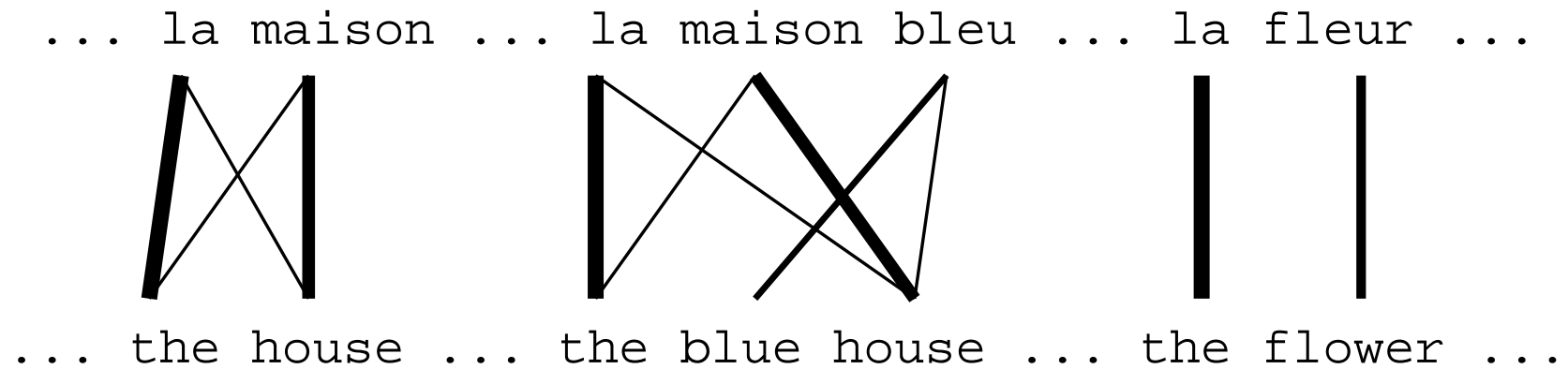
- 初始步驟：任何對應（缺口資訊）都有可能，機率 = $\frac{1}{(l_f+1)}$ (或簡化為 $\frac{1}{l_f}$)
- 在每句中 **the** 的對應一開始不確定
- 但是，把不確定次數加總，模型漸漸學到 **the** 對應到 **la** 的確機率很高
- 下一迴圈，**the** 對應到 **la** 的確定性提高，期望值也提高

EM 演算法



- 一次迴圈後
- 真正的對應 (如 **la** 和 **the**) 愈來愈可能

EM 演算法



- 再一個迴圈
- 愈來愈明顯，**fleur** 可能對應到 **flower** (鴿洞原理，**la** 不可能對應到 **flower**)

EM 演算法

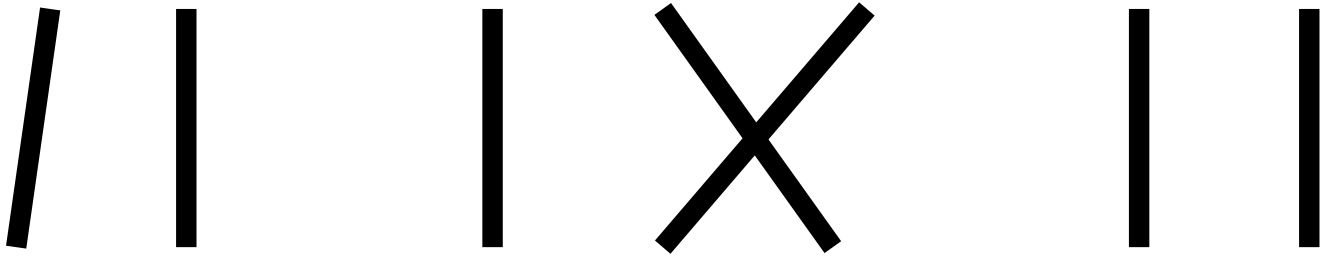
... la maison ... la maison bleu ... la fleur ...
/ | | X | |
... the house ... the blue house ... the flower ...

- 幾個迴圈後，模型收斂
- EM 終於揭露了隱藏的結構

EM 演算法

- 語料庫對應好了，用最後的結果估算參數（機率函數）

... la maison ... la maison bleu ... la fleur ...
... the house ... the blue house ... the flower ...



$$\begin{aligned}p(\text{la}|\text{the}) &= 0.453 \\p(\text{le}|\text{the}) &= 0.334 \\p(\text{maison}|\text{house}) &= 0.876 \\p(\text{bleu}|\text{blue}) &= 0.563 \\&\dots\end{aligned}$$

IBM 模型 1 和 EM 演算法

- EM 演算法有兩個步驟
 - 期望值步驟：用模型來處理資料
 - * 對於資訊隱藏的部份 (詞到詞的對應)
 - * 運用模型，指定事件數值的機率
 - 最大似然步驟：用資料隱藏資訊的期望值，估算模型參數
 - * 取模型指定的事件值（與機率值）
 - * 累積事件的次數 (不是完整的次數，而是機率加權的零頭次數)
 - * 用詞數估計模型參數的機率值
- 反覆執行這兩個步驟，直到收斂為止

IBM 模型 1 和 EM 演算法

- 所以，我們需要計算：
 - E 步驟：句中詞彙對應的機率（用翻譯機率）
 - M 步驟：計算整個語料庫的詞對詞次數（加總零頭次數）

IBM 模型 1 和 EM 演算法

- 詞彙翻譯機率

$$\begin{aligned} p(\text{the}|\text{la}) &= 0.7 & p(\text{house}|\text{la}) &= 0.05 \\ p(\text{the}|\text{maison}) &= 0.1 & p(\text{house}|\text{maison}) &= 0.8 \end{aligned}$$

- 某句的整體對應



$$p(\mathbf{e}, a|\mathbf{f}) = 0.56 \quad p(\mathbf{e}, a|\mathbf{f}) = 0.035 \quad p(\mathbf{e}, a|\mathbf{f}) = 0.08 \quad p(\mathbf{e}, a|\mathbf{f}) = 0.005$$

$$p(a|\mathbf{e}, \mathbf{f}) = 0.824 \quad p(a|\mathbf{e}, \mathbf{f}) = 0.052 \quad p(a|\mathbf{e}, \mathbf{f}) = 0.118 \quad p(a|\mathbf{e}, \mathbf{f}) = 0.007$$

