

Digital Speech Processing

數位語音處理概論

第六講 Language Modelling

授課教師：國立臺灣大學 電機工程學系 李琳山 教授



【本著作除另有註明外，採取 [創用 CC](#)
「姓名標示-非商業性-相同方式分享」台灣 3.0 版
授權釋出】

References: 1. 11.2.2, 11.3, 11.4 of Huang or
2. 6.1- 6.8 of Becchetti, or
3. 4.1- 4.5, 8.3 of Jelinek

Language Modeling: providing linguistic constraints to help the selection of correct words

Fig. 1

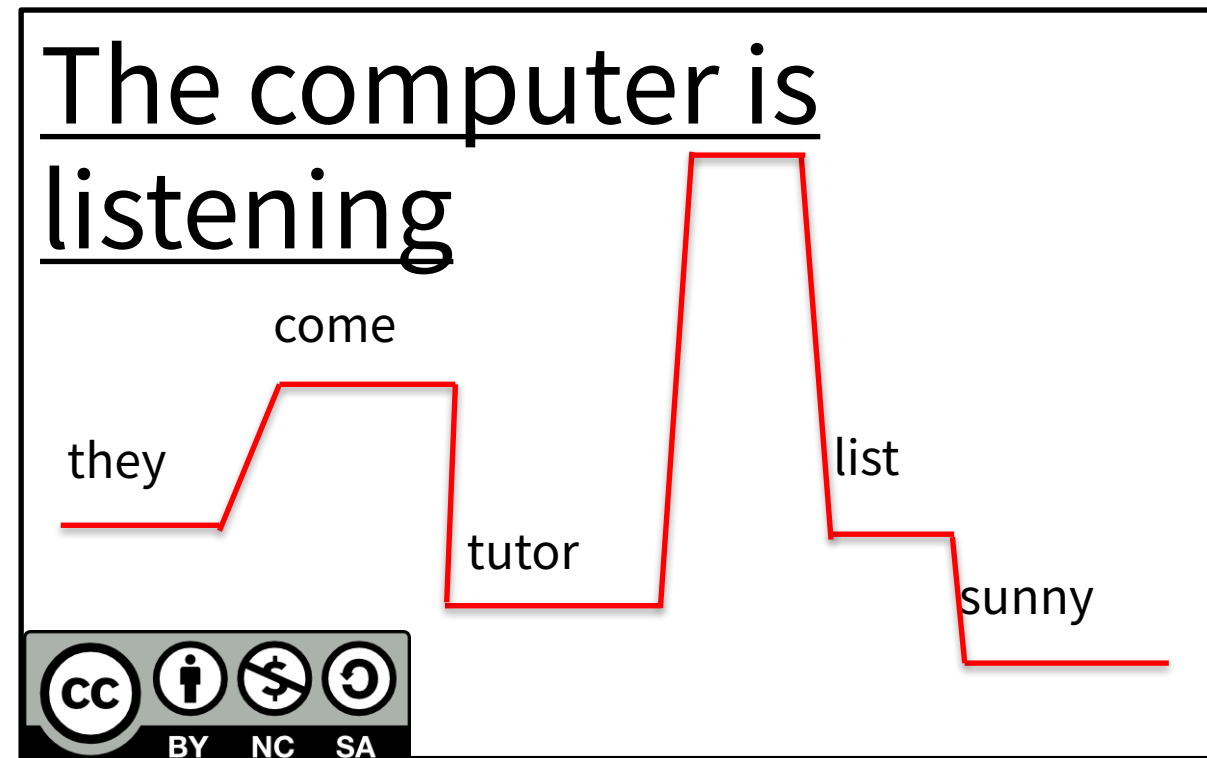
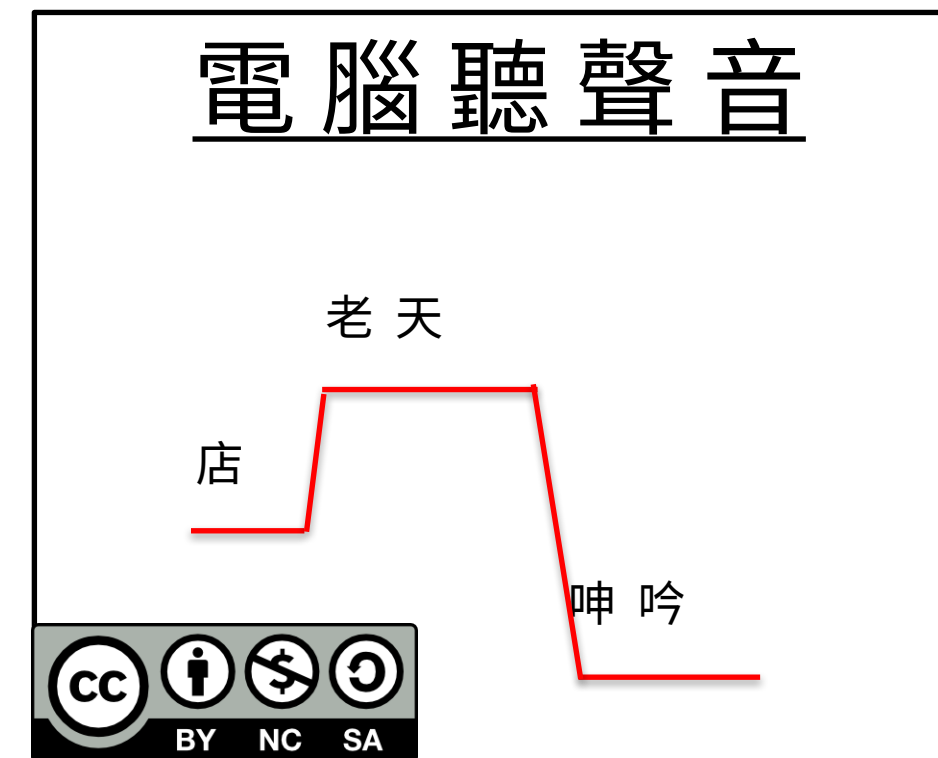


Fig. 2



Prob [the computer is listening] > Prob [they come tutor is list sunny]

Prob [電腦聽聲音] > Prob [店老天呻吟]

• Examples for Languages

$$0 \leq H(S) \leq \log M$$

– Source of English text generation

S ——— this course is about speech.....

- the random variable is the character $\Rightarrow 26 \times 2 + \dots < 64 = 2^6$
 $H(S) < 6$ bits (of information) per character
- the random variable is the word \Rightarrow assume total number of words = 30,000 $< 2^{15}$
 $H(S) < 15$ bits (of information) per word

– Source of speech for Mandarin Chinese

這一門課有關語音

- the random variable is the syllable (including the tone) $\Rightarrow 1300 < 2^{11}$
 $H(S) < 11$ bits (of information) per syllable (including the tone)
- the random variable is the syllable (ignoring the tone) $\Rightarrow 400 < 2^9$
 $H(S) < 9$ bits (of information) per syllable (ignoring the tone)
- the random variable is the character $\Rightarrow 8,000 < 2^{13}$
 $H(S) < 13$ bits (of information) per character

