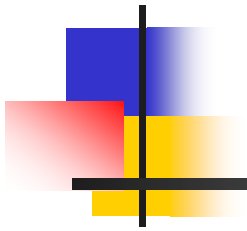
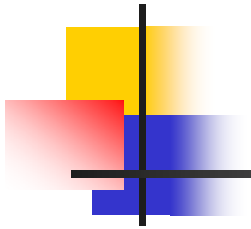


SRI-LM toolkit

2003/12/16

Louis Tsai
Speech Lab, CSIE, NTNU
louis@csie.ntnu.edu.tw



SRILM

- 由SRI International所開發出來的tool，用在Statistical Language Models
- http://www.speech.sri.com/projects/srilm
- 目前版本 Ver 1.3.3　2003/03

Create language model

$$\tilde{W} = \arg\max_{W} P(W \mid X) \cong \arg\max_{W} P(X \mid W) P(W)$$

- SRI-LM toolkit :
  - Step 1. Calculate ***ngram count*** from ***training data***
  - Step 2. Create ***LM*** from ***ngram count***

training data
(ssp.train)

已斷好詞的文件

關心 孩子 學業
可是 又 苦 無時 間 爲 孩子 挑選 參考書 的 家長 們
從 今天 起
多 了 一個 輕鬆 的 選擇
只要 趁 工作 之餘 逛 到 巷 口 的
填 個 國小 參考書 預購 單
就 能 以 八 五 折 的 價格 在 限定 時間 內 取 貨 了
這項 服務 明年 還 將 擴展到 國中 參考書
由於 國小 教科書 版本 不同
參考書 也 相當 多
各 書店 進 書 狀況 和數 量 也 不 相同

Vocabulary
(Lexicon2003-72k.txt)

巴
八
扒
叭
…
巴巴
叭叭
八寶
八方
八德
八堵
吧女
…

共有 71695 個words

ngram count file (ssp.n3.count)

在整個training data中出現的次數

Unigram

Bigram

Trigram

```
想像得到          4
想像得到 的       3
想像得到 的 事    2
想像得到 的 </s>          1
想像得到 球員     1
想像得到 球員 可以        1
鳳凰      123
鳳凰 騄 4
鳳凰 騄 </s>      4
鳳凰 花 5
鳳凰 花 開        1
鳳凰 花 </s>      4
鳳凰 </s>         20
鳳凰 自行車       1
鳳凰 自行車 </s>          1
鳳凰 城 20
鳳凰 城 </s>      8
鳳凰 城 大學      1
```

```
鳳凰 城 的        1
鳳凰 城 警方      1
鳳凰 城 太陽      5
鳳凰 城 電        3
鳳凰 城 水星      1
鳳凰 機場         1
鳳凰 機場 供      1
鳳凰 一樣         1
鳳凰 一樣 快速    1
鳳凰 中隊         1
鳳凰 中隊 </s>    1
鳳凰 自然         1
鳳凰 自然 來      1
鳳凰 般 3
鳳凰 般 重        1
鳳凰 般 再度      2
鳳凰 科技         6
……
```

LM Format
(ssp.n3.kn.lm)

以\data\開頭

各種ngram的個數

\1-grams:是unigram的開頭

最高的ngram沒有backoff weight

```
\data\
ngram 1=71697
ngram 2=2795822
ngram 3=1700527

\1-grams:
-1.804081       </s>
-99     <s>     -0.5467538
-2.548589       一      -0.6464233
-3.989615       一一    -0.2506604
-6.685257       一一恐怖        -0.1400223
-6.685257       一一恐怖攻擊    -0.1128442
-6.075799       一丁點  -0.1827688
-4.105376       一九九  -1.257224
-5.808214       一了百了        -0.308177
-5.921716       一刀兩斷        -0.3545529
-3.940887       一下    -0.367599
...
-0.4833062      糊 裏 糊塗
-0.6593975      餐廳 裏 往
-0.6593975      館 裏 大家
-0.2190889      鎮 裏 不同
-0.4833062      鐵窗 裏 竄
-1.223669       <s> 恒 大
-1.223669       <s> 恒 生
-0.3077702      <s> 恒 春
-0.3159989      小型 恒 指
-0.4833062      屏東 恒 春
-0.2190889      范 恒 山
-0.4833062      從 恒 春
-0.4833062      邊 恒 雄
-0.6593975      上 粧 </s>
-0.2759993      化 粧 品

\end\
```