取有貢獻的alignment

- 計算次數

$$\begin{aligned} c(\text{the}|\text{la}) &= 0.824 + 0.052 & c(\text{house}|\text{la}) &= 0.052 + 0.007 \\ c(\text{the}|\text{maison}) &= 0.118 + 0.007 & c(\text{house}|\text{maison}) &= 0.824 + 0.118 \end{aligned}$$

IBM 模型 1 和 EM: E 步驟

- 我們需要計算 $p(a|\mathbf{e}, \mathbf{f})$
- 運用 chain rule : $p(a|\mathbf{e}, \mathbf{f}) = \frac{p(\mathbf{e}, a|\mathbf{f})}{p(\mathbf{e}|\mathbf{f})}$
- 我們已經有 $p(\mathbf{e}, a|\mathbf{f})$ 的式子 (Model 1 定義，講異第 12 頁)

IBM 模型 1 和 EM: E 步驟的分母

- 我們還需要計算 $p(\mathbf{e}|\mathbf{f})$ （除了這個分母，機率加總才會等於 0）
- 計算需要窮舉所有的對應，一個組合（指數型）的計算

$$\begin{aligned} p(\mathbf{e}|\mathbf{f}) &= \sum_a p(\mathbf{e}, a|\mathbf{f}) \quad \text{看哪個英文翻到 F} \\ &= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} p(\mathbf{e}, a|\mathbf{f}) \\ &= \sum_{a(1)=0}^{l_f} \dots \sum_{a(l_e)=0}^{l_f} \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) \end{aligned}$$

IBM 模型 1 和 EM: E 步驟的簡化

$$\begin{aligned} p(\mathbf{e}|\mathbf{f}) &= \sum_{a(1)=0}^{l_f} \cdots \sum_{a(l_e)=0}^{l_f} \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) \\ &= \frac{\epsilon}{(l_f + 1)^{l_e}} \sum_{a(1)=0}^{l_f} \cdots \sum_{a(l_e)=0}^{l_f} \prod_{j=1}^{l_e} t(e_j | f_{a(j)}) \\ &= \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j | f_i) \end{aligned}$$

- 提示：最後一行的技巧，避免衆多機率的乘積 (本來需乘 $(l_f + 1)^{l_e}$ 次)
- (所有對齊的加權 = 除以 $\text{sum}(\text{所有外語 } f_0^{l_f} \text{ 到 } e \text{ 的翻譯機率})$)
→ 這樣 IBM 模型 1 的估算變得可行 tractable (不需要指數型的時間)

技巧：因素分解 $\mathbf{ax+ay+bx+by} = \mathbf{a(x+y)+b(x+y)}$

(case $l_e = l_f = 2$)

$$\sum_{a(1)=0}^2 \sum_{a(2)=0}^2 \prod_{j=1}^2 t(e_j|f_{a(j)}) =$$

$$\begin{aligned} &= t(e_1|f_0) t(e_2|f_0) + t(e_1|f_0) t(e_2|f_1) + t(e_1|f_0) t(e_2|f_2) + \\ &\quad t(e_1|f_1) t(e_2|f_0) + t(e_1|f_1) t(e_2|f_1) + t(e_1|f_1) t(e_2|f_2) + \\ &\quad t(e_1|f_2) t(e_2|f_0) + t(e_1|f_2) t(e_2|f_1) + t(e_1|f_2) t(e_2|f_2) \end{aligned}$$

$$\begin{aligned} &= t(e_1|f_0) (t(e_2|f_0) + t(e_2|f_1) + t(e_2|f_2)) + \\ &\quad t(e_1|f_1) (t(e_2|f_0) + t(e_2|f_1) + t(e_2|f_2)) + \\ &\quad t(e_1|f_2) (t(e_2|f_0) + t(e_2|f_1) + t(e_2|f_2)) \end{aligned}$$

$$= (t(e_1|f_0) + t(e_1|f_1) + t(e_1|f_2)) (t(e_2|f_0) + t(e_2|f_1) + t(e_2|f_2))$$

IBM 模型 1 和 EM: E 步驟

- 合併起來：

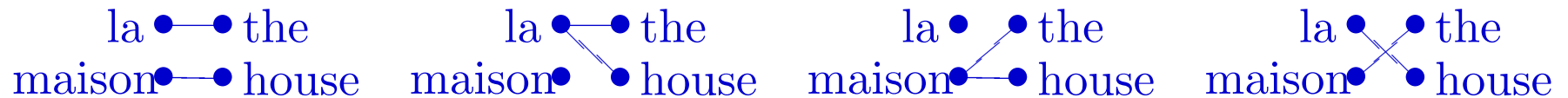
$$\begin{aligned} p(\mathbf{a}|\mathbf{e}, \mathbf{f}) &= p(\mathbf{e}, \mathbf{a}|\mathbf{f}) / \underline{p(\mathbf{e}|\mathbf{f})} \\ &= \frac{\frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)})}{\frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i)} \xrightarrow{\text{simplify}} \\ &= \prod_{j=1}^{l_e} \frac{t(e_j|f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j|f_i)} \end{aligned}$$

演示：不需要展開所有可能的對應

- 詞彙翻譯機率

$$\begin{array}{ll} p(\text{the}|\text{la}) = 0.7 & p(\text{house}|\text{la}) = 0.05 \\ p(\text{the}|\text{maison}) = 0.1 & p(\text{house}|\text{maison}) = 0.8 \end{array}$$

- 某句的整體對應



$$p(\mathbf{e}, a|\mathbf{f}) = 0.56 \quad p(\mathbf{e}, a|\mathbf{f}) = 0.035 \quad p(\mathbf{e}, a|\mathbf{f}) = 0.08 \quad p(\mathbf{e}, a|\mathbf{f}) = 0.005$$

$$p(a|\mathbf{e}, \mathbf{f}) = 0.824 \quad p(a|\mathbf{e}, \mathbf{f}) = 0.052 \quad p(a|\mathbf{e}, \mathbf{f}) = 0.118 \quad p(a|\mathbf{e}, \mathbf{f}) = 0.007$$

