Lecture 5 : Sequence Tagging and Language Processing

楊立偉教授
台灣大學工管系

wyang@ntu.edu.tw

2017

So Far What We Have

- L1: Document Similarity
  - "a Bag of Words" Model, Term Weighting (TF-IDF, χ2, MI)
  - Vector Space Model (VSM)
- L2: Co-occurrence
  - Association
  - Link analysis : Co-citation & Coupling
- L3: Classification
  - Naïve Bayes, k Nearest Neighbors, Support Vector Machine
  - Decision Tree, Bagging and Boosting, Random Forest
  - Neural Networks

- L4: Clustering
  - k Means, DBSCAN, Hierarchical Agglomerative Clustering
  - Topic Modeling

Agenda

- Sequence tagging
- Chinese processing
- Other language issues

Sequence Tagging

- To assign of a label to each member of a sequence of observed values. For example:
  - part of speech tagging and voice recognition in language processing 語意分析中的詞性標記及語音辨識
  - Applications of sequence analysis or prediction in finance / bioinformatics 財務或生物資訊上的序列分析應用
- This can be done…
  - as a set of independent classification tasks. For example, step forward one per member of the sequence a time.
  - by making the optimal label for a given element dependent on the choices of nearby elements.

Example (1)

- "美國是個自由的國家"
  - moving a cursor from left to right, use the last n words to tag the current one.
  - as a classification task, for example, to Decision Tree or SVM.

W_{i-4}	W_{i-3}	W_{i-2}	W_{i-1}	W_i
美	國	是	個	自
國	是	個	自	由
是	個	自	由	的
個	自	由	的	國
自	由	的	國	家

Use the last 4 words to tag the current one

Example (1)

- "美國是個自由的國家"
  - to obtain probability $P(W_i|W_{i-1}, W_{i-2},\ldots, W_{i-n})$
  - for the sequence, when i=1, the probability is

P("美國是個自由的國家")=

P(國|美)xP(是|國)x P(個|是)xP(自|個)x P(由|自)xP(的|由)x P(國|的)xP(家|國)

* instead of product of the probabilities, sum of log is used usually in programming.

Example (1)

- "美國是個自由的國家"
  - in voice recognition, the probabilities of various candidates are evaluated, and the highest one is chosen.

P(“美國是個自由的國家”)=0.08 ←chosen

P(“美國似個自由的國家”)=0.05

P(“美國是個製油的國家")=0.07

Example (2)

- 若想利用股票今日價量尋找明日價格漲跌之關係，採用序列模型編碼如下

	Day_1	Day_2	Day_3	Day_4	Day_5	Day_6	Day_7	Day_8
價格	升	升	平	升	降	降	降	平
成交量	升	升	降	降	升	降	平	平

- 同樣可以使用分類、或連續的條件機率進行分析

Hidden Markov Model

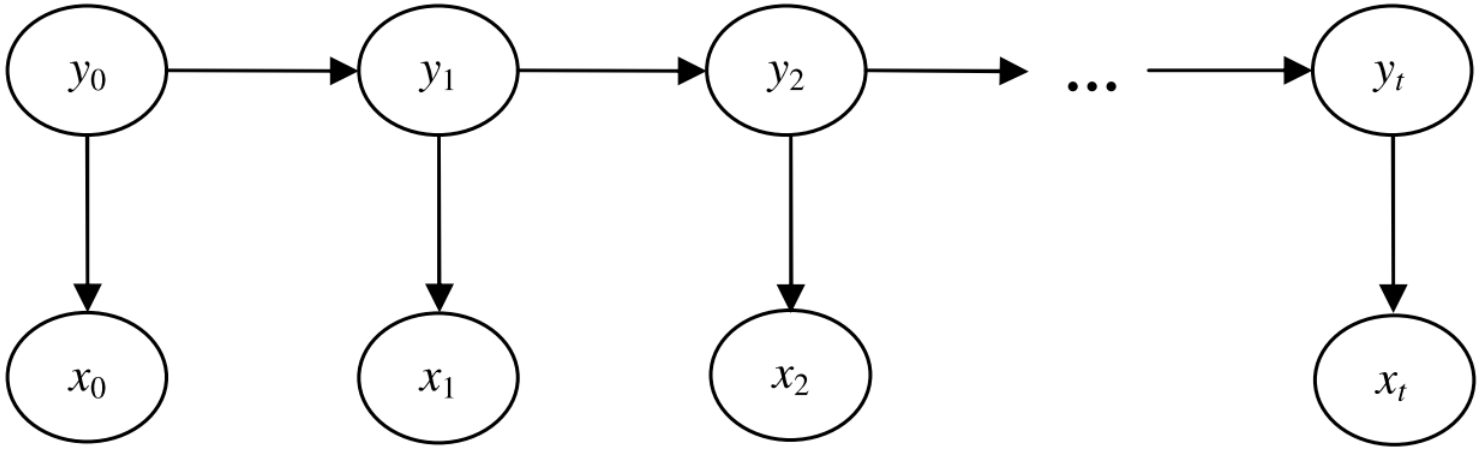


Fig. 3 Hidden Markov model

We have

$$Y = < y_0 , y_1 , \ldots y_t > = \text{hidden state sequence}$$

$$X = < x_0 , x_1 , \ldots x_t > = \text{observation sequence}$$

Ref. Lei Zhang and Bing Liu, "Aspect and Entity Extraction for Opinion Mining", 2014