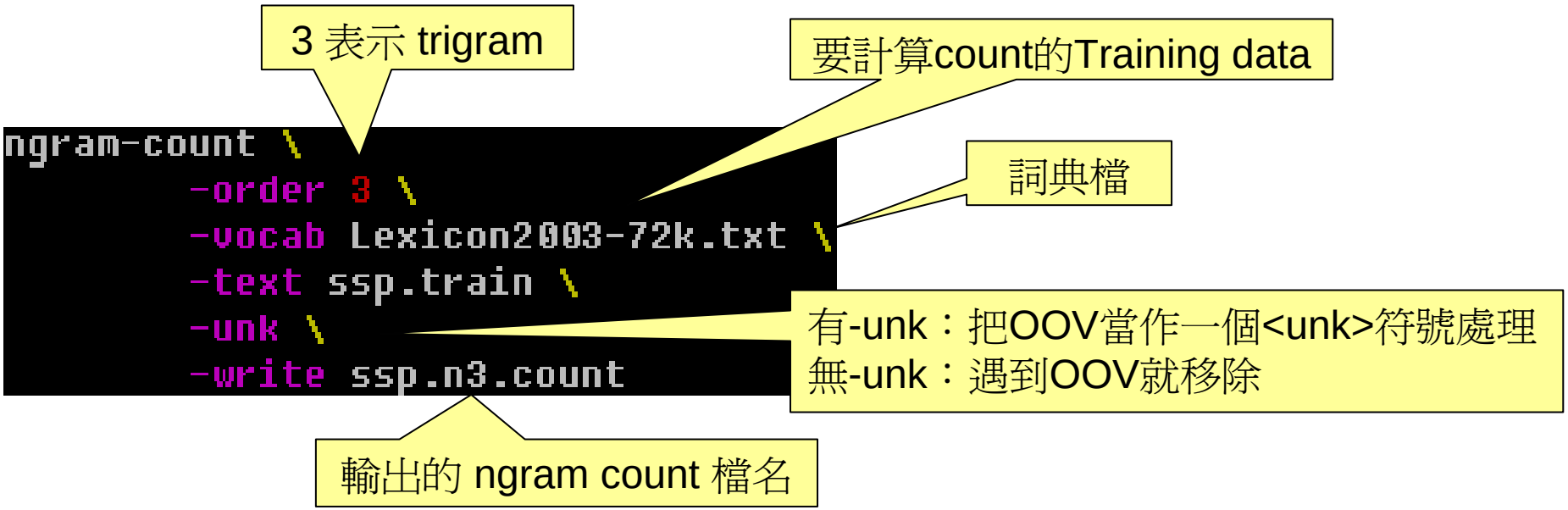
Log probability (Base 10)

Log of backoff weight (Base 10)