- 計算次數

$$\begin{array}{ll} c(\text{the}|\text{la}) = 0.824 + 0.052 & c(\text{house}|\text{la}) = 0.052 + 0.007 \\ c(\text{the}|\text{maison}) = 0.118 + 0.007 & c(\text{house}|\text{maison}) = 0.824 + 0.118 \end{array}$$

IBM 模型 1 和 EM: M 步驟

- 接著，我們收集詞與詞對應，計算次數：
- 句子翻譯配對 \mathbf{e}, \mathbf{f} 中，英語詞 e 是外語詞 f 的翻譯，有如下的零頭次數：

$$c(e|f; \mathbf{e}, \mathbf{f}) = \sum_a p(a|\mathbf{e}, \mathbf{f}) \sum_{j=1}^{l_e} \delta(e, e_j) \delta(f, f_{a(j)})$$

- 依照前面的公式，簡化（所有對齊的加權 = 除以 $\text{sum}(\text{所有外語 } f_0^{l_f} \text{ 到 } e \text{ 的翻譯機率})$ ）

$$c(e|f; \mathbf{e}, \mathbf{f}) = \frac{t(e|f)}{\sum_{i=0}^{l_f} t(e|f_i)} \sum_{j=1}^{l_e} \delta(e, e_j) \sum_{i=0}^{l_f} \delta(f, f_i)$$

IBM Model 1 and EM: Maximization Step

- 收集整個語料庫的所有句子與翻譯配對後，我們就可以估算模型參數：

$$t(e|f; \mathbf{e}, \mathbf{f}) = \frac{\sum_{(\mathbf{e}, \mathbf{f})} c(e|f; \mathbf{e}, \mathbf{f}))}{\sum_f \sum_{(\mathbf{e}, \mathbf{f})} c(e|f; \mathbf{e}, \mathbf{f}))}$$

IBM Model 1/EM 演算法：虛擬程式碼

Input: set of sentence pairs (\mathbf{e}, \mathbf{f})

Output: translation prob. $t(e|f)$

```
1: initialize  $t(e|f)$  uniformly
2: while not converged do
3:   // initialize
4:    $\text{count}(e|f) = 0$  for all  $e, f$ 
5:    $\text{total}(f) = 0$  for all  $f$ 
6:   for all sentence pairs ( $\mathbf{e}, \mathbf{f}$ ) do
7:     // compute normalization
8:     for all words  $e$  in  $\mathbf{e}$  do
9:        $\text{s-total}(e) = 0$ 
10:      for all words  $f$  in  $\mathbf{f}$  do
11:         $\text{s-total}(e) += t(e|f)$ 
12:      end for
13:    end for
```

```
14:   // collect counts
15:   for all words  $e$  in  $\mathbf{e}$  do
16:     for all words  $f$  in  $\mathbf{f}$  do
17:       計算每個word期望值 :  $\text{count}(e|f) += \frac{t(e|f)}{\text{s-total}(e)}$ 
18:        $\text{total}(f) += \frac{t(e|f)}{\text{s-total}(e)}$ 
19:     end for
20:   end for
21: end for
22: // estimate probabilities
23: for all foreign words  $f$  do
24:   for all English words  $e$  do
25:      $t(e|f) = \frac{\text{count}(e|f)}{\text{total}(f)}$ 
26:   end for
27: end for
28: end while
```

E 步驟 6-21 行

- 8-13 行 (s-total):

$$\sum_{i=0}^{l_f} t(e|f_i)$$

- 17 行: (加權詞彙機率 $t(e|f)$)

$$\frac{t(e|f)}{\sum_{i=0}^{l_f} t(e|f_i)}$$

- 18 行 (total)

$$c(e|f; \mathbf{e}, \mathbf{f}) = \frac{t(e|f)}{\sum_{i=0}^{l_f} t(e|f_i)} \sum_{j=1}^{l_e} \delta(e, e_j) \sum_{i=0}^{l_f} \delta(f, f_i)$$

M 步驟 23-27 行

- 在 M-步驟中，累積 $\text{count}(e | f)$ 和 $\text{total}(f)$ 後，我們估算 $t(e | f)$:

$$t(e|f; \mathbf{e}, \mathbf{f}) = \frac{\sum_{(\mathbf{e}, \mathbf{f})} c(e|f; \mathbf{e}, \mathbf{f})}{\sum_f \sum_{(\mathbf{e}, \mathbf{f})} c(e|f; \mathbf{e}, \mathbf{f})}$$

- EM 演算法: 反覆執行 2-28 行

收斂

das Haus
the house

das Buch
the book

ein Buch
a book

e	f	initial	1st it.	2nd it.	3rd it.	...	final
the	das	0.25	0.5	0.6364	0.7479	...	1
book	das	0.25	0.25	0.1818	0.1208	...	0
house	das	0.25	0.25	0.1818	0.1313	...	0
the	buch	0.25	0.25	0.1818	0.1208	...	0
book	buch	0.25	0.5	0.6364	0.7479	...	1
a	buch	0.25	0.25	0.1818	0.1313	...	0
book	ein	0.25	0.5	0.4286	0.3466	...	0
a	ein	0.25	0.5	0.5714	0.6534	...	1
the	haus	0.25	0.5	0.4286	0.3466	...	0
house	haus	0.25	0.5	0.5714	0.6534	...	1

複雜度 perplexity

- 最後模型很合適資料嗎？
- 複雜度：用訓練資料的機率導出複雜度

$$\log_2 PP = - \sum_s \log_2 p(\mathbf{e}_s | \mathbf{f}_s)$$

- 範例 (假設 $\epsilon=1$)

機率提高，取對數越低

機率 / 複雜度	起始	第1回	第2回	第3回	...	最後
$p(\text{the haus} \text{das haus})$	0.0625	0.1875	0.1905	0.1913	...	0.1875
$p(\text{the book} \text{das buch})$	0.0625	0.1406	0.1790	0.2075	...	0.25
$p(\text{a book} \text{ein buch})$	0.0625	0.1875	0.1907	0.1913	...	0.1875
複雜度	4095	202.3	153.6	131.6	...	113.8