HMM models a sequence of observations X by assuming that there is a *hidden* sequence of states Y. Observations are dependent on states. Each state has a probability distribution over the possible observations. To model the joint distribution $p(y, x)$ tractably, two independence assumptions are made. First, it assumes that state y_t only depends on its immediate predecessor state y_{t-1}. y_t is independent of all its ancestor $y_1, y_2, y_3, \ldots, y_{t-2}$. This is also called the *Markov* property. Second, the observation x_t only depends on the current state y_t. With these assumptions, we can specify HMM using three probability distributions: $p(y_0)$ over initial state, state transition distribution $p(y_t \mid y_{t-1})$ and observation distribution $p(x_t \mid y_t)$. That is, the joint probability of a state sequence Y and an observation sequence X factorizes as follows.

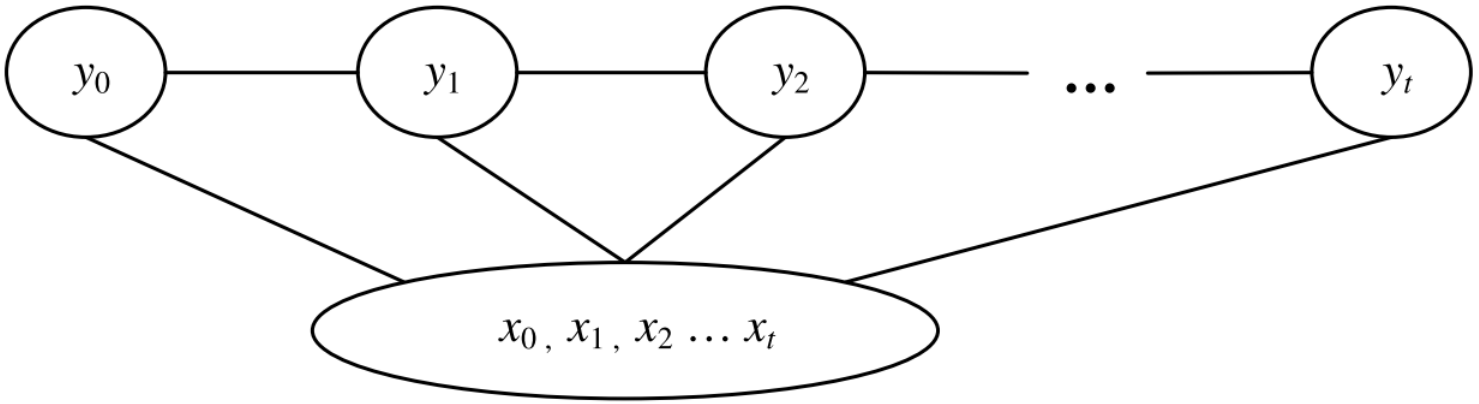
$$p(Y, X) = \prod_{t=1}^{t} p(y_t \mid y_{t-1}) p(x_t \mid y_t) \qquad (2)$$

where we write the initial state distribution $p(y_1)$ as $p(y_1|y_0)$.

Given some observation sequences, we can learn the model parameter of HMM that maximizes the observation probability. That is, the learning of HMM can be done by building a model to best fit the training data. With the learned model, we can find an optimal state sequence for new observation sequences.

Conditional Random Fields

One limitation of HMM is that its assumptions may not be adequate for real-life problems, which leads to reduced performance. To address the limitation, linear-chain Conditional Random fields (CRF) (Lafferty et al., 2001; Sutton and McCallum, 2006) is proposed as an undirected sequence model, which models a conditional probability $p(Y|X)$ over hidden sequence Y given observation sequence X. That is, the conditional model is trained to label an unknown observation sequence X by selecting the hidden sequence Y which maximizes $p(Y|X)$. Thereby, the model allows relaxation of the strong independence assumptions made by HMM. The linear-chain CRF model is illustrated in Figure 4.



Ref. Lei Zhang and Bing Liu, "Aspect and Entity Extraction for Opinion Mining", 2014

- CRF implementation
  - CRFsuite http://www.chokkan.org/software/crfsuite/
  - CRF++ https://taku910.github.io/crfpp/
  - MALLET http://mallet.cs.umass.edu/

Chinese Processing

Problems in Chinese Processing

- "小明日記：今日王叔叔來我家玩媽媽，說我做完作業後，可以吃點心。然後，王叔叔誇我作業做的好，於是抱起了我媽，媽叫叔叔小心一點，之後叔叔又親了我媽媽，也親了我。"
- "老師批復：拿回家讓你爸看看，是標點符號有問題，還是你王叔叔和你媽媽有問題。"

Problems in Chinese Processing

- 新詞 (out-of-vocabulary, OOV)
  - 九把刀拍了部新電影叫等一個人咖啡
- 斷詞 (term segmentation) 消除歧義
  - 我國代表現在正面臨很大的壓力
  - 不可以營利為目的
  - 在書中，蔡英文筆下的政治理念清晰可見

Problems in Chinese Processing

- 斷詞的困難：有時需依照更多的上下文意
  - 全台大停電　power failure in NTU
  - 全台大停電　power failure in Taiwan
- 同字(詞) 多意之問題
  - 統一(公司)大陸廠 : 統一為公司名稱，而非動詞
  - 喜歡上一個人 Like someone
  - 喜歡上一個人 Like to be alone
  - 喜歡上一個人 Like the last one

What's the Difference in Chinese ?