– Comparison: speech — 語音, girl — 女孩, computer — 計算機

Entropy and Perplexity

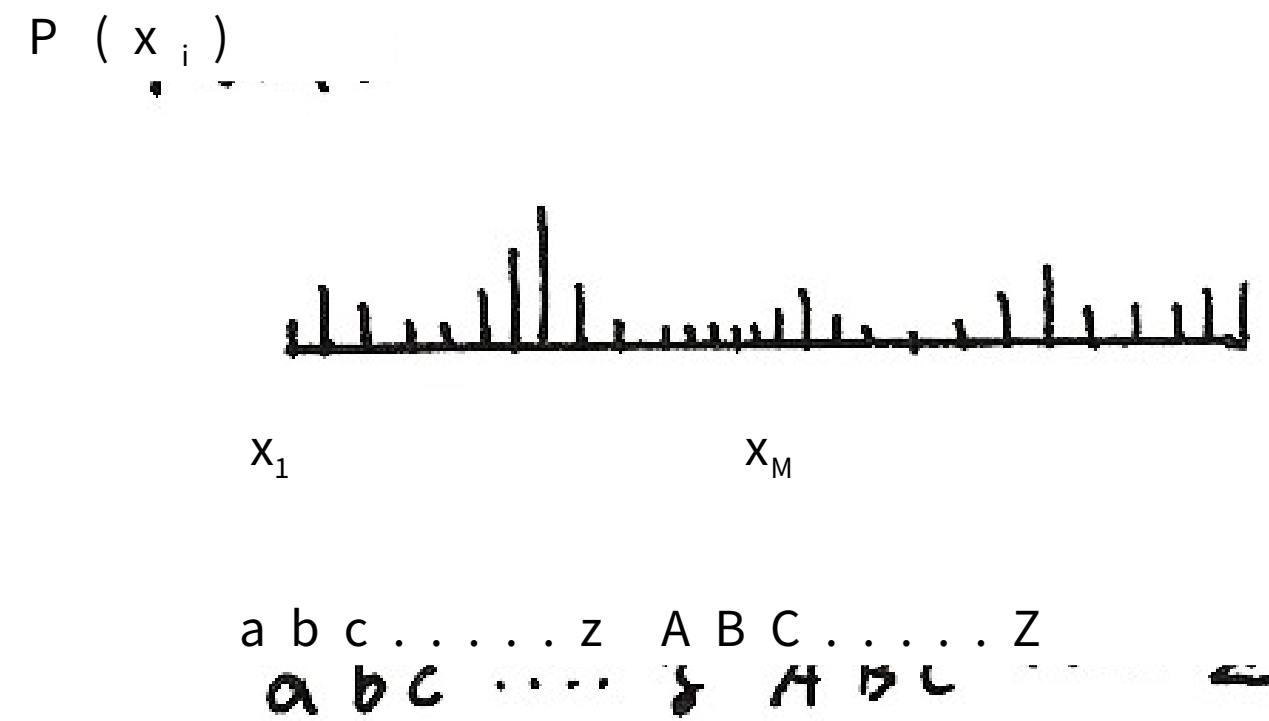
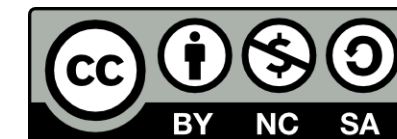
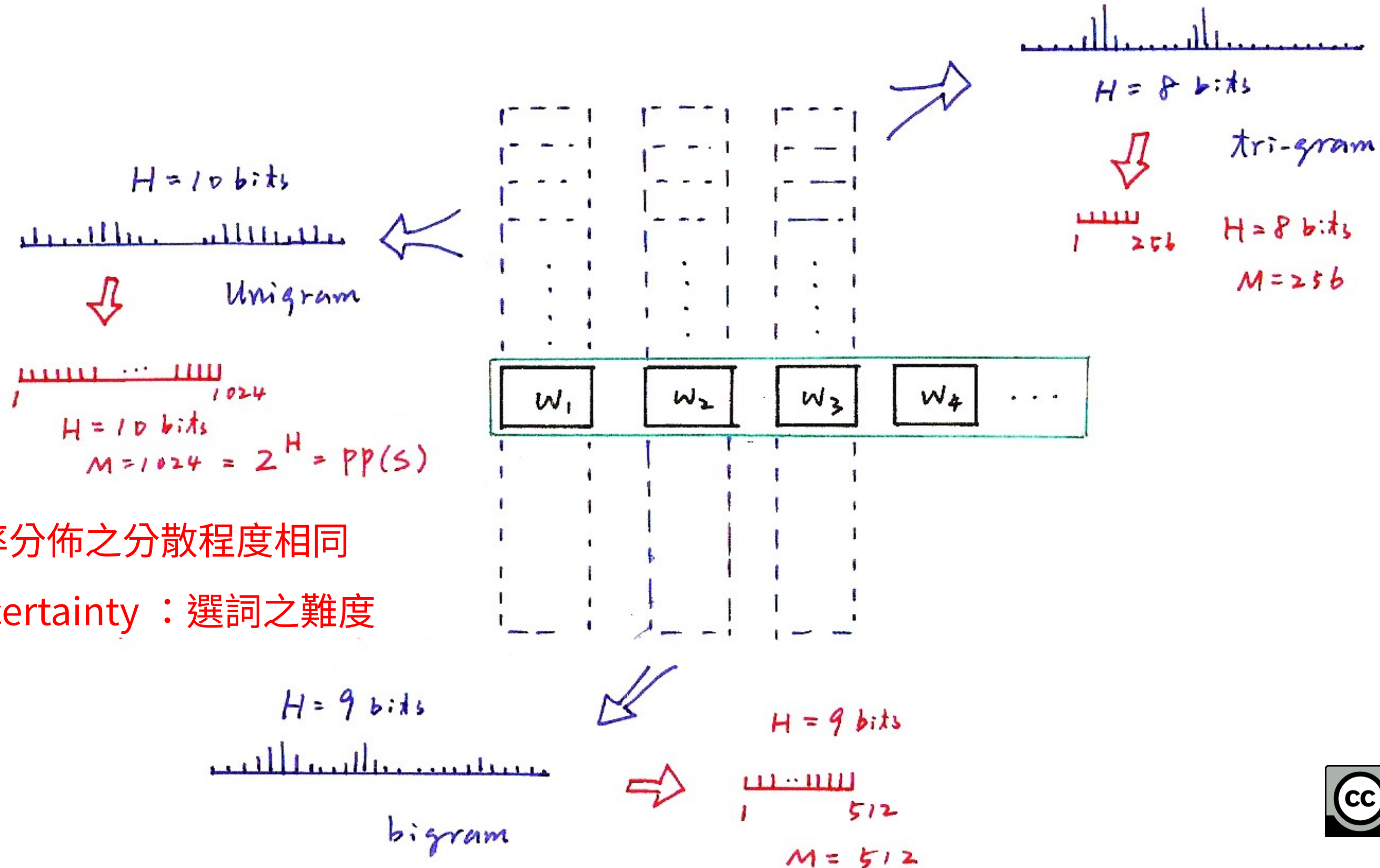


Fig. 3

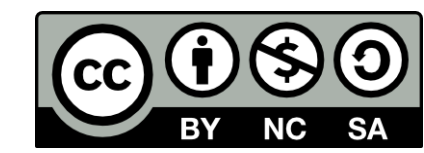


Entropy and Perplexity



機率分佈之分散程度相同
Uncertainty : 選詞之難度

Fig. 4



- **Perplexity of A Language Source S**

$$H(S) = - \sum_i p(x_i) \log_2 p(x_i)$$

(perplexity: 混淆度)

$$PP(S) = 2^{H(S)}$$

–size of a “virtual vocabulary” in which all words (or units) are equally probable

- e.g. 1024 words each with probability $\frac{1}{1024}$, =10 bits (of information)

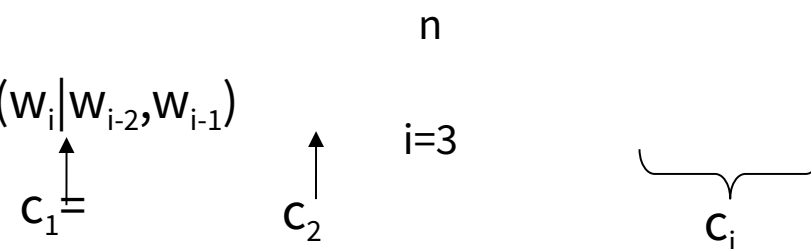
$$H(S) = 10 \text{ bits (of information)}, PP(S) = 1024$$