如何確保翻譯流暢 **Fluent Output**

- 光有詞彙 (如 小) 的翻譯，無法決定用 **small** 或 **little** (總不能都用最高頻)
- 要看上下文 (如 **step**) 決定 **small** 或 **little** 比較合適。
- 直覺上，可以查 Google 採用次數比較高的那一個
 - **small step** 2,070,000
 - **little step** 257,000 language model : markov 使用前n word 預測下一個單字
- 相當於「n-連詞語言模型」——用 n-連詞統計資訊，估算英文詞串的機率

$$\begin{aligned} p(\mathbf{e}) &= p(e_1, e_2, \dots, e_n) \\ &= p(e_1)p(e_2|e_1)\dots p(e_n|e_1, e_2, \dots, e_{n-1}) \\ &\simeq p(e_1)p(e_2|e_1)\dots p(e_n|e_{n-2}, e_{n-1}) \end{aligned}$$

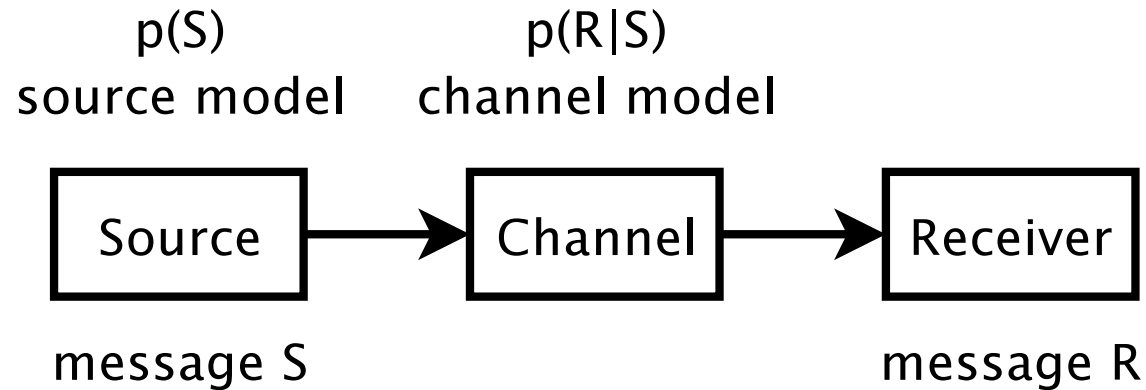
雜訊通道模型 Noisy Channel Model

- 把語言模型整合到機器翻譯系統
- 如何：貝氏定理 Bayes rule

$$\begin{aligned}\operatorname{argmax}_e p(e|f) &= \operatorname{argmax}_e \frac{p(f|e) p(e)}{p(f)} \\ &= \operatorname{argmax}_e \underbrace{p(f|e)}_{\substack{\text{因應輸入值此值可變化}}} p(e)\end{aligned}$$

- 因為 $p(f)$ 是輸入，數值不變，所以可以省略

雜訊通道模型 Noisy Channel Model



- 運用貝氏定理，得到 noisy channel model
 - 看到有雜訊的訊息 R (外語 f)
 - 模型描述如何產生這樣的雜訊 (翻譯模型)
 - 模型描述正確的訊息 S 如何產生 (語言模型)
 - 目的：如何把 R 轉換（解碼）為 S (英語句 e)

更高階的 IBM 模型 2-5

IBM Model 1	詞彙翻譯
IBM Model 2	加入絕對位置重組模型 absolute reordering model
IBM Model 3	加入滋生性模型 fertility model ->可描述一對多情況
IBM Model 4	改成相對性重組模型 relative reordering model
IBM Model 5	處理公式的缺項 deficiency

- 只有 IBM 模型 1 可以收斂到全域最佳值 global maximum
 - 用模型 1 初始化，然後訓練模型 2 (類推到模型 3, 4, 5)
- 模型 3 的計算量非常大，困難實施
 - 簡化技巧行不通，以至於全面的計算次數，變得花費太多時間
 - 方法：取樣以減少計算量（取哪些呢？高機率的詞與詞對應）

Reminder: IBM Model 1

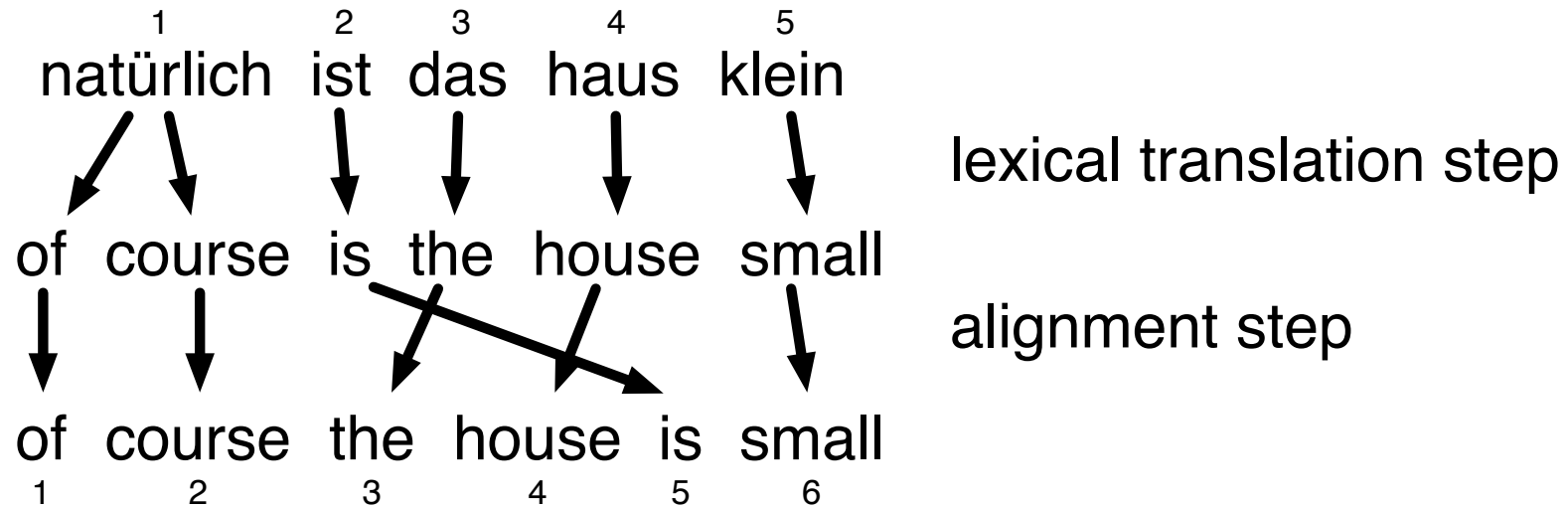
- 生成性模型：把句子翻譯過程，分解更小的詞的翻譯小步驟
 - IBM Model 1 只用到詞彙翻譯的資訊（不管詞的位置的關係）
- 翻譯機率 translation probability
 - 給予外語句 $\mathbf{f} = (f_1, \dots, f_{l_f})$ l_f 為其長度
 - to an English sentence $\mathbf{e} = (e_1, \dots, e_{l_e})$ of length l_e
 - 以及對應的英語翻譯句， e_j 為其長度
 - 對齊函數 $a : j \rightarrow i$ 則