- Algorithms in English are based in "Term"
  前述主要演算法均基於詞做運算
  - Document → Paragraph → Sentence → Term
  - Some expand to Phrase 有些擴充至片語
  - Some change to n-gram 有些改用n-gram
- The major difference in Chinese
  - Character range space is much larger
    中文字元個數遠多過於其它語言
  - No obvious boundary between characters / terms.
    中文字或詞之間無明顯分隔符號

What's the Difference in Chinese ?

	中文	英文
單位	字(元) Character 詞 片語 句子 段落 文件	字母 Letter 字 Word 片語 Phrase 句子 Sentence 段落 Paragraph 文件 Document
統計資料	BIG5: 常用字約5000個, 次常用字約8000個 Unicode: 約4萬個漢字 注音: 共376個音(不含四聲變化) 中研院CKIP辭庫: 二字以上約13萬詞	Webster Dictionary: 470,000

Problem in Chinese Processing (1)

- Term Segmentation 斷詞 (i.e. 搶詞問題)
  - Example

    我國 代表 現在 正 面臨 很 大 的 壓力

    我 到 達文西 博物館
  - Solution

    1. 字典法：例如長詞優先法

    2. 法則式：例如文法式、構詞法則、歧義解決法則

    3. 訓練統計式：例如詞頻法 (最大詞頻組合) 等

    4. 自動分類式：將斷詞轉為分類問題
  - Result

    現今主要第 3 , 4 類方法, 正確率可達 9 成以上

Problem in Chinese Processing (2)

- Part-of-Speech Tagging 詞性標定
  - Example

我國	代表	現在	正	面臨	很	大	的	壓力
Nc	Na	Nd	Neqb	Nv	Dfa	Na	Na	Na
	VC		Na	VK		Nv	T	
	VK		Nv			VA	De	
			VC			VH	Di	
			VH			VJ		
			A			A		
			D					
			Da					

N.. 名詞
V.. 動詞
D.. 副詞
A 形容詞
T 語助詞

Problem in Chinese Processing (2)

- Part-of-Speech Tagging 詞性標定
  - Solution
    1. 訓練統計式：
       例如馬可夫機率模型
    2. 自動分類式：
       將詞性標定轉為分類問題
  - Result
    正確率可達 9 成以上
    可衍生出許多應用

表：中研院平衡語料庫詞類標記

I	/*感嘆詞*/
P	/*介詞*/
T	/*語助詞*/
VA	/*動作不及物動詞*/
VAC	/*動作使動動詞*/
VB	/*動作類及物動詞*/
VC	/*動作及物動詞*/
VCL	/*動作接地方賓語動詞*/
VD	/*雙賓動詞*/
VE	/*動作句賓動詞*/
VF	/*動作謂賓動詞*/
VG	/*分類動詞*/
VH	/*狀態不及物動詞*/
VHC	/*狀態使動動詞/
VI	/*狀態類及物動詞*/
VJ	/*狀態及物動詞*/
VK	/*狀態句賓動詞*/
VL	/*狀態謂賓動詞*/
V_2	/*有*/
DE	
SHI	
FW	

A	/*非謂形容詞*/
Caa	/*對等連接詞，如：和、跟*/
Cab	/*連接詞，如：等等*/
Cba	/*連接詞，如：的話*/
Cbb	/*關聯連接詞*/
Da	/*數量副詞*/
Dfa	/*動詞前程度副詞*/
Dfb	/*動詞後程度副詞*/
Di	/*時態標記*/
Dk	/*句副詞*/
D	/*副詞*/
Na	/*普通名詞*/
Nb	/*專有名稱*/
Nc	/*地方詞*/
Ncd	/*位置詞*/
Nd	/*時間詞*/
Neu	/*數詞定詞*/.
Nes	/*特指定詞*/
Nep	/*指代定詞*/
Neqa	/*數量定詞*/
Neqb	/*後置數量定詞*/
Nf	/*量詞*/
Ng	/*後置詞*/
Nh	/*代名詞*/

Problem in Chinese Processing (3)

- Unknown Term 未知詞 (或稱Out-of-Vocabulary)
  - Example

    新鮮人倪安東見面簽唱會 歌迷熱情喊凍蒜

    國際運動仲裁庭祕書長瑞伯表示世跆盟可拒仲裁
  - Solution

    1. 先經過斷詞，再處理未知部份

       未知部份以構詞法則處理，或n-gram統計學習

    2. 不經過斷詞，直接以訓練統計式處理
  - Result

    正確率可達 7~8 成 (含詞性標定)