–branching factor estimate for the language

- **A Language Model**

–assigning a probability $P(w_i | c_i)$ for the next possible word w_i given a condition c_i

$$\text{e.g. } P(W=w_1, w_2, w_3, w_4, \dots, w_n) = P(w_1) P(w_2 | w_1) \prod_{i=3}^n P(w_i | w_{i-2}, w_{i-1})$$



- **A Test Corpus D of N sentences, with the i-th sentence W_i has n_i words and total words N_D**

$$D = [W_1, W_2, \dots, W_N], \quad W_i = w_1, w_2, w_3, \dots, w_{n_i}$$

$$N_D = \sum_{i=1}^N n_i$$

- **Perplexity of A Language Model $P(w_i|c_i)$ with respect to a Test Corpus D**

$$-H(P; D) = -\frac{1}{N_D} \sum_{i=1}^N \left[\sum_{j=1}^{n_i} \log P(w_j|c_j) \right]$$

average of all $\log P(w_j|c_j)$ over the whole corpus D

$$= \sum_{i=1}^N \sum_{j=1}^{n_j} \log \left[P(w_j|c_j)^{\frac{1}{N_D}} \right]$$

logarithm of geometric mean of $P(w_j|c_j)$

$$-pp(P; D) = 2^{H(P; D)}$$

average branching factor (in the sense of geometrical mean of reciprocals)

e.g. $P(W=w_1w_2\dots w_n) = P(w_1) P(w_2|w_1) P(w_3|w_1, w_2) P(w_4|w_2, w_3) P(w_5|w_3, w_4) \dots$

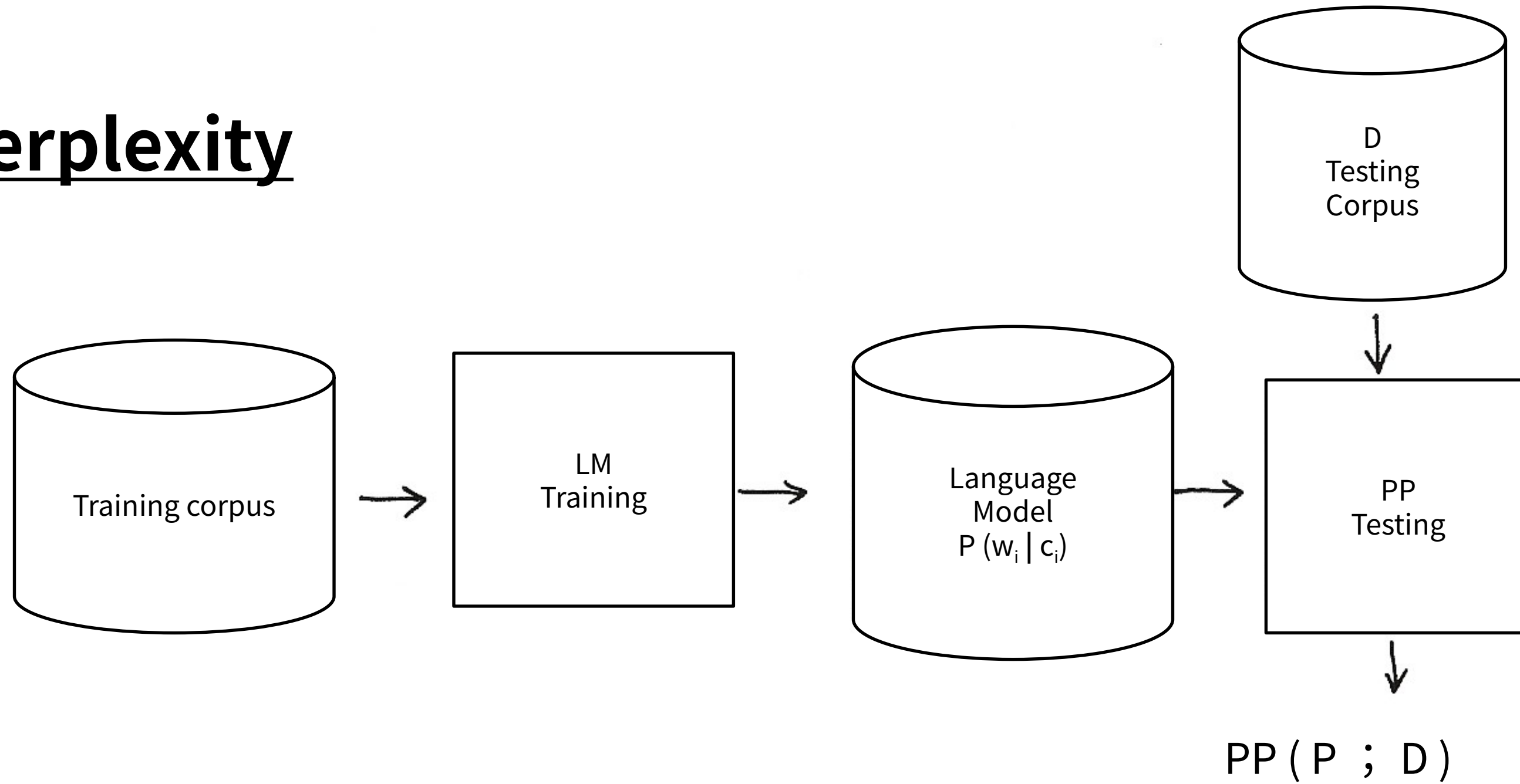
$$\begin{array}{ccccccccc} & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\ & & \frac{1}{1024} & & \frac{1}{512} & & \frac{1}{256} & & \frac{1}{128} & & \frac{1}{256} & & \frac{1}{256} \end{array}$$

$$\Rightarrow \left(\left[\left(\frac{1}{1024} \right) \left(\frac{1}{512} \right) \left(\frac{1}{256} \right) \left(\frac{1}{128} \right) \left(\frac{1}{256} \right) \dots \right]^{\frac{1}{n}} \right)^{-1} = 312$$

- the capabilities of the language model in predicting the next word given the linguistic constraints extracted from the training corpus

- the smaller the better, performance measure for a language

Perplexity



An Perplexity Analysis Example with Respect to Different Subject Domains

- Domain-specific Language Models Trained with Domain Specific Corpus of Much Smaller Size very often Perform Better than a General Domain Model

– Training corpus: Internet news in Chinese language

1	politics	19.6 M
2	congress	2.7 M
3	business	8.9 M
4	culture	4.3 M
5	sports	2.1 M
6	transportation	1.6 M
7	society	10.8 M
8	local	8.1 M
9	general(average)	58.1 M