$$p(\mathbf{e}, a | \mathbf{f}) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j | f_{a(j)})$$

- ϵ 是正規化的常數

IBM Model 2

- 加入（絕對位置）對齊機率



IBM 模型 2

- 在詞彙之上，增加一個（決定位置）對齊機率分布 alignment probability
- 此一函數，描述第 i 外語詞對齊到第 j 個英語詞的機率

$$a(i|j, l_e, l_f)$$

- 把方程式合在一起，得到

$$p(\mathbf{e}, \mathbf{a}|\mathbf{f}) = \epsilon \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) a(a(j)|j, l_e, l_f)$$

- 這個 IBM 模型 2 的 EM 訓練演算法，和 IBM 模型 1 的演算法，完全一樣

插曲：HMM 模型

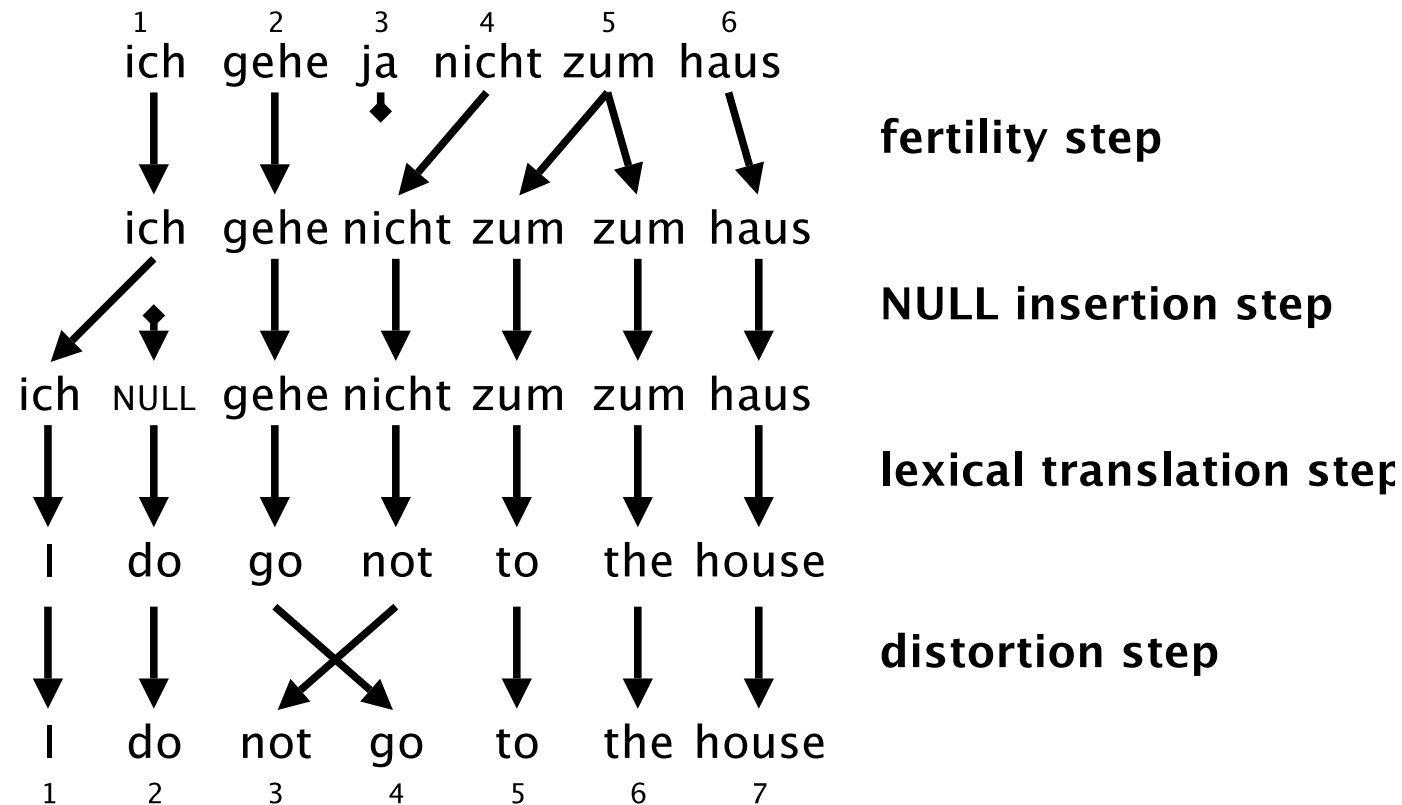
- 詞翻譯到目標語言時，不會各自獨立移動
 - 而是一組一組的一起移動
 - 因此，其移動應該以前一詞移動量為條件，設定為（相對性）移動條件機率
- HMM 對齊模型：

$$p(a(j)|a(j-1), l_e)$$

- 比較困難使用 EM 演算法
- 在 E-步驟，需要用動態規劃 dynamic programming 來估算機率
- IBM 模型 4 類似，而且其條件是詞的群組，而不是詞