Tool for Chinese Processing (1)

- Jieba 結巴中文分詞

  https://github.com/fxsjy/jieba

  - 開放程式碼，支援多種語言
  - 已知詞部分採詞頻法，找尋最大得分路徑
  - 未知詞部分採HMM標記
  - 準確率受詞典影響大
    - 例如：無法完整斷出「蔡英文」，因詞典中「英文」之得分大

Tool for Chinese Processing (2)

- eLand ETool
  - 開放完整API
  - 主要功能
    - 自動關鍵詞
    - 自動摘要
    - 斷詞與詞性標定
    - 情緒判定
- 試用展示

自動關鍵字 原理

- 範例
  - 今日馬英九總統…有馬英九粉絲忽然…馬英九立刻回應…
- 以雙連文進行統計
  - p(馬英|freq=1) ~ 1/n … > t 門檻值
  - p(馬英|freq=2) ~ $(1/n)^2$ … > t
  - p(馬英|freq=3) ~ $(1/n)^3$ … < t, found
  - p(馬英|freq=4) ~ $(1/n)^4$ …
  - 判定 馬英 係可能有意義的子字串
- 將所有可能有意義的子字串進行合併
  - p(馬英|freq=3) & p(英九|freq=3) < t
  - merge(馬英, 英九) = 馬英九

自動關鍵字 演算法

- 以n-gram，找出最長且最常結伴出現的字元串
  - 需指定所謂 "最常出現" 的次數門檻值

其演算法之主要步驟爲：

1. 先使用 Bigram 取出所有可能之中文雙連文，並計算各個雙連文的出現次數（occurrence frequency）。將所有雙連文置於 List 中
2. 當 List 不爲空時
   2.1 將 MergeList 設爲空
   2.2 於 List 之結尾處放入一特殊標記
   2.3 對於 List 中兩兩相連之元素 K1 與 K2，
   若 K1 與 K2 之出現次數皆高於 threshold T，代表可結合
   將 K1 與 K2 結合爲 K，將 K 置於 MergeList 中
   將 K 之出現次數加 1
   不然
   若 K1 之出現次數高於 T 且 K1 還沒有跟之前其它元素結合過
   將 K1 放入 FinalList
   2.4 將 List 設爲 MergeList
3. 將 FinalList 經適當過濾後即爲該文件之關鍵詞

以BACDBCDABACD為例
設定 threshold T = 1

FinalList會得到
CD：3
BACD：2

代表擷取出兩個關鍵字

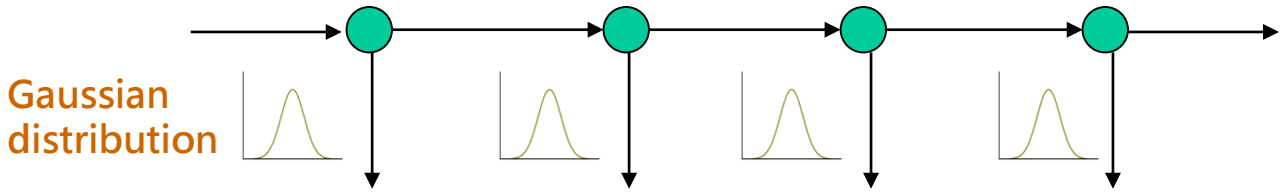
註: 類似LCS問題 (Longest Common Subsequence)、但限相鄰之算法

自動摘要

- 重新組合重要的句子
  - "句子" 作為單位
  - 以關鍵詞計算每個句子的得分
  - 由句子得分篩選固定比例的句子作為文章摘要

HMM 斷詞

- Hidden Markov Model
  - 統計機率式的模型
  - 序列資料的描述 (sequence)



Transition Prob.

	S_0	S_1	S_2
S_0	P_{00}	P_{01}	P_{02}
S_1	P_{10}	P_{11}	P_{12}
S_2	P_{20}	P_{21}	P_{22}

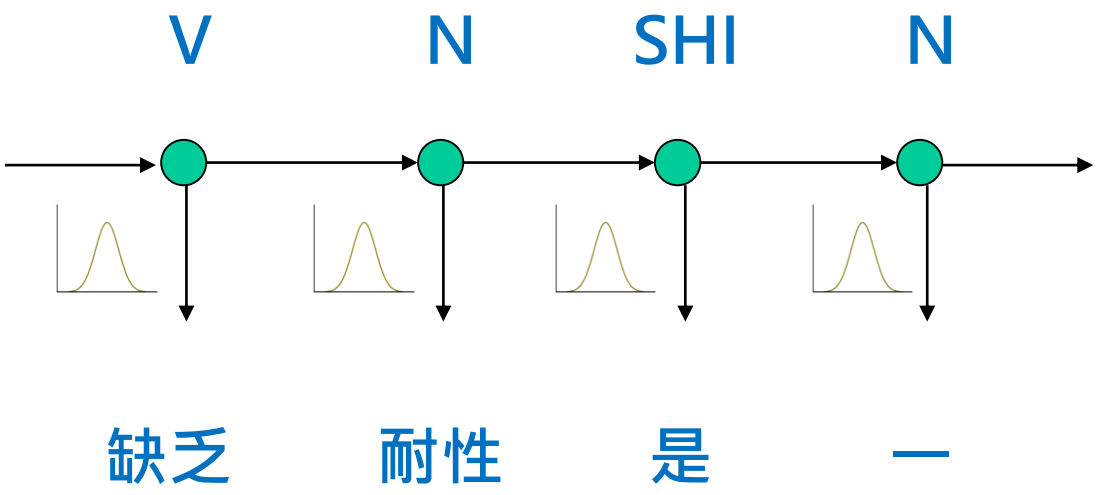
Observation Prob.