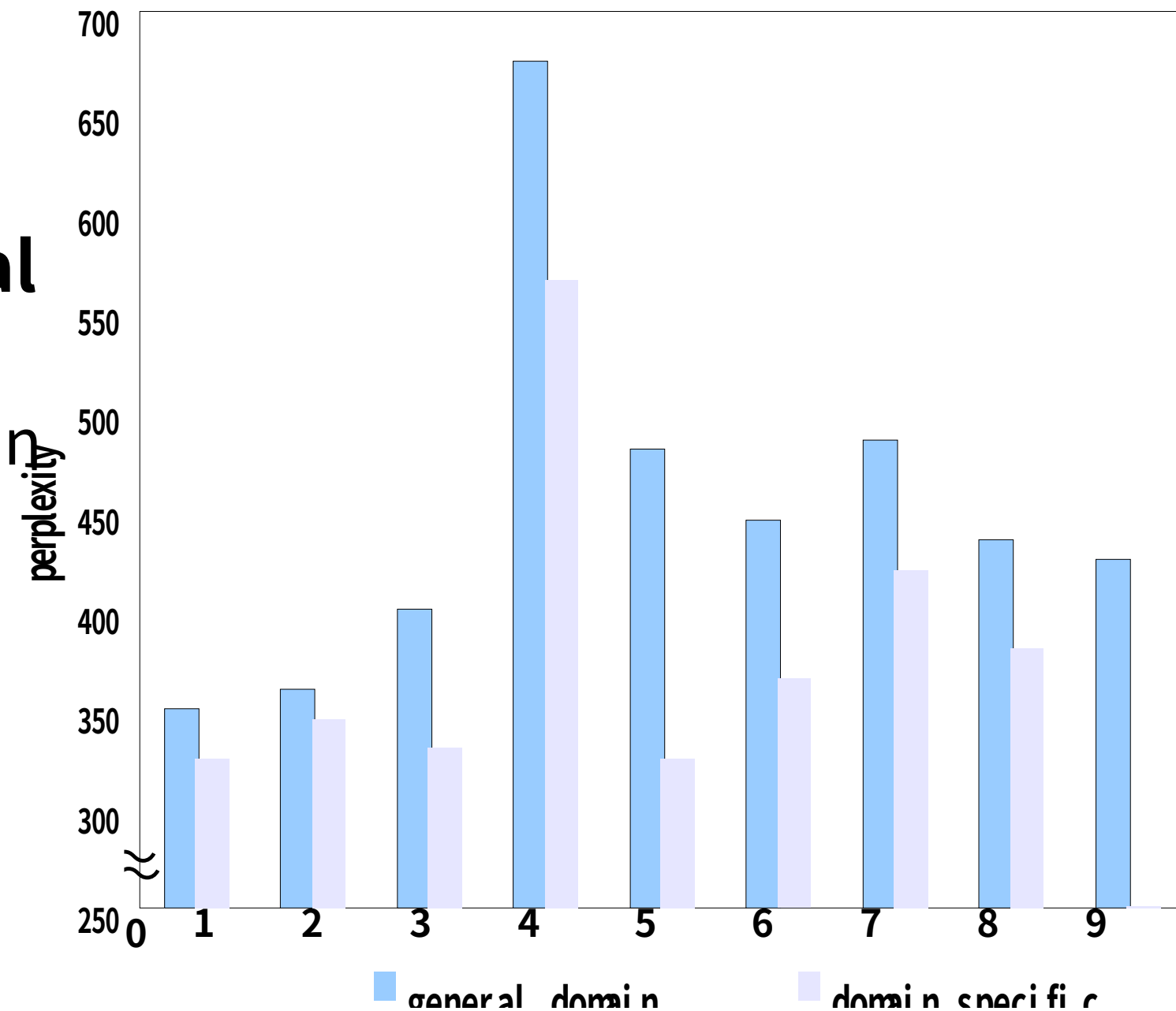


Fig. 5

– Sports section gives the lowest perplexity even with very small training corpus

- **KL Divergence or Cross-Entropy**

$$D[p(x)||q(x)] = \sum_i p(x_i) \log \left[\frac{p(x_i)}{q(x_i)} \right] \geq 0$$

– Jensen’s Inequality

$$- \sum_i p(x_i) \log p(x_i) \leq - \sum_i p(x_i) \log q(x_i)$$

– entropy when $p(x)$ is incorrectly estimated as $q(x)$ (leads to some entropy increase) Someone call this “cross-entropy” = $X[p(x) || q(x)]$

- **The True Probabilities $P(w_i|c_i)$ incorrectly estimated as $P(w_i|c_i)$ by the language model**

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=1}^N \log q(x_k) = \sum_i p(x_i) \log q(x_i)$$

(averaging by all samples) (averaging if $p(x_i)$ is law of large numbers)

- **The Perplexity is a kind “Cross-Entropy” when the true statistical characteristics of the test corpus D is incorrectly estimated as $p(w_i|c_i)$ by the language model**

- $H(P; D) = X(D || P)$
- the larger the worse

Law of Large Numbers

值	次數	
a_1	n_1	
a_2	n_2	
⋮	⋮	
+		
a_k	n_k	N

$$Ave = \frac{1}{N} \left(\sum_i a_i n_i \right) = \sum_i a_i \left(\frac{n_i}{N} \right) \equiv \sum_i a_i p_i$$

- **Data Sparseness**

- many events never occur in the training data

- e.g. Prob [Jason immediately stands up]=0 because Prob [immediately| Jason]=0

- smoothing: trying to assign some non-zero probabilities to all events even if they never occur in the training data

- **Add-one Smoothing**

- assuming all events occur once more than it actually does

- e.g. bigram

$$P(w^j | w^k) = \frac{N(\langle w^k, w^j \rangle)}{N(w^k)} = \frac{N(\langle w^k, w^j \rangle)}{\sum_j N(\langle w^k, w^j \rangle)} \Rightarrow \frac{N(\langle w^k, w^j \rangle) + 1}{\sum_j N(\langle w^k, w^j \rangle) + V}$$

- V: total number of distinct words in the vocabulary

Smoothing : Unseen Events –

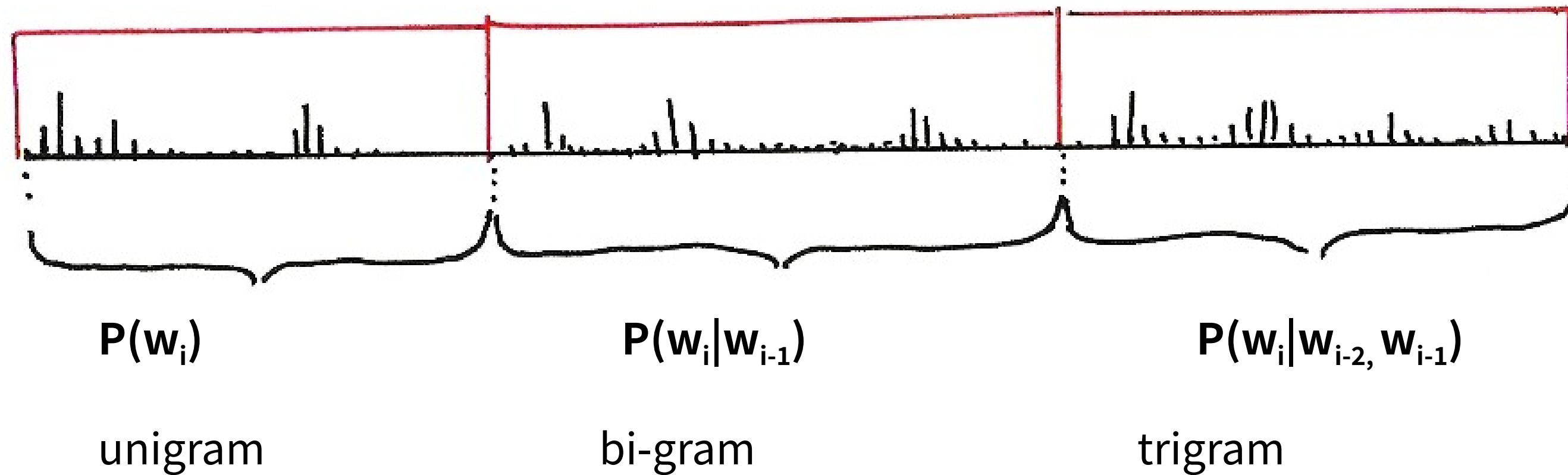
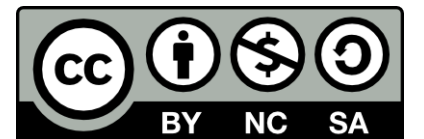