IBM 模型 3

- 加入（特定詞彙）的滋生度機率



IBM 模型 3: 滋生度

- 滋生度：外語詞翻譯成英語詞的個數
- 滋生度的機率函數： $n(\phi|f)$
- 例子：

$$n(1|\text{haus}) \simeq 1$$

$$n(2|\text{zum}) \simeq 1$$

$$n(0|\text{ja}) \simeq 1$$

在對齊空間取樣本

- 用 EM 演算法，訓練 IBM 模型 3
 - 簡化技巧行不通
 - 全面的計算次數，變得花費太多時間
- 用 hillclimbing 的方式，找到最可能的對齊：
 - 以某一對齊開始（用較低階的翻譯模型）
 - 改變個別詞的對應詞（以提高機率值）
 - 反覆執行，直到收斂
- 抽樣:：蒐集對齊的各種變形 variations，加以統計
 - 在 hillclimbing 發現的對齊 alignments 都列入考慮
 - 哪些：相差不多（一次移動或互換）的對齊「鄰居」neighbors

IBM 模型 4

- 更精確的詞序重組模型 reordering model

- 在 IBM 模型 2 和 3 （一樣）

$$d(j||i, l_e, l_f)$$

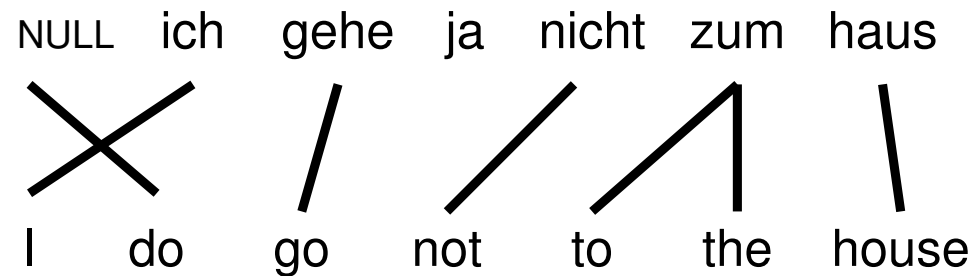
- 這個絕對位置（ i 和 j ）對於短句很有效
- 對長句效果不佳（高 l_f 和 l_e ）導致複雜度高、資料稀疏、統計不可靠

- 觀察

- 翻譯過程中，詞不是獨立移動（和絕對位置無關）
- 而是和前後詞一起移動（相對位置複雜度低）
- 用相對於前一詞的移動位置，比較好

IBM 模型 4: 詞組 cepts

- 外語詞編組：對應到同樣英語詞一同組，無對應者——忽略之（共5組）
- 計算每組的英語詞中心位置，取平均值（小數點進位） $\text{ceiling}(\text{avg}(j))$



外語詞組 π_i	π_1	π_2	π_3	π_4	π_5
外語詞位置 i	1	2	4	5	6
外語詞 f_i	ich	gehe	nicht	zum	haus
英語詞 (可多詞) $\{e_j\}$	I	go	not	to,the	house
英語詞位置 $\{j\}$	1	4	3	5,6	7
詞組翻譯中心 \odot_i	1	4	3	6	7

IBM 模型 4：相對扭曲 distortion

- 參考位置：前一組 \odot_{i-1} 的中心位置
- 相對扭曲：外語詞 f_i 的翻譯 e_j 的相對扭曲 $j - \odot_{i-1}$
- 分成三種狀況：
 - 如果 $i = 0, f_0 = null$ ，則相對扭曲機率 = 1（平均機率）
 - 對每組第一詞：用函數 d_1
 - 對每組其他詞：用函數 $d_{>1}$

英語詞位置 j	1	2	3	4	5	6	7
英語詞 e_j	I	do	not	go	to	the	house
詞群位置	$\pi_{1,0}$	$\pi_{0,0}$	$\pi_{3,0}$	$\pi_{2,0}$	$\pi_{4,0}$	$\pi_{4,1}$	$\pi_{5,0}$
詞組中心 \odot_{i-1}	0	-	4	1	3	-	6
扭曲 $j - \odot_{i-1}$	+1	-	-1	+3	+2	-	+1
扭曲函數	$d_1(+1)$	1	$d_1(-1)$	$d_1(+3)$	$d_1(+2)$	$d_{>1}(+1)$	$d_1(+1)$

相對於前一個字的翻譯

扭曲機率：加入詞彙、詞群的條件

- 考慮到不同的詞彙引發不同的扭曲：加入詞彙的條件
 - 對每組的第一詞： $d_1(j - \odot_{i-1} | f_{i-1}, e_j)$
 - 對每組的其他詞： $d_{>1}(j - \pi_{i,k-1} | e_j)$
- 以詞為條件可能造成資料稀疏，可以改用詞群 A 和 B
 - 對每組的第一詞： $d_1(j - \odot_{[i-1]} | A(f_{i-1}), B(e_j))$
 - 對每組的其他詞： $d_{>1}(j - \pi_{i,k-1} | B(e_j))$

IBM 模型 5

- IBM 模型 1-4 漏列一些機率
 - 有些不可能的翻譯，有正的機率值（應該是0）
 - 位置函數的描述不嚴謹（多個翻譯詞，可以擺在同一位置→ 有些機率值沒有列出，機率總數 mass 浪費掉了
- IBM 模型 5 記錄空位（尚未被翻譯詞佔用），以修正以上的缺失

IBM 模型小結

- IBM 模型是最早的統計式機器翻譯模型
- IBM 模型介紹了重要的新概念
 - 生成性模型
 - EM 訓練法
 - 詞序重組模型
- 運用在詞彙對應工具 (如 GIZA++)
- 比較少原原本本地用來「解碼」(只用在特定應用，如簡體到繁體的轉換)

詞彙對齊 Word Alignment

	michael	geht	davon	aus	,	dass	er	im	haus	bleibt
michael										
assumes										
that										
he										
will										
stay										
in										
the										
house										

IBM 模型 Word Alignment?