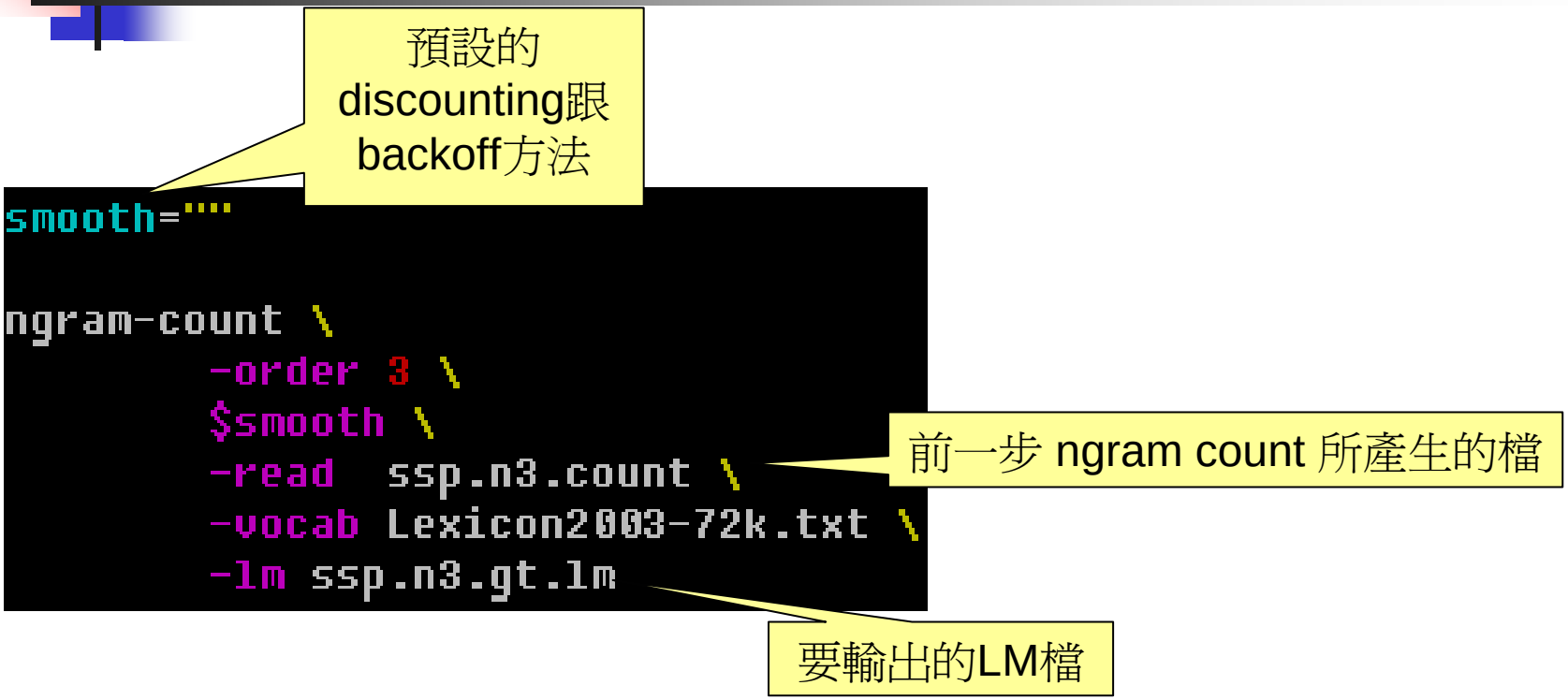
以\end\結尾

ngram-count

- 計算training data中每個出現的ngram的出現次數



Good-Turing Discounting and Katz Backoff



Compute perplexity

```
ngram \
        -lm ssp.n3.gt.lm \
        -ppl ssp.test
```

LM檔

要算 perplexity 的 test data

輸出範例：

```
file ssp.test: 89 sentences, 6979 words, 0 OOVs
0 zeroprobs, logprob= -21066.5 ppl= 956.182 ppl1= 1043.64
```

$10^{-\frac{\text{logprob}}{\text{句數+字數}}}$

$10^{-\frac{\text{logprob}}{\text{字數}}}$

Good-Turing Discounting Parameters

$$P_{Katz}(w_i|w_{i-1}) = \begin{cases} C(w_{i-1}w_i)/C(w_{i-1}) & r > k \\ d_r C(w_{i-1}w_i)/C(w_{i-1}) & k \geq r > 0 \\ \alpha(w_{i-1})P(w_i) & r = 0 \end{cases}$$

maxcount

mincount

$$\text{where } d_r = \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}} \quad \text{and} \quad \alpha(w_{i-1}) = \frac{1 - \sum_{w_i: r>0} P_{Katz}(w_i|w_{i-1})}{1 - \sum_{w_i: r>0} P_{Katz}(w_i)}$$

自行指定mincount跟maxcount :

```
smooth=""

ngram-count \
        -order 3 \
        $smooth \
        -gt1min 3 -gt1max 7 -gt2min 3 -gt2max 7 -gt3min 3 -gt3max 7 \
        -read  ssp.n3.count \
        -vocab Lexicon2003-72k.txt \
        -lm ssp.n3.gt.lm
```

Modified Kneser-Ney Discounting

```
smooth="-kndiscount1 -kndiscount2 -kndiscount3"

ngram-count \
        -order 3 \
        $smooth \
        -read  ssp.n3.count \
        -vocab Lexicon2003-72k.txt \
        -lm ssp.n3.kn.lm
```

- Take the conditional probabilities of bigrams (*m*-gram, *m*=2) for example:

$$0 \le D \le 1$$

$$P_{KN}\left(w_i \middle| w_{i-1}\right) = \begin{cases} \dfrac{\max\left\{C\left[w_{i-1}, w_i\right] - D,\ 0\right\}}{C\left[w_{i-1}\right]} & \text{if } C\left[w_{i-1}, w_i\right] > 0 \\ \alpha\left(w_{i-1}\right) P_{KN}\left(w_i\right) & \text{otherwise} \end{cases}$$

1. $P_{KN}\left(w_i\right) = \boldsymbol{C}\left[\bullet\, w_i\right] / \sum_{w_j} \boldsymbol{C}\left[\bullet\, w_j\right]$,