Fig. 6



• Back-off Smoothing

$$(w_i | w_{i-n+1}, w_{i-n+2}, \dots, w_{i-1}) = \begin{cases} P(w_i | w_{i-n+1}, w_{i-n+2}, \dots, w_{i-1}), & \text{if } N(\langle w_{i-n+1}, \dots, w_{i-1}, w_i \rangle) > 0 \\ a(w_{i-n+1}, \dots, w_{i-1}) (w_i | w_{i-n+2}, \dots, w_{i-1}), & \text{if } N(\langle w_{i-n+1}, \dots, w_{i-1}, w_i \rangle) = 0 \end{cases}$$

$$\left(\overline{\triangle}_{\triangle} = \begin{cases} \triangle_{\triangle} & , \text{ if } \triangle_{\triangle} > 0 \\ \overline{\triangle}_{\triangle-1} & , \text{ if } \triangle_{\triangle} = 0 \end{cases} \right) \begin{array}{l} \text{: n-gram} \\ \text{smoothed n-gram} \end{array}$$

– back-off to lower-order if the count is zero, prob (you| see) > prob (thou| see)

• Interpolation Smoothing

$$(w_i | w_{i-n+1}, w_{i-n+2}, \dots, w_{i-1}) = b(w_{i-n+1}, \dots, w_{i-1}) P(w_i | w_{i-n+1}, \dots, w_{i-1}) + (1 - b(w_{i-n+1}, \dots, w_{i-1})) (w_i | w_{i-n+2}, \dots, w_{i-1})$$

– interpolated with lower-order model even for events with non-zero counts

– also useful for smoothing a special domain language model with a background model, or adapting a general domain language model to a special domain

- **Good-Turing Smoothing**

- Good-Turning Estimates: properly decreasing relative frequencies for observed events and allocate some frequencies to unseen events

- Assuming a total of K events $\{1, 2, 3, \dots, k, \dots, K\}$
 number of observed occurrences for event k : $n(k)$,

$$N = \sum_{k=1}^K n(k)$$

N : total number of observations,

n_r : number of distinct events that occur r times (number of different events k such that $n(k) = r$)

- Good-Turing Estimates:
$$N = \sum_r r n_r$$
 - total counts assigned to unseen events = n_1
 - total occurrences for events having occurred r times: $r n_r \rightarrow (r+1)n_{r+1}$
 - an event occurring r times is assumed to have occurred r^* times,

$$r^* = (r+1) \frac{n_{r+1}}{n_r}$$

- r^* for

- $$\sum_r r^* n_r = \sum_r (r+1) \frac{n_{r+1}}{n_r} n_r = \sum_r (r+1) n_{r+1} = N$$

Good-Turing

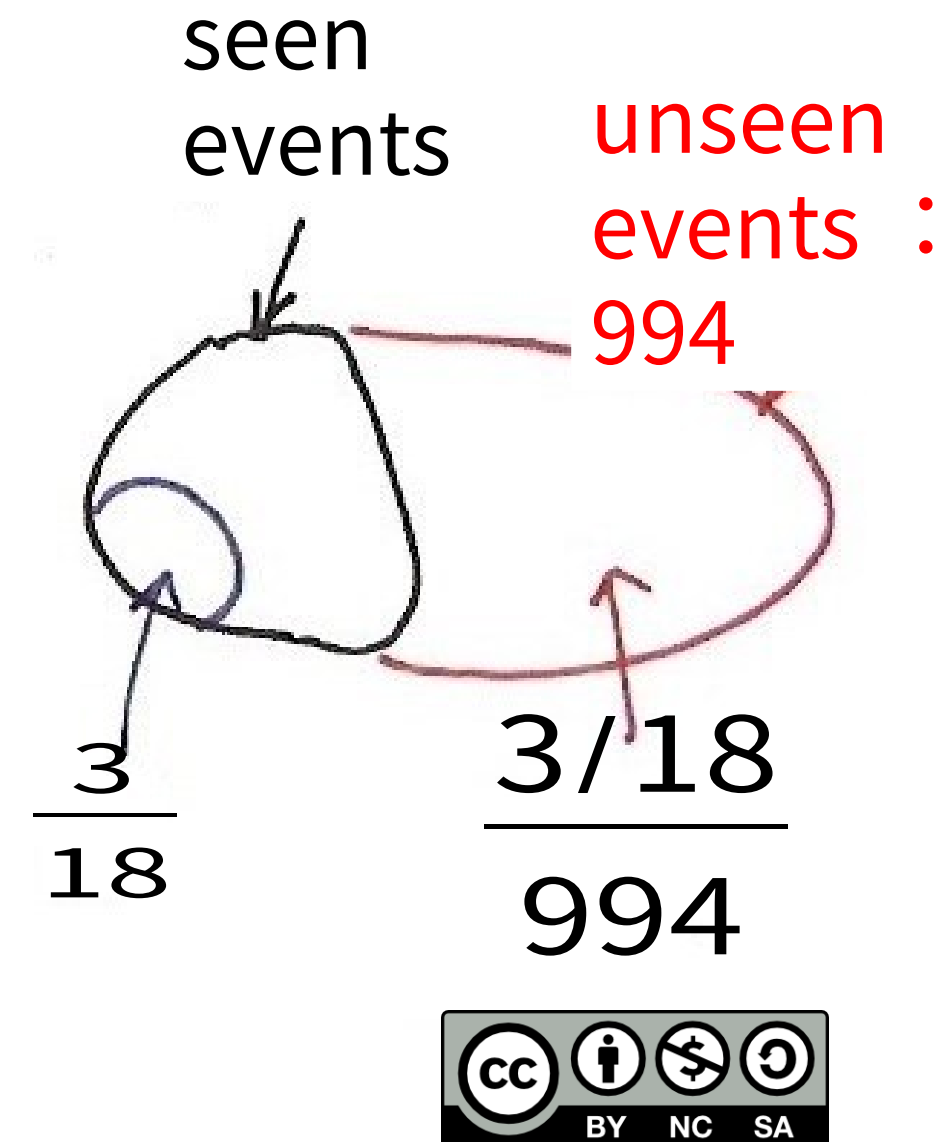
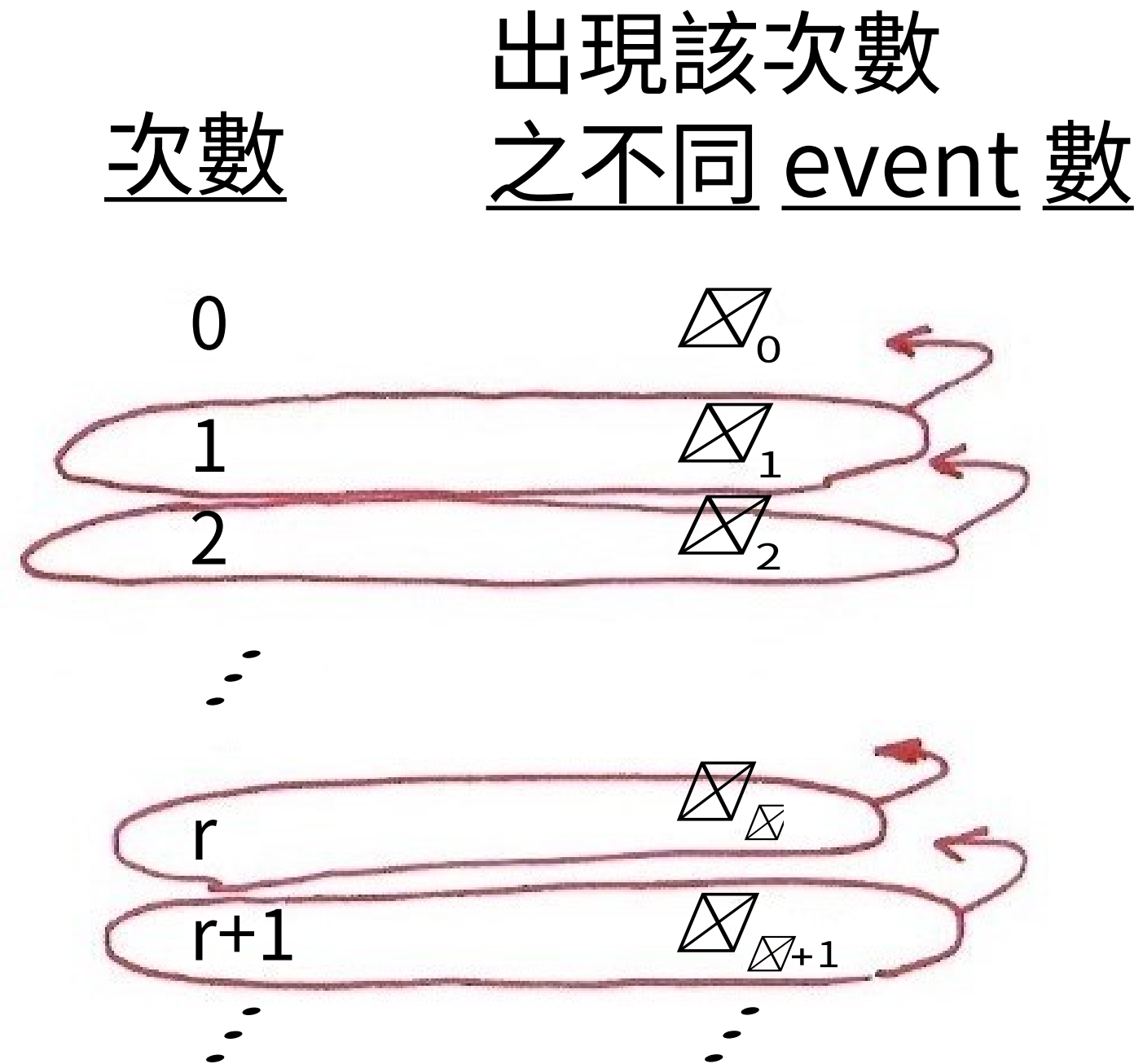
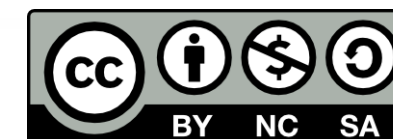


