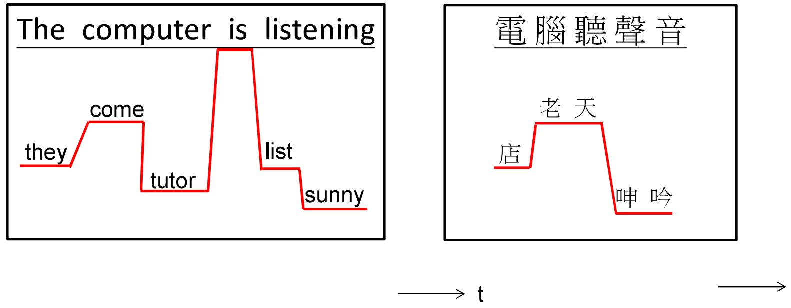
6.0 Language Modeling

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



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

Prob [電腦聽聲音] > Prob [店老天呻吟]

From Fundamentals of Information Theory

- **Examples for Languages**

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

  – Source of English text generation

  $\boxed{S}$ ⟶ this course is about speech.....

  - the random variable is the character ⇒ $26*2+.....<64=2^6$

    H (S) < 6 bits (of information) per character
  - the random variable is the word ⇒ assume total number of words=$30{,}000<2^{15}$

    H (S) < 15 bits (of information) per word

  – Source of speech for Mandarin Chinese

  $\boxed{S}$ ⟶ 這一門課有關語音.....

  - the random variable is the syllable (including the tone) ⇒ $1300 < 2^{11}$

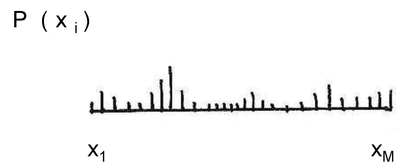
    H (S) < 11 bits (of information) per syllable (including the tone)
  - the random variable is the syllable (ignoring the tone) ⇒ $400 < 2^9$

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

    H (S) < 13 bits (of information) per character

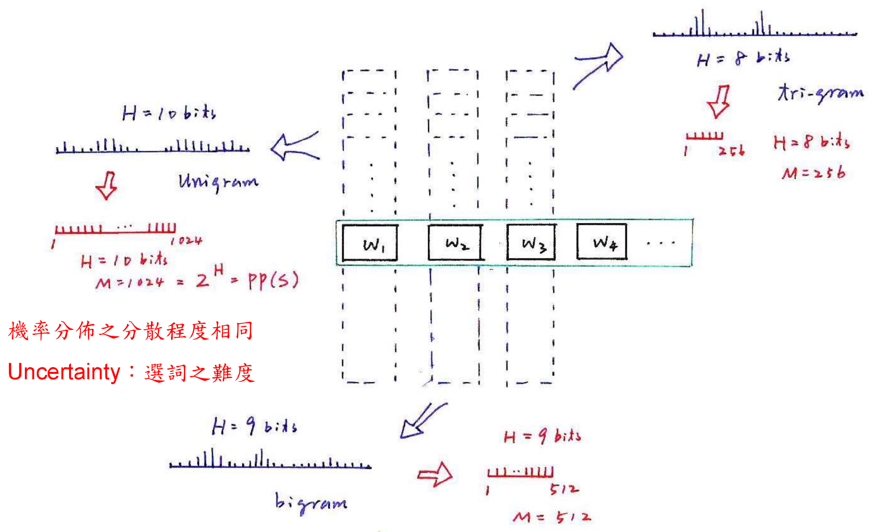
  – Comparison: speech— 語音, girl— 女孩, computer— 計算機

Entropy and Perplexity



a b c . . . . . z A B C . . . . . Z

Entropy and Perplexity



Perplexity

- **Perplexity of A Language Source S** (perplexity:混淆度)

$$H(S) = -\sum_i p(x_i)\log\left[p(x_i)\right]$$

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

  – size of a "virtual vocabulary" in which all words (or units) are equally probable
    - e.g. 1024 words each with probability $\frac{1}{1024}$, $I(x_i)$=10 bits (of information) H(S)= 10 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

$$\text{e.g. } P(W=w_1,w_2,w_3,w_4....w_n)=\underset{c_1=\phi}{P(w_1)}\underset{c_2}{P(w_2|w_1)}\prod_{i=3}^{n}P(w_i|\underbrace{w_{i-2},w_{i-1}}_{c_i})$$