$$P_{S_i}(O_t)$$

HMM 斷詞

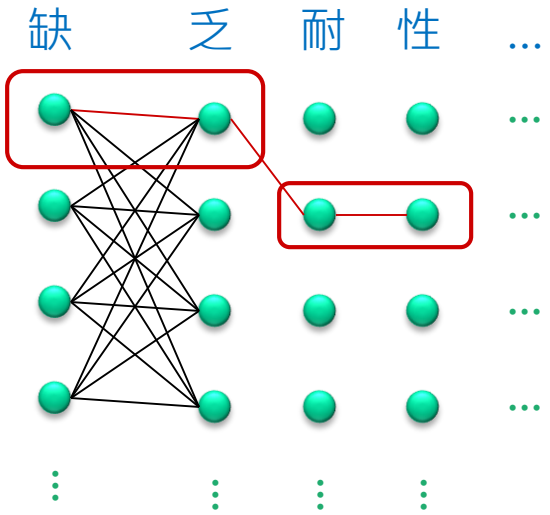
- 中文斷詞的應用
  - Ex.

    缺乏(V)耐性(N)是(SHI)一(N)項(N)莫大(A)的(D)致命傷(N)



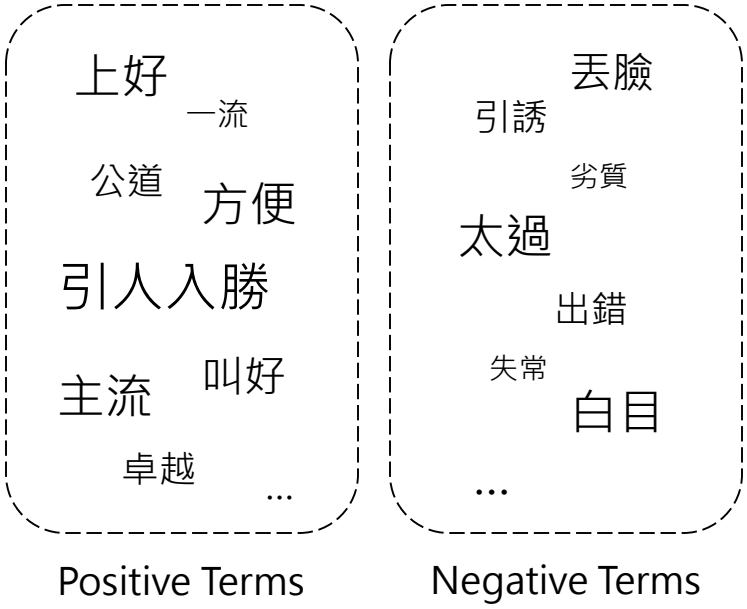
HMM 斷詞

- 中文斷詞的應用
  - 取得機率最高的路徑



情緒判別

- A bag-of-words



Okapi BM25

document

term set

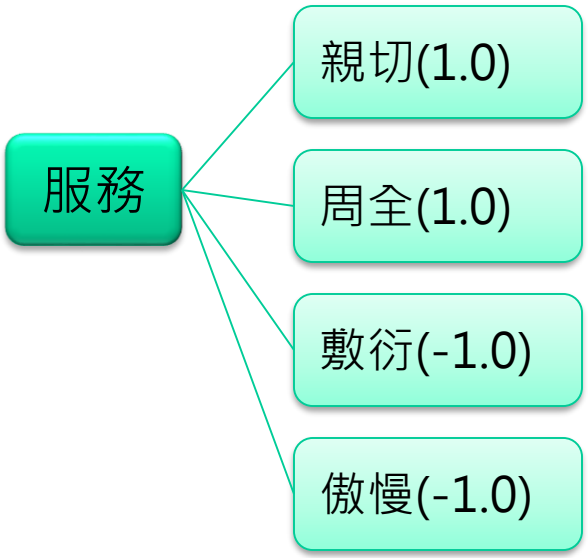
document length

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i,D)\cdot(k+1)}{f(q_i,D)+k(1-b+b\cdot\frac{|D|}{avgd})}$$

avg. document length

情緒判別

- Associate Attitude
  - 建立關聯態度詞庫



...這家代理商的**服務**一點也不周全...

...這家代理商的**服務**一點也不周全...