Fig. 7



- An analogy: during fishing, getting each kind of fish is an event
an example: $n(1)=10, n(2)=3, n(3)=2, n(4)=n(5)=n(6)=1, N=18$
prob (next fish got is of a new kind) = prob (those occurring only once)

- **Katz Smoothing**

- large counts are reliable, so unchanged
- small counts are discounted, with total reduced counts assigned to unseen events, based on Good-Turing estimates

$$\sum_{r=1}^{r_0} n_r (1 - d_r) = n_1 \quad d_r: \text{discount ratio for events with } r \text{ times}$$

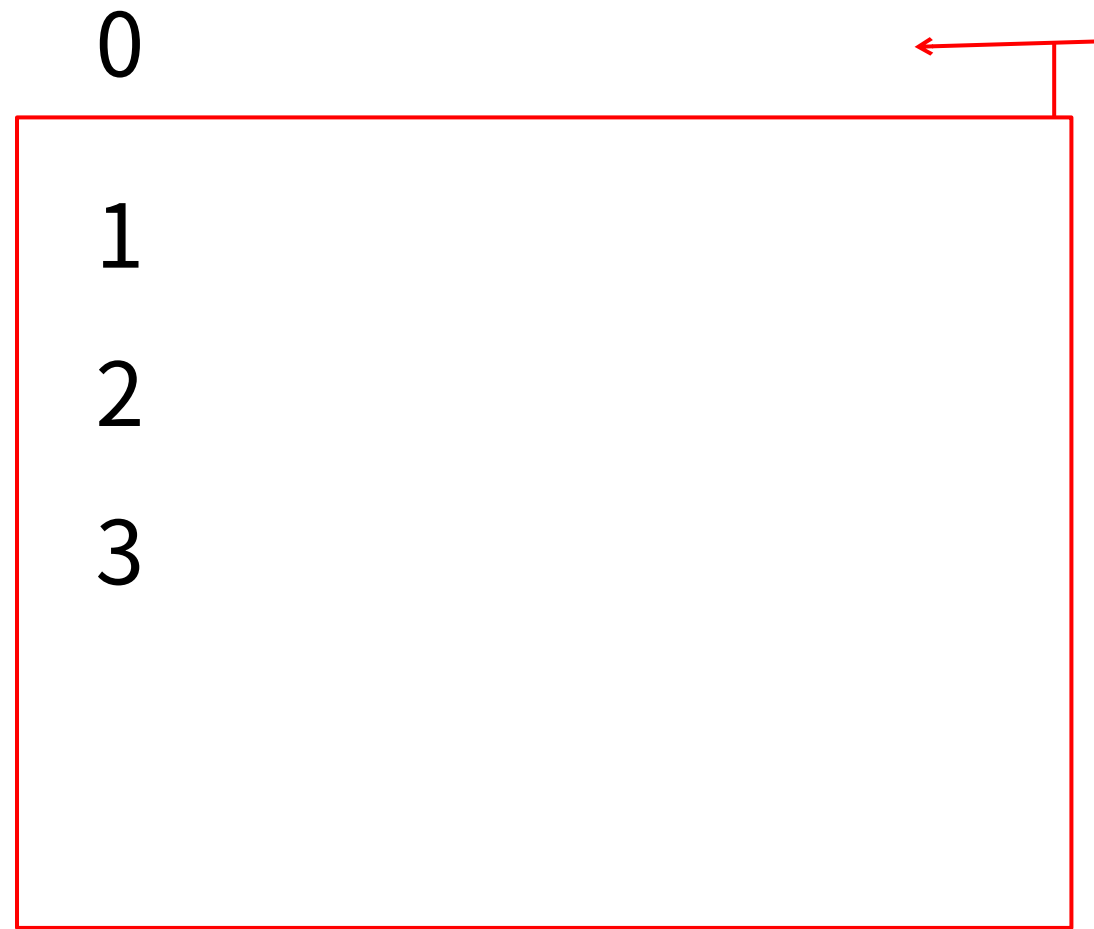
- distribution of counts among unseen events based on next-lower-order model: back off
- an example for bigram:

$$\bar{P}(w_i | w_{i-1}) = \begin{cases} N(\langle w_{i-1}, w_i \rangle) / N(w_i) & , r > r_0 \\ d_r \cdot N(\langle w_{i-1}, w_i \rangle) / N(w_i) & , r_0 \geq r > 0 \\ a(w_{i-1}, w_i) P(w_i) & , r = 0 \end{cases}$$