- 有時（如英語 *does*），難決定對應詞—德語 *wohnt* (動詞) 或 *nicht* (否定)

	john	wohnt	hier	nicht
john				
does		?		?
not				
live				
here				

詞彙搭配適當嗎？

- 對於英語成語 **kicked the bucket**（字面：踢桶子）以及德語 **biss ins grass**（字面：吃草）很難決定以詞的層次，應該如何對齊
- 詞彙對應不恰當—在此之外，**bucket** 從來就不是翻譯成 **grass**

	john	biss	ins	grass
john				
kicked				
the				
bucket				

評估詞彙對齊的品質

- 人工對齊語料庫：標註對應，品質分兩種 *sure* (S) 和 *sure or possible* (P)
alignment points ($S \subseteq P$)
- 常用評估指標：對應錯誤率 Alignment Error Rate (AER)

$$\text{PREC}(A) = \frac{|A \cap P|}{|A|} \quad \text{RECALL}(A) = \frac{|A \cap S|}{|S|}$$

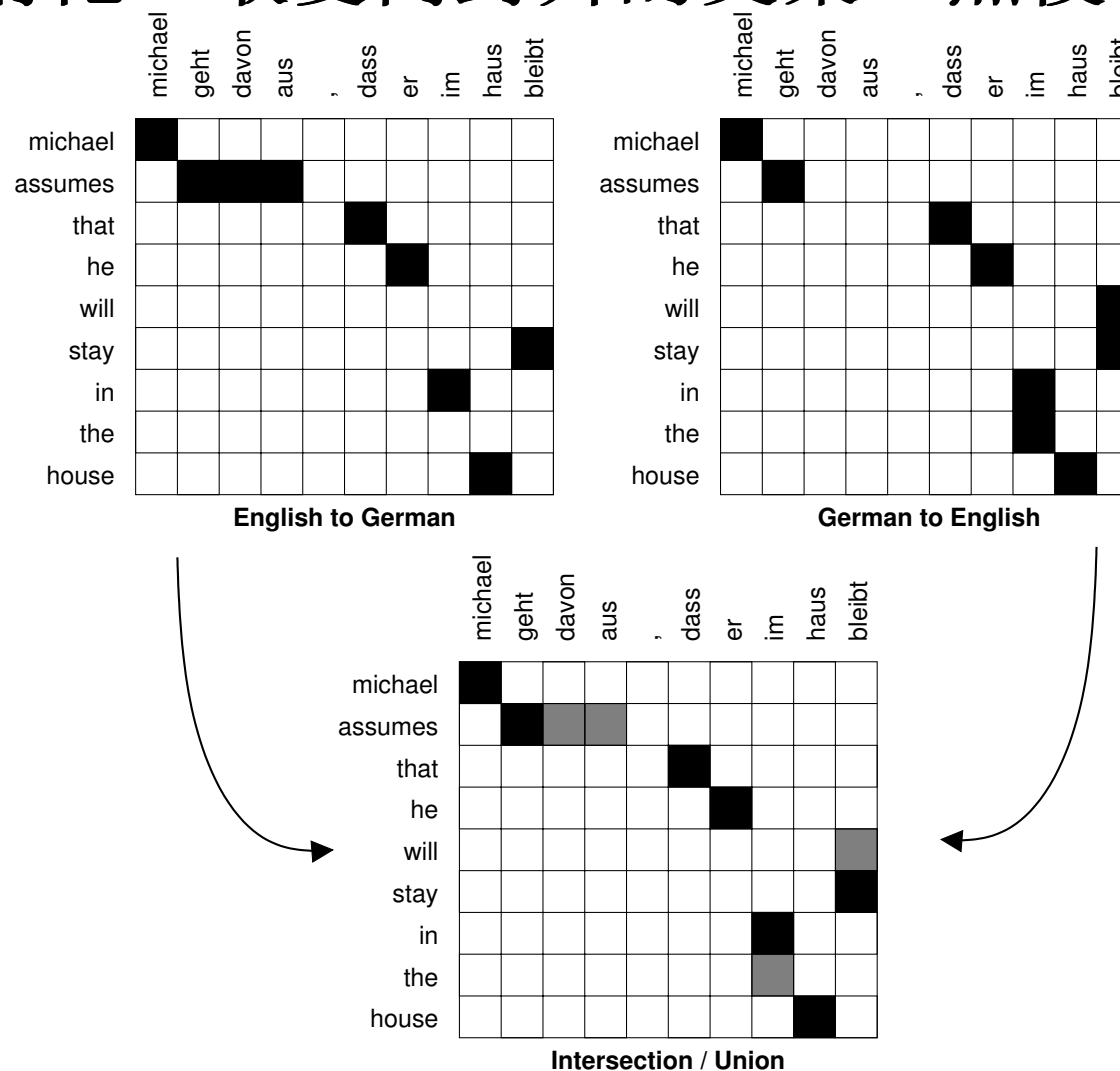
$$\text{AER}(S, P; A) = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}$$

- $\text{AER} = 0$ 如果對應答案 A 涵蓋所有的 S 和部分的 P (強調查全率 RECALL)
- 不同的應用，可能會強調精確率 (PREC)

用 IBM 模型對齊詞彙 Word Alignment

- IBM 模型產生 多對一 的詞彙對齊
 - 用對齊函數表示詞與詞對應資訊
 - 多個輸入詞可以透過對齊函數，對應到單一輸出 (多對一)
 - 然而，函數不能回傳多個數值 (不能表示一對多)
- 真正的詞彙對齊，有可能是「多對多」 (不能表達為函數)

對稱化—取雙向對齊的交集，然後增長



增長對齊點的策略 Growing heuristic

grow-diag-final(e2f,f2e)

- 1: neighboring = $\{(-1,0),(0,-1),(1,0),(0,1),(-1,-1),(-1,1),(1,-1),(1,1)\}$
- 2: alignment $A = \text{intersect}(e2f,f2e)$; **grow-diag**(); **final**(e2f); **final**(f2e);

grow-diag()

- 1: **while** new points added **do**
- 2: **for all** English word $e \in [1...e_n]$, foreign word $f \in [1...f_n]$, $(e, f) \in A$ **do**
- 3: **for all** neighboring alignment points $(e_{\text{new}}, f_{\text{new}})$ **do**
- 4: **if** $(e_{\text{new}} \text{ unaligned OR } f_{\text{new}} \text{ unaligned}) \text{ AND } (e_{\text{new}}, f_{\text{new}}) \in \text{union}(e2f,f2e)$ **then**
- 5: add $(e_{\text{new}}, f_{\text{new}})$ to A
- 6: **end if**
- 7: **end for**
- 8: **end for**
- 9: **end while**

final()