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

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

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

Perplexity

- **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

$$= -\sum_{i=1}^{N}\sum_{j=1}^{n_i}\log\left[P(w_j|c_j)^{\frac{1}{N_D}}\right]$$, logarithm of geometric mean of $P(w_j|c_j)$

  – pp (P ; D) $=2^{H(P;D)}$

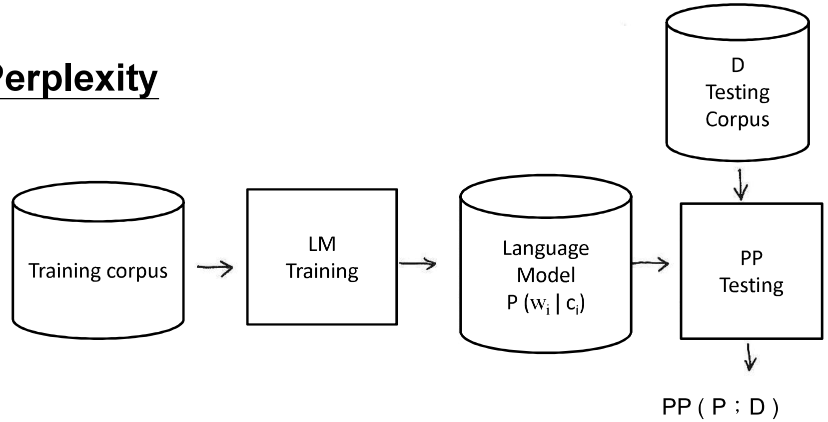
    average branching factor (in the sense of geometrical mean of reciprocals)

$$\text{e.g. } P(W=w_1w_2...w_n)=\underset{\frac{1}{1024}}{P(w_1)}\ \underset{\frac{1}{512}}{P(w_2|w_1)}\ \underset{\frac{1}{256}}{P(w_3|w_1,w_2)}\ \underset{\frac{1}{128}}{P(w_4|w_2,w_3)}\ \underset{\frac{1}{256}}{P(w_5|w_3,w_4)}\ .....$$

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

  – 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 model with respect to a test corpus
  – a function of a language model P and text corpus D

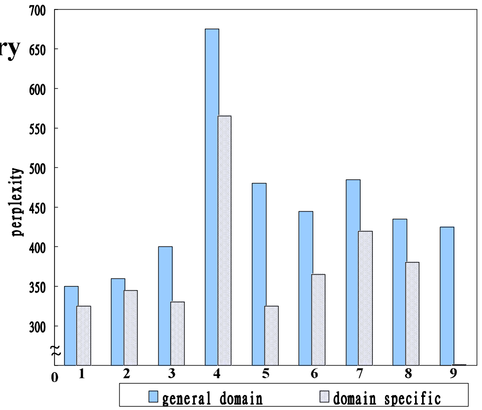
Perplexity



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



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

Perplexity

- **KL Divergence or Cross-Entropy**

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

  - Jensen's Inequality

$$-\sum_i p(x_i)\log\left[p(x_i)\right]\leq -\sum_i p(x_i)\log\left[q(x_i)\right]$$

Someone call this "cross-entropy" = X[p(x) || q(x)]

  - entropy when p(x) is incorrectly estimated as q(x) (leads to some entropy increase)

- **The True Probabilities $\boxed{\bar{P}(w_i|c_i)}$ incorrectly estimated as $\boxed{P(w_i|c_i)}$ by the language model**

$$\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}\log\left[q(x_k)\right]=\sum_i \boxed{p(x_i)}\log\left[\boxed{q(x_i)}\right]$$

(averaging by all samples) (averaging if $p(x_i)$ is known)