$a(w_{i-1}, w_i)$: such that the total counts equal to those assigned

Katz Smoothing

次數 不同 event 數

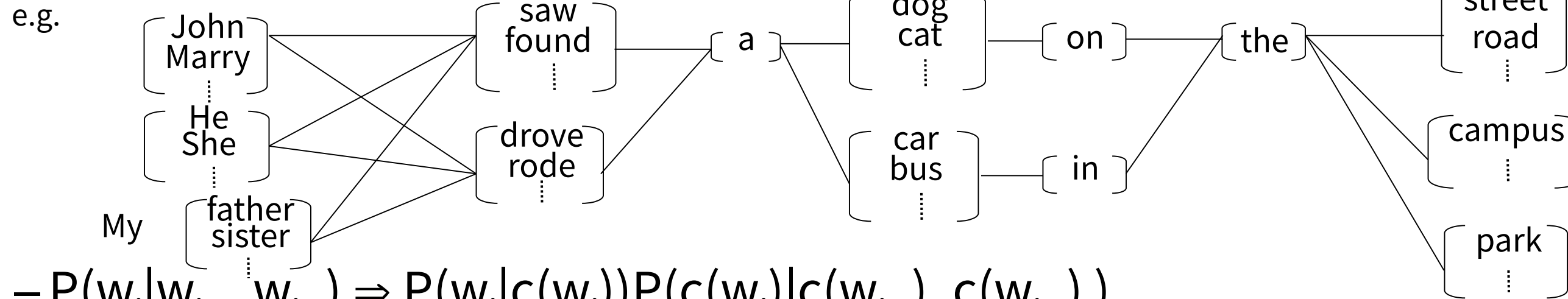


$$x_1 = \sum_{x=1}^{x_0} x_{x-1} (1 - x_{x-1}) x$$

$$x_{x-1} \propto \frac{x^*}{x}$$

unchanged

- **Clustering Words with Similar Semantic/Grammatical Behavior into Classes**

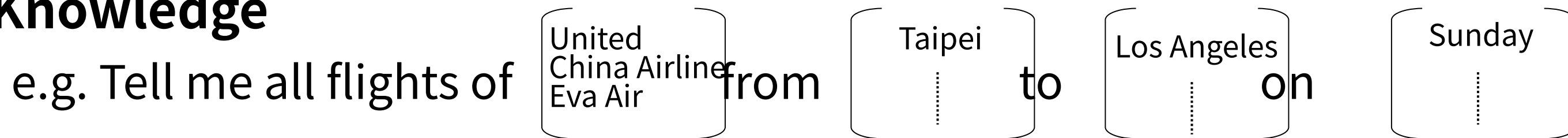


$$- P(w_i | w_{i-2}, w_{i-1}) \Rightarrow P(w_i | c(w_i)) P(c(w_i) | c(w_{i-2}), c(w_{i-1}))$$

$c(w_j)$: the class including w_j

- Smoothing effect: back-off to classes when too few counts, classes complementing the lower order models
- parameter size reduced

- **Limited Domain Applications: Rule-based Clustering by Human Knowledge**



- new items can be easily added without training data

- **General Domain Applications: Data-driven Clustering (probably aided by rule-based knowledge)**

• Data-driven Word Clustering Algorithm Examples

– Example 1:

- initially each word belongs to a different cluster
- in each iteration a pair of clusters was identified and merged into a cluster which minimizes the overall perplexity
- stops when no further (significant) reduction in perplexity can be achieved

Reference: “Cluster-based N-gram Models of Natural Language”,
Computational Linguistics, 1992 (4), pp. 467-479

– Example 2:

$$\text{Prob}[W = w_1 w_2 w_3 \dots w_n] = \prod_{i=1}^n \text{Prob}(w_i | w_1, w_2, \dots, w_{i-1}) = \prod_{i=1}^n \text{Prob}(w_i | h_i)$$

$h_i: w_1, w_2, \dots, w_{i-1}$, history of w_i

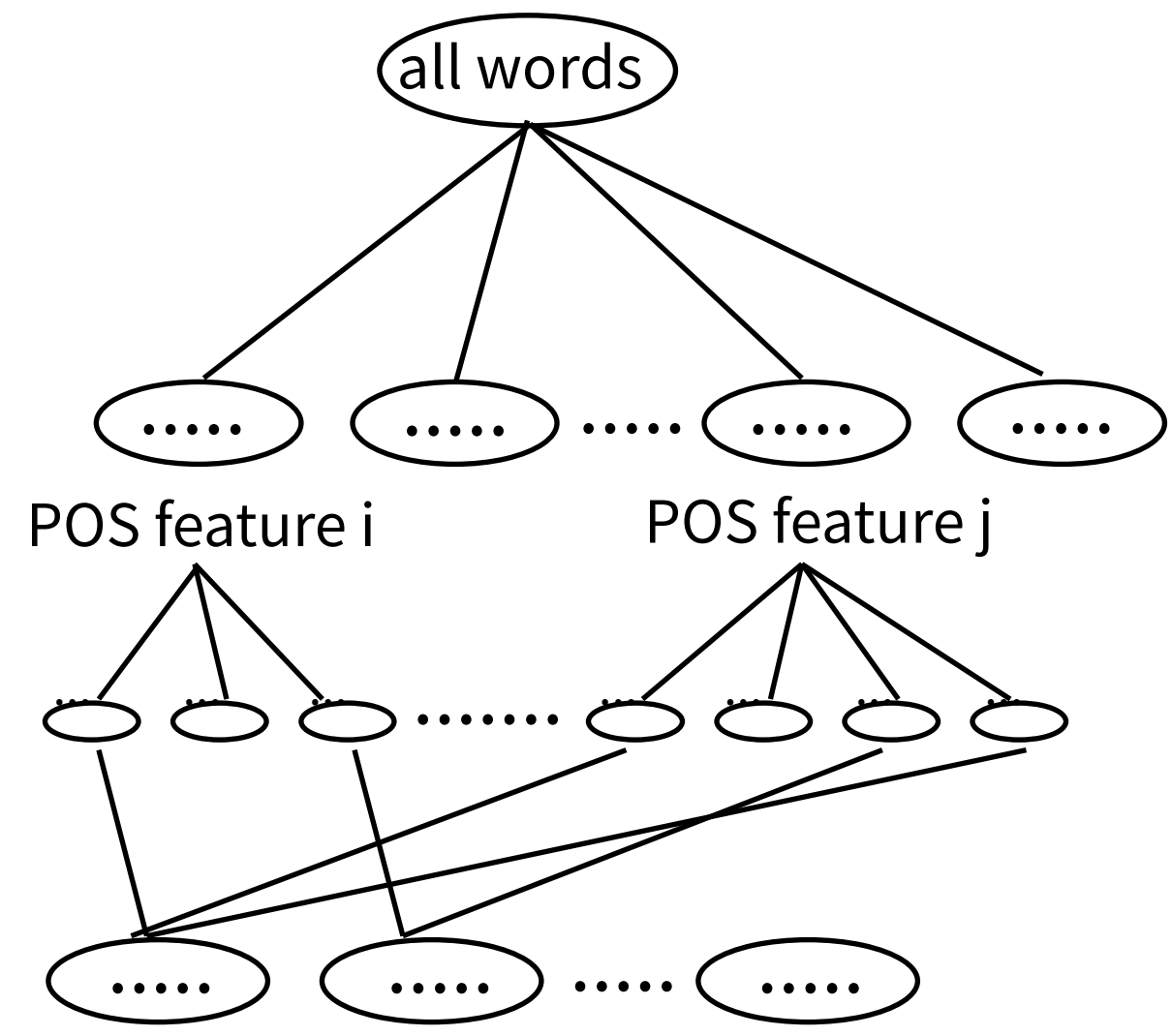
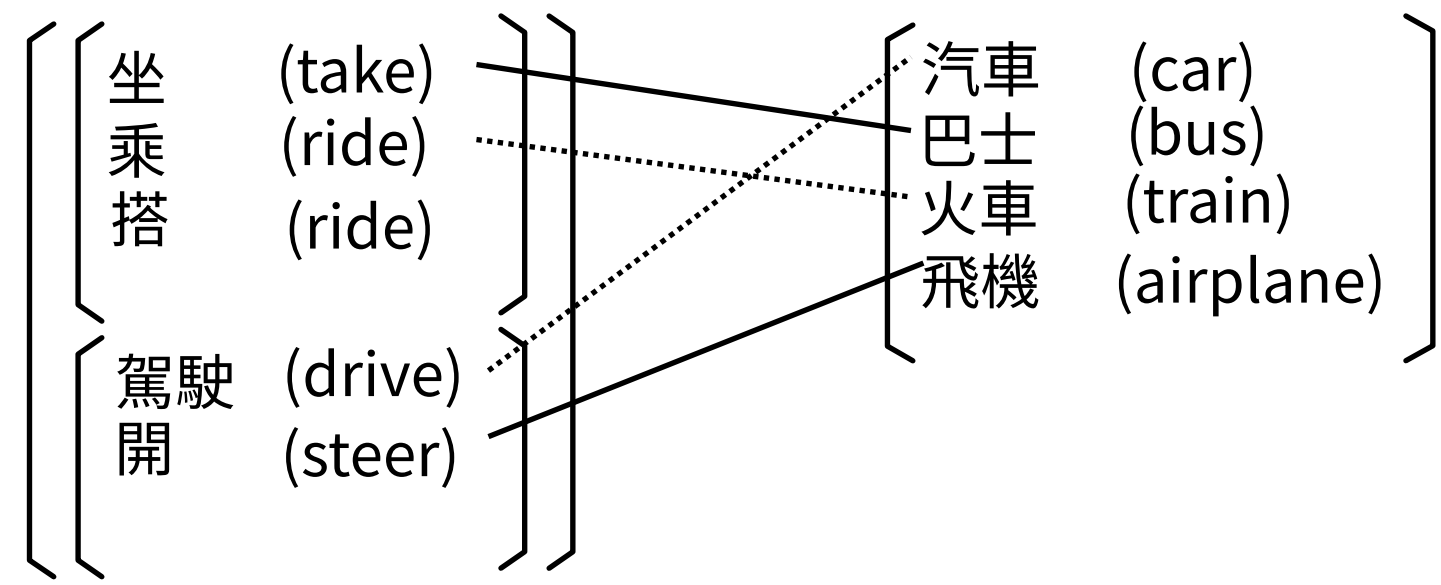
- clustering the histories into classes by decision trees (CART)
- developing a question set, entropy as a criterion
- may include both grammatic and statistical knowledge, both local and long-distance relationship