$\boldsymbol{C}\left[\bullet\, w_i\right]$ is the unique words preceding w_i

2. normalizing constant

$$\alpha\left(w_{i-1}\right) = \frac{1 - \sum_{w_i : C\left[w_{i-1} w_i\right] > 0} \dfrac{\max\left\{C\left[w_{i-1} w_i\right] - D,\ 0\right\}}{C\left[w_{i-1}\right]}}{\sum_{w_i : C\left[w_{i-1} w_i\right] = 0} P_{KN}\left(w_i\right)}$$

Witten-Bell Discounting

```
smooth="-wbdiscount1 -wbdiscount2 -wbdiscount3"

ngram-count \
        -order 3 \
        $smooth \
        -read  ssp.n3.count \
        -vocab Lexicon2003-72k.txt \
        -lm ssp.n3.wb.lm
```

Witten-Bell Discounting

- **Key Concept—Things Seen Once:** Use the count of things you've seen once to help estimate the count of things you've never seen
- So we estimate the *total* probability mass of all the zero *N*-grams with the number of types divided by the number of tokens plus observed types:

$$\sum_{i:c_i=0} p_i^* = \frac{T}{N+T}$$

N : the number of tokens
T : observed types

Witten-Bell Discounting

- T/(N+T) gives the total "probability of unseen *N*-grams", we need to divide this up among all the zero *N*-grams
- We could just choose to divide it equally

$$Z = \sum_{i:c_i=0} 1$$

Z is the total number of *N*-grams with count zero

$$p_i^* = \frac{T}{Z(N+T)}$$

Witten-Bell Discounting

$$p_i^* = \frac{c_i}{N+T} \quad \text{if } (c_i > 0)$$

Alternatively, we can represent the smoothed counts directly as:

$$c_i^* = \begin{cases} \frac{T}{Z}\frac{N}{N+T}, & \text{if} \quad c_i = 0 \\ c_i \frac{N}{N+T}, & \text{if} \quad c_i > 0 \end{cases}$$

Witten-Bell Discounting

$$\frac{T}{Z(N+T)} \cdot Z + \sum_{i:c_i>0} \frac{c_i}{N+T}$$

$$= \frac{T}{N+T} + \frac{N}{N+T}$$

$$= 1$$

$$\frac{T}{Z}\frac{N}{N+T} \cdot Z + \sum_{i:c_i>0} c_i \frac{N}{N+T}$$

$$= \frac{NT}{N+T} + \frac{N^2}{N+T}$$

$$= \frac{N(N+T)}{N+T}$$

$$= N$$

Witten-Bell Discounting

- For bigram

$$\sum_{i:c(w_x w_i)=0} p^*(w_i \mid w_x) = \frac{T(w_x)}{N(w_x)+T(w_x)}$$

T: the number of bigram types, N: the number of bigram token

$$Z(w_x) = \sum_{i:c(w_x w_i)=0} 1$$

$$p^*(w_i \mid w_{i-1}) = \frac{T(w_{i-1})}{Z(w_{i-1})(N(w_{i-1})+T(w_{i-1}))} \quad \text{if}(c_{w_{i-1}w_i} = 0)$$

$$\sum_{i:c(w_x w_i)>0} p^*(w_i \mid w_x) = \frac{c(w_x w_i)}{c(w_x)+T(w_x)}$$

Absolute Discounting

```
smooth="-cdiscount1 0.5 -cdiscount2 0.5 -cdiscount3 0.5"

ngram-count \
        -order 3 \
        $smooth \
        -read  ssp.n3.count \
        -vocab Lexicon2003-72k.txt \
        -lm ssp.n3.abs.lm
```

Smoothing

absolute discounting

- Absolute discounting subtracting a fixed discount $D <= 1$ from each nonzero count

$$P_{abs}(w_i \mid w_{i-n+1} w_{i-1})$$
$$= \frac{C(w_{i-n+1} \ldots w_{i-1}) - D}{\sum_{w_i} C(w_{i-n+1} \ldots w_i)} + (1 - \lambda_{w_{i-n+1} \ldots w_{i-1}}) P_{abs}(w_i \mid w_{i-n+2} \ldots w_{i-1})$$

Ristad's Natural Discounting

```
smooth="-ndiscount1 -ndiscount2 -ndiscount3"

ngram-count \
        -order 3 \
        $smooth \
        -read  ssp.n3.count \
        -vocab Lexicon2003-72k.txt \
        -lm ssp.n3.nd.lm
```

Eric Ristad's Natural Law of Succession -- The discount factor d is identical for all counts,.

d = (n(n+1) + q(1- q)) / (n^2 + n + 2q)

where n is the total number events tokens, q is the number of observed event types. If q equals the vocabulary size no discounting is performed.