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

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

Smoothing of Language Models

- **Data Sparseness**
  - many events never occur in the training data

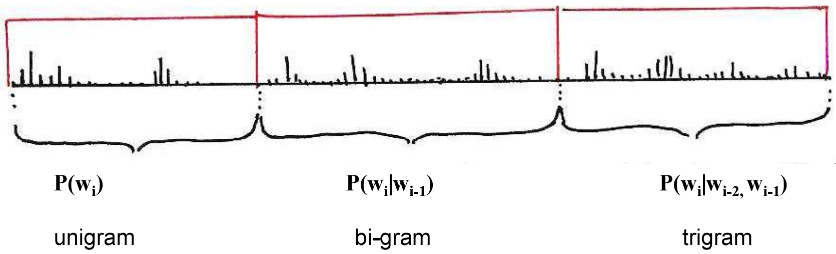
    e.g. Prob [Jason immediately stands up]=0 because Prob [immediately| Jason]=0
  - smoothing: trying to assign some non-zero probabilities to all events even if they never occur in the training data
- **Add-one Smoothing**
  - assuming all events occur once more than it actually does

    e.g. bigram

$$p(w^j|w^k)=\frac{N(<w^k,w^j>)}{N(w^k)}=\frac{N(<w^k,w^j>)}{\sum_j N(<w^k,w^j>)}\Rightarrow\frac{N(<w^k,w^j>)+1}{\sum_j N(<w^k,w^j>)+V}$$

V: total number of distinct words in the vocabulary

Smoothing : Unseen Events



Smoothing of Language Models

- **Back-off Smoothing**

$$\bar{P}(w_i|w_{i-n+1}, w_{i-n+2},\ldots w_{i-1})=\begin{cases} P(w_i|w_{i-n+1}, w_{i-n+2},\ldots w_{i-1}), \text{ if } N(<w_{i-n+1},\ldots w_{i-1}, w_i>)>0 \\ a(w_{i-n+1},\ldots w_{i-1})\ \bar{P}(w_i|w_{i-n+2},\ldots w_{i-1}), \text{ if } N(<w_{i-n+1},\ldots w_{i-1}, w_i>)=0 \end{cases}$$

$$\left(\bar{P}_n = \begin{cases} P_n & , \text{ if } P_n > 0 \\ a\bar{P}_{n-1} & , \text{ if } P_n = 0 \end{cases}\right)$$

P_n: n-gram
$\bar{P}_n$: smoothed n-gram

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

- **Interpolation Smoothing**

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

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

$$(\bar{P}_n = bP_n + (1-b)\bar{P}_{n-1})$$

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

$$P = bP_s + (1-b)P_b$$

Smoothing of Language Models

- **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...,k,.....K}
  number of observed occurrences for event k: n(k),

  N: total number of observations, $N = \sum_{k=1}^{K} n(k)$

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

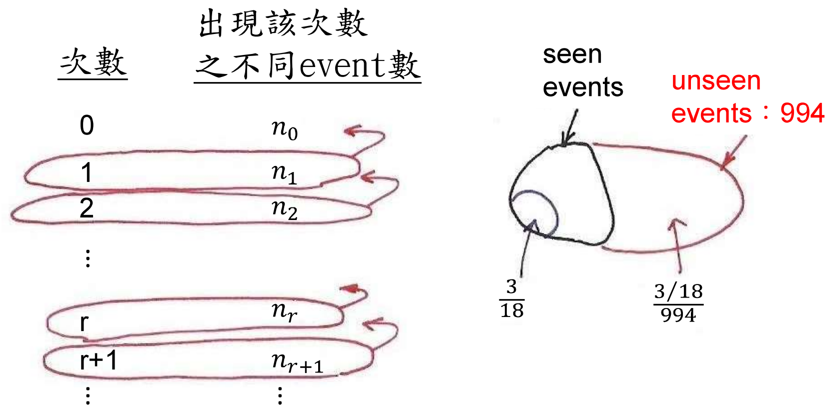
$$N = \sum_r r\, n_r$$

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

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

    - $r^* = \frac{n_1}{n_0}$ for r = 0
    - $\sum_r r^* n_r = \sum_r (r+1)\frac{n_{r+1}}{n_r} n_r = \sum_r (r+1)\, n_{r+1} = N$