態度反轉

...這家代理商的**服務**一點也不**周全**...

Appendix

Chinese Text Retrieval without Using a Dictionary (Chen *et al*, SIGIR97)

- Segmentation
  - Break a string of characters into words
- Chinese characters and words
  - Most Chinese words consist of two characters (趙元任)
  - 26.7% unigrams, 69.8% bigrams, 2.7% trigrams (北京，現代漢語頻率辭典)
  - 5% unigrams, 75% bigrams, 14% trigrams, 6% others (Liu)
- Word segmentation
  - statistical methods, e.g., mutual information statistics
  - rule-based methods, e.g., morphological rules, longest-match rules, ...
  - hybrid methods

Indexing Techniques

- Unigram Indexing
  - Break a sequence of Chinese characters into individual ones.
  - Regard each individual character as an indexing unit.
  - GB2312-80: 6763 characters
- Bigram Indexing
  - Regard all adjacent pairs of hanzi characters in text as indexing terms.
- Trigram Indexing
  - Regard all the consecutive sequence of three hanzi characters as indexing terms.

Examples

sentence	$c_1 c_2 c_3 c_4 c_5 c_6$
unigrams	c_1, c_2, c_3, c_4, c_5, c_6
bigrams	c_1c_2, c_2c_3, c_3c_4, c_4c_5, c_5c_6
trigrams	$c_1c_2c_3$, $c_2c_3c_4$, $c_3c_4c_5$, $c_4c_5c_6$

Table 1: n-gram indexing methods

shows the number of unique and total number of unigrams, bigrams and trigrams in the TREC-5 Chinese test collection, about one third of possible Chinese bigrams occur at least once in the Chinese collection.

n-gram	no. distinct n-grams	no. n-grams
unigrams	6,236	64,611,662
bigrams	1,393,488	54,362,319
trigrams	8,119,574	49,886,331

Table 2: n-gram size of TREC-5 Chinese collection

Indexing Techniques (*Continued*)

- Statistical Indexing
  - Collect occurrence frequency in the collection for all Chinese characters occurring at least once in the collection.
  - Collect occurrence frequency in the collection for all Chinese bigrams occurring at lease once in the collection.
  - Compute the mutual information for all Chinese bigrams.

    $I(x,y)=\log_2(p(x,y)/(p(x)*p(y)))$

    $= \log_2((f(x,y)/N) /((f(x)/N)*(f(y)/N)))$

    $= \log_2((f(x,y)*N) /(f(x)*f(y)))$
  - Strongly related: much larger value
  - Not related: close to 0
  - Negatively related: negative

$I(x,y)=\log_2(p(x,y)/$
$(p(x)*p(y)))$
$=\log_2 (p(x)/ (p(x)*p(y))$
$=\log_2 (1/ *p(y))$

$I(x,y)=\log_2(p(x,y)/(p(x)*p(y)))$
$=\log_2(p(x|y)/p(x))$
$=\log_2(p(x|y)/p(x))=0$

f(*c*1): the occurrence frequency value of the first Chinese character of a bigram
f(*c*2): the occurrence frequency value of the second Chinese character
f(*c*1*c*2): the occurrence frequency value of a bigram
I(*c*1,*c*2): mutual information
I(*c*1,*c*2) >> 0, *c*1 and *c*2 have strong relationship
I(*c*1,*c*2) ~ 0, *c*1 and *c*2 have no relationship
I(*c*1,*c*2) << 0, *c*1 and *c*2 have complementrary relationship

bigrams	f(c1)	f(c2)	f(c1c2)	I(c1,c2)	
淘汰(eliminate)	1,549	1,632	1,343	15.06	
苹果(apple)	1,208	50,416	1,021	10.08	
漂亮(beautiful)	1,445	6,301	859	12.57	> 0
非常(unusually)	37,579	50,257	7,157	7.93	
如果(if)	57,975	50,416	10,884	7.91	
不水(not water)	311,474	90,495	1	-8.76	< 0

Table 3: Mutual information of six Chinese bigrams

bigrams	f(c1)	f(c2)	f(c1c2)	I(c1,c2)	
中国	615,222	925,353	228,090	3	4.69
国大	925,353	417,826	6,791	5	0.18
大陆	417,826	15,331	6,946	2	6.13
陆新	15,331	256,559	22	9	-1.46
新发	256,559	328,500	1,058	7	-0.30
发现	328,500	139,630	11,946	4	4.07
现的	139,630	2,017,405	4,340	6	-0.00
的油	2,017,405	26,690	676	7	-0.30
油田	26,690	24,869	2,412	1	7.87

Table 4: Mutual information values of bigrams.

step	phrases	action
1	中国大陆新发现的油田	remove 油田(oil fields)
2	中国大陆新发现的	remove 大陆(mainland)
3	中国 新发现的	remove 发现(discover)
4	中国 新 的	stop

Table 5: Word segmentation process using mutual information.