Reference: “A Tree-based Statistical Language Model for Natural Language Speech Recognition”, IEEE Trans. Acoustics, Speech and Signal Processing, 1989, 37 (7), pp. 1001-1008

An Example Class-based Chinese Language Model

• A Three-stage Hierarchical Word Classification Algorithm

- stage 1 : classification by 198
POS features (syntactic & semantic)
 - *each word belonging to one class only*
 - *each class characterized by a set of POS's*
- stage 2 : further classification with data-driven approaches
- stage 3 : final merging with data-driven approaches



- rarely used words classified by human knowledge
- both data-driven and human-knowledge-driven

POS features

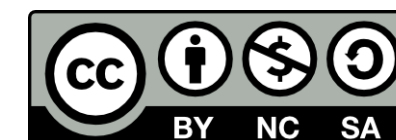
組織

($\omega_1, \omega_2, \omega_3, \dots$)

Data-driven Approach Example

	ω_1	ω_2	ω_3	ω_N
ω_1						
ω_2	..	58	164	..
ω_3						
...						
...	...	79	251	..
...						
ω_N

Fig. 8



- **Almost Each Character with Its Own Meaning, thus Playing Some Linguistic Role Independently**
- **No Natural Word Boundaries in a Chinese Sentence**

電腦科技的進步改變了人類的生活和工作方式

- word segmentation not unique
- words not well defined
- commonly accepted lexicon not existing

- **Open (Essentially Unlimited) Vocabulary with Flexible Wording Structure**

- new words easily created everyday 電 (electricity)+ 腦 (brain)→ 電腦 (computer)
- long word arbitrarily abbreviated 臺灣大學 (Taiwan University) → 臺大
- name/title 李登輝前總統 (former President T.H. Lee) → 李前總統登輝
- unlimited number of compound words 高 (high)+ 速 (speed)+ 公路 (highway)→ 高速公路 (freeway)

- **Difficult for Word-based Approaches Popularly Used in Alphabetic Languages**

- serious out-of-vocabulary(OOV) problem

~~Word-based and Class-based Language Modeling~~

- words are the primary building blocks of sentences
 - more information may be added
 - lexicon plays the key role
 - flexible wording structure makes it difficult to have a good enough lexicon
 - accurate word segmentation needed for training corpus
 - serious “out-of-vocabulary(OOV)” problem in many cases
 - all characters included as “mono-character words”
- **Character-based Language Modeling**
 - avoiding the difficult problem of flexible wording structure and undefined word boundaries
 - relatively weak without word-level information
 - higher order N-gram needed for good performance, which is relatively difficult to realize
 - **Integration of Class-based/Word-based/Character-based Models**
 - word-based models are more precise for frequently used words
 - back-off to class-based models for events with inadequate counts
 - each single word is a class if frequent enough

Segment Pattern Lexicon for Chinese – An Example Approach

- **Segment Patterns Replacing the Words in the Lexicon**
 - segments of a few characters often appear together : one or a few words
 - regardless of the flexible wording structure
 - automatically extracted from the training corpus (or network information) statistically
 - including all important patterns by minimizing the perplexity
- **Advantages**
 - bypassing the problem that the word is not well-defined
 - new words or special phrases can be automatically included as long as they appear frequently in the corpus (or network information)
 - can construct multiple lexicons for different task domains as long as the corpora are given (or available via the network)

Example Segment Patterns Extracted from Network News Outside of A Standard Lexicon

- **Patterns with 2 Characters**

- 一套，他很，再往，在向，但從，苗市，記在
深表，這篇，單就，無權，開低，蜂炮，暫不

- **Patterns with 3 Characters**

- 今年初，反六輕，半年後，必要時，在七月
次微米，卻只有，副主委，第五次，陳水扁，開發中

- **Patterns with 4 Characters**

- 大受影響，交易價格，在現階段，省民政廳，專責警力
通盤檢討，造成不少，進行了解，暫停通話，擴大臨檢

Samples

•With Extracted Segment Pattern

交通部 考慮 禁止 民眾 開車 時
使用 大哥大
已 委由 逢甲大學 研究中
預計 六月底 完成
至於 實施 時程
因涉及 交通 處罰 條例 的修正
必須 經立法院 三讀通過
交通部 無法確定
交通部 官員表示
世界 各國對 應否 立法 禁止 民眾
開車 時 打 大哥大
意見 相當 分歧

•With A Standard Lexicon

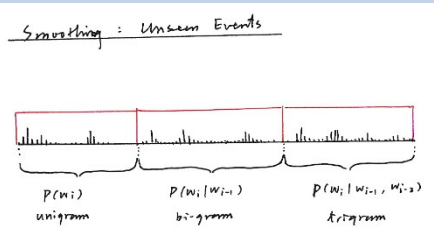
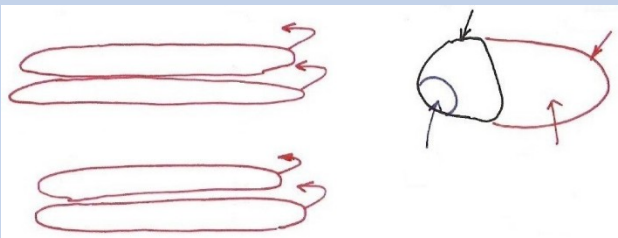

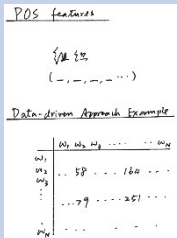

交通部 考慮 禁止 民眾 開車 時
使用 大哥大
已 委由 逢甲大學 研究中
預計 六月底 完成
至於 實施 時程
因涉及 交通 處罰 條例 的修正
必須 經立法院 三讀通過
交通部 無法確定
交通部 官員表示
世界 各國 對 應否 立法 禁止
民眾 開車 時 打 大哥大
意見 相當 分歧

•Percentage of Patterns outside of the Standard Lexicon :
28%

版權聲明

頁碼	作品	版權標示	作者／來源
2			國立台灣大學電機工程學系 李琳山 教授。本作品採用創用CC「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
4			國立台灣大學電機工程學系 李琳山 教授。本作品採用創用CC「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
5			國立台灣大學電機工程學系 李琳山 教授。本作品採用創用CC「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
9			國立台灣大學電機工程學系 李琳山 教授。本作品採用創用CC「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。

版權聲明

頁碼	作品	版權標示	作者／來源
13			國立台灣大學電機工程學系 李琳山 教授。本作品採用創用CC「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
16			國立台灣大學電機工程學系 李琳山 教授。本作品採用創用CC「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
22			國立台灣大學電機工程學系 李琳山 教授。本作品採用創用CC「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。