Good-Turing



– 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) $=\frac{3}{18}$

Smoothing of Language Models

- **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

    $$\sum_{r=1}^{r_0} n_r (1-d_r)\, r = n_1 \quad , \quad d_r\text{: discount ratio for events with r times}$$
  - distribution of counts among unseen events based on next-lower-order model: back off
  - an example for bigram:

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

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

Katz Smoothing

次數	不同event數
0	n_0
1 $(1-d_1)$	n_1
2 $(1-d_2)$	n_2
3 $(1-d_3)$	n_3
⋮	⋮
r_0 $(1-d_{r_0})$	n_{r_0}
r_0+1	n_{r_0+1}
⋮	⋮
R_0	n_{R_0}

$$n_1 = \sum_{r=1}^{r_0} n_r (1-d_r)\, r$$

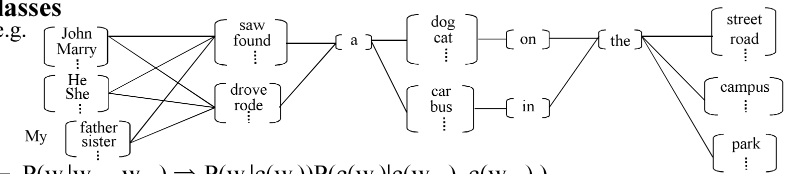
$$d_r \propto \frac{r^*}{r}$$

unchanged

Class-based Language Modeling

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

e.g.



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

    $c(w_j)$: 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**

  e.g. Tell me all flights of [United / China Airline / Eva Air / ⋮] from [Taipei / ⋮] to [Los Angeles / ⋮] on [Sunday / ⋮]

  - new items can be easily added without training data
- **General Domain Applications: Data-driven Clustering (probably aided by rule-based knowledge)**

Class-based Language Modeling

- **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:

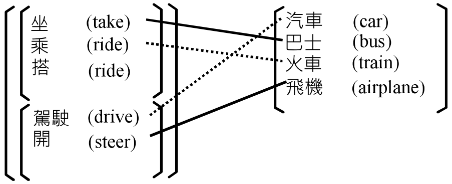
    $$\text{Prob}\,[W= w_1 w_2 w_3 \ldots w_n] = \prod_{i=1}^{n} \text{Prob}(w_i|w_1,w_2 \ldots w_{i-1}) = \prod_{i=1}^{n} \text{Prob}(w_i|h_i)$$

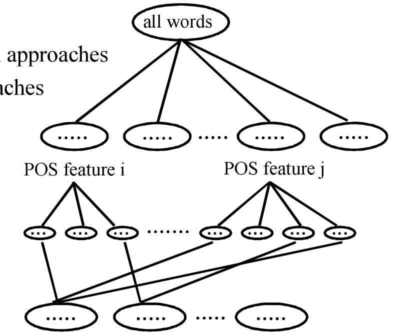
    h_i: $w_1,w_2,\ldots w_{i-1}$, history of w_i
    - clustering the histories into classes by decision trees (CART)
    - developing a question set, entropy as a criterion
    - may include both grammatic and statistical knowledge, both local and long-distance relationship

    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





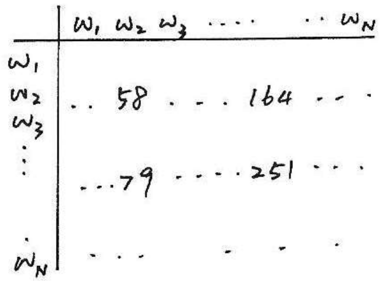
  - rarely used words classified by human knowledge
  - both data-driven and human-knowledge-driven

POS features

組織

(_ , _ , _ , _ . . .)

Data-driven Approach Example



Structural Features of Chinese Language

- **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 Character-based Chinese Language Models

- **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
  - character-based models offer flexibility for wording structure

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

Word/Segment Pattern Segmentation 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 分岐

• 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 分岐

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