Segmentation as Classification

- 我國 代表 現在 正 面臨 很大 的 壓力

  B E B E B E S B E B E S B E

- 九把刀 不 同意

  B I E S B E

Table 1. Position tags in a word (BIES tags)

Tag	Description
S	one-character word
B	first character in a multi-character word
I	intermediate character in a multi-character word (for words longer than two characters)
E	last character in a multi-character word

5 Feature templates

(a) Default feature

(b) The current character (C_0)

(c) The previous (next) two characters (C_{-2}, C_{-1}, C_1, C_2)

(d) The previous (next) character and the current character (C_{-1} C_0, C_0 C_1),

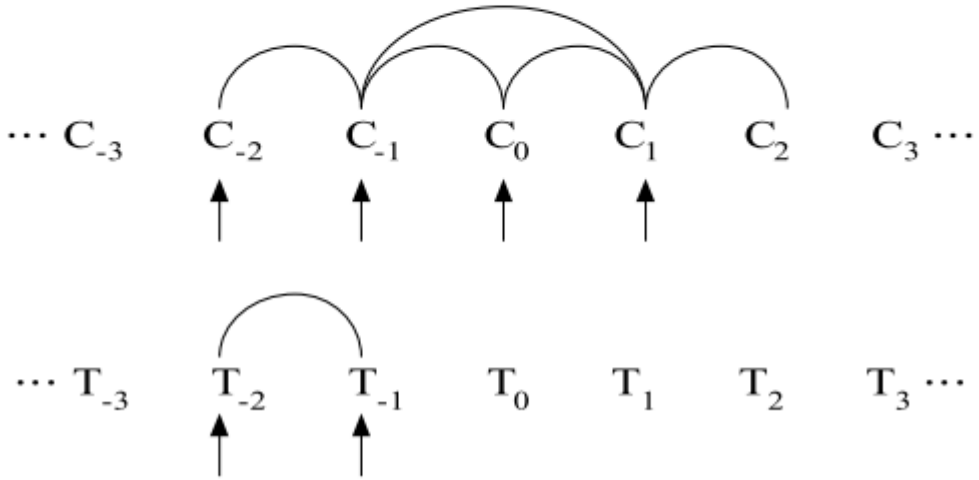
the previous two characters (C_{-2} C_{-1}), and

the next two characters (C_1 C_2)

(e) The previous and the next character (C_{-1} C_1)

(f) The tag of the previous character (T_{-1}), and

the tag of the character two before the current character (T_{-2})



– Training data : input features and the target

C-2	T-2	C-1	T-1	C1	T1	C2	T2	C0	T0 目標欄位
我	B	國	E	表	E	現	B	代	B
國	E	代	B	現	B	在	E	表	E
…									

Table 5. Bakeoff data

Corpus	# of train words	# of test words	Unknown word rate	Size of original dictionary	Size of dictionary used
PKU	1.1M	17,194	6.9%	55,226	36,830
CHTB	250K	39,922	18.1%	19,730	12,274
AS	5.8M	11,985	2.2%	146,226	100,161
HK	240K	34,955	7.1%	23,747	17,207

Table 6. Segmentation results obtained with bakeoff data

Corpus	*Recall*	*Precision*	*F-measure*	$Recall_{unknown}$	$Recall_{known}$
PKU	95.5	94.1	94.7	71.0	97.3
CHTB	86.0	83.5	84.7	57.7	92.2
HK	95.4	92.1	93.7	65.5	97.7
AS	97.0	94.8	95.9	69.0	97.6

from "Chinese Word Segmentation by Classification of Characters", 2005

討論：擴大應用

- Word Vector and Word Embedding
  - 將Word視為向量，計算Word的相似性
  - 更進一步，以上下文字，而非整篇文章為單位，可找出替換詞
    - Ex. 今天的天氣不錯，明天的天氣不錯：今天←→明天
    - Ex. 馬英九總統...，蔡英文總統...：馬英九←→蔡英文

Recap: the weight matrix

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth . . .
ANTHONY	5.25	3.18	0.0	0.0	0.0	0.35
BRUTUS	1.21	6.10	0.0	1.0	0.0	0.0
CAESAR	8.59	2.54	0.0	1.51	0.25	0.0
CALPURNIA	0.0	1.54	0.0	0.0	0.0	0.0
CLEOPATRA	2.85	0.0	0.0	0.0	0.0	0.0
MERCY	1.51	0.0	1.90	0.12	5.25	0.88
WORSER	1.37	0.0	0.11	4.15	0.25	1.95
. . .						

Each document is now represented as a real-valued vector of tf idf weights $\in R^{|V|}$.

Each word may be represented as a real-valued vector, too.