- 1: **for all** English word $e_{\text{new}} \in [1...e_n]$, foreign word $f_{\text{new}} \in [1...f_n]$ **do**
- 2: **if** $(e_{\text{new}} \text{ unaligned OR } f_{\text{new}} \text{ unaligned}) \text{ AND } (e_{\text{new}}, f_{\text{new}}) \in \text{union}(e2f,f2e)$ **then**
- 3: add $(e_{\text{new}}, f_{\text{new}})$ to A
- 4: **end if**
- 5: **end for**

最新的對稱化的研究

- 每次 EM 的迴圈都做一次對稱化動作 [Matusov et al., 2004]
 - 對兩個方向各做一次 E-步驟，然後做對稱化
 - 計算整個語料庫的詞與詞對應次數 (M-步驟)
- 在對稱化過程中，使用後驗機率 posterior probabilities
 - 每個方向，產生 n-best 詞彙對齊 alignments
 - 計算對齊點的發生頻率
 - 在對稱化過程中，使用後驗機率 posterior

對應鏈結刪增模型 Link Deletion / Addition

- 鏈結刪除法 [Fossum et al., 2008]
 - 開始：雙向 IBM 模型的對應點的聯集
 - 然後：使用類神經網路分類器一次刪除一個對應點
(考慮刪除連接點對於後續學習文法規則的有效性)
- 鏈結增加法 [Ren et al., 2007] [Ma et al., 2008]
 - 開始：或許就由一組高機率的對應點
 - 然後，一次增加一個對應點

判別式的訓練法 Discriminative Training Methods

- 必須使用有標示的資料 (標示詞與詞對應) 以便採用督導式學習方法
- 結構預測 Structured prediction
 - 不是單一分類問題
 - 解答有多筆互相關連資料，必須按步驟求解
- 有很多不同的方法：maximum entropy, neural networks, support vector machines, conditional random fields, MIRA
- 可以做非督導式學習（如何對應），然後用少量資料，調整模型參數 [Fraser and Marcu, 2007]

更好的生成式模型

- 直接對應片語
 - 聯合機率模型 joint model [Marcu and Wong, 2002]
 - 問題：EM 演算法會（不正確地）優先採用常詞
- Fraser 的 LEAF 法
 - 把 word alignment 分結成幾個步驟（類似 IBM 模型）
 - 其中一步，把詞結合成「片語」

Fast_align 更好的詞序重組模型 減少參數 → 增加效率

- 分享程式碼：github.com/clab/fast_align
- 重點：位置對應的參數化 simple log-linear reparameterization

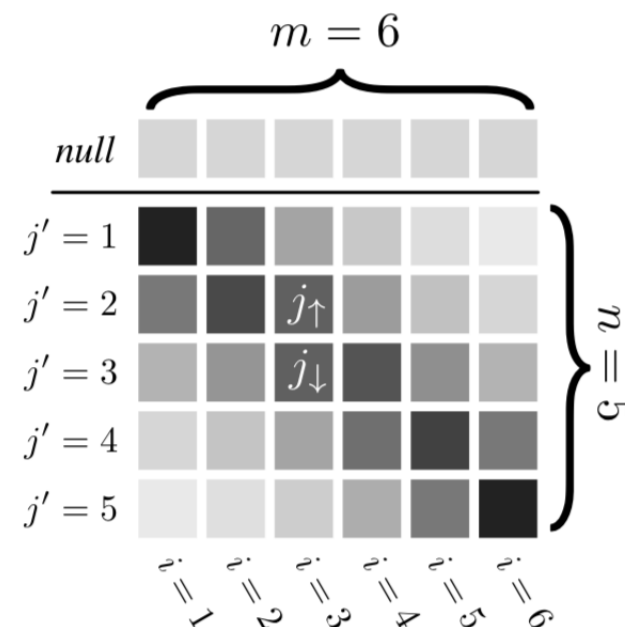
Given : \mathbf{f} , $n = |\mathbf{f}|$, $m = |\mathbf{e}|$, p_0 , λ , θ

$$h(i, j, m, n) = - \left| \frac{i}{m} - \frac{j}{n} \right|$$

$$\delta(a_i = j \mid i, m, n) = \begin{cases} p_0 & j = 0 \\ (1 - p_0) \times \frac{e^{\lambda h(i, j, m, n)}}{Z_\lambda(i, m, n)} & 0 < j \leq n \\ 0 & \text{otherwise} \end{cases}$$

$$a_i \mid i, m, n \sim \delta(\cdot \mid i, m, n) \quad 1 \leq i \leq m$$

$$e_i \mid a_i, f_{a_i} \sim \theta(\cdot \mid f_{a_i}) \quad 1 \leq i \leq m$$



更適合的分詞處理

- 讓中文斷詞（「末日 vs. 末、日」）能夠和英文的分詞（last day）一致
 - 增加 1-1 減少 2-1, 1-2 對應
 - 提高對齊模型的精確度
 - 提高對齊工具的精確度
 - 全面改變斷詞 vs. 現有斷詞＋後處理
- 整合「華語詞」、「華語字」和「英文詞」的兩種、雙向模型
- 加入更多資訊
 - 使用有標註的訓練資料（supervised learning/self or distant supervision）
 - 運用語言資訊：詞性、構詞、時態
 - 運用統計資訊：相互資訊、IBM 模型 1 (詞對詞＋字對詞＋字對字)

摘要

- 詞彙翻譯是核心
- 對齊函數 Alignment 形式化地處理翻譯
- EM 演算法可以發掘文本翻譯隱藏的結構
- 雜訊通道模型 Noisy Channel Model 整合翻譯模型、語言模型
- IBM 模型 1–5
 - IBM 模型 1: 詞彙翻譯 lexical translation
 - IBM 模型 2: 對齊模型 alignment model
 - IBM 模型 3: 滋生度 fertility
 - IBM 模型 4: 相對對齊扭曲模型 relative alignment model

- IBM 模型 5: 補充缺失 deficiency