OpView Dictionary

Term

全聯

請輸入一個關鍵詞

數量

10

請輸入回傳數量

查詢來源

ptt-r1612

顯示結果　取得json

備註

部分關鍵字經過簡轉繁處理，搜尋時若找不到，建議嘗試繁體。

例如： 台 -> 臺

已輸入資料

關鍵字
全聯
數量
10
來源
ptt-r1612
狀態
查詢成功

全聯←→ 頂好、家樂福、大潤發、愛買、
美廉社、量販店、寶雅、屈臣氏、楓康

no.	詞彙	相似度	功能
1	頂好	0.6567696332931519	反查
2	家樂福	0.6359816193580627	反查
3	大潤發	0.6202948093414307	反查
4	愛買	0.5848792195320129	反查
5	美廉社	0.5537295937538147	反查
6	量販店	0.531093180179596	反查
7	寶雅	0.5226638317108154	反查
8	屈臣氏	0.5093675851821899	反查
9	楓康	0.5076067447662354	反查

Other Language Issues

Tokenization

- Input: "***Friends, Romans and Countrymen***"
- Output: Tokens
  - ***Friends***
  - ***Romans***
  - ***Countrymen***
- Each such token is now a candidate for further processing
  - 正規化及語言處理
  - 在此一階段就直接丟棄 (保留) 哪些資訊？
  - 索引與查詢(分析)時的處理要一致

Tokenization

- n-gram or n-word ?
  - 德文 Taiwan verbietet Verzehr von Hunden und Katzen
  - 印尼文 Taiwan Segera Berlakukan Undang-undang Larangan Makan Daging Anjing dan Kucing
  - 義大利文 mangiare cani e gatti diventa illegale
  - 越南文 Cộng hòa xã hội chủ nghĩa Việt Nam
  - 日文 シンガポール
  - 韓文 대만쇼케이스
  - 泰文 ฟุตบอลหญิงชิงแชมป์เอเชีย

Tokenization

- Issues in tokenization:
  - ***Finland's capital*** → ***Finland? Finlands? Finland's***?
  - ***Hewlett-Packard*** → ***Hewlett*** and ***Packard*** as two tokens?
  - ***San Francisco***: one token or two? How do you decide it is one token?

Numbers

- ***3/12/91 Mar. 12, 1991***
- ***55 B.C.***
- ***B-52***
- ***My PGP key is 324a3df234cb23e***
- ***100.2.86.144***
  - Often, don't index as text.
    - But often very useful
    - mixed with text: ex. 產品型號 Nikon D700

    (One answer is using n-grams)

Tokenization: Language issues

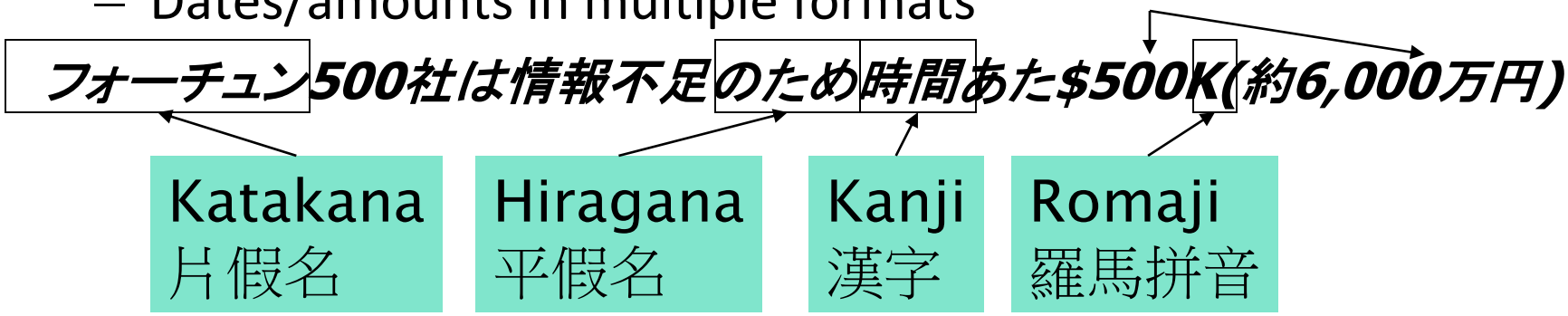
- ***L'ensemble*** → one token or two?
  - ***L*** ? ***L'*** ? ***Le*** ?
  - Want ***l'ensemble*** to match with ***un ensemble***
- German noun compounds are not segmented
  - Lebensversicherungsgesellschaftsangestellter
  - 'life insurance company employee'

Tokenization: language issues

- Chinese and Japanese have no spaces between words:
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - Not always guaranteed a unique tokenization

斷詞問題

- Further complicated in Japanese, with multiple alphabets intermingled 混合使用
  - Dates/amounts in multiple formats



Normalization

- Need to “normalize” terms in indexed text as well as query terms into the same form
  - We want to match ***U.S.A.*** and ***USA***
  - 索引與查詢(分析)時的處理要一致
- Alternative is to have multiple tokenization
  - mixed language processing and n-gram approach

Normalization: other languages

- Accents: ***résumé*** vs. ***resume***.
- Most important criterion:
  - Even in languages that standardly have accents, users often may not type them
  - How would you like to present in the final result ?
- German: Tuebingen vs. Tübingen
  - Should be equivalent
- 7月30日 vs. 7/30

Case folding

- Reduce all letters to lower case
  - exception: upper case (in mid-sentence?)
    - e.g., ***General Motors***
    - ***Fed*** vs. ***fed***
    - ***SAIL*** vs. ***sail***
  - One approach is to lower case everything in analysis, meanwhile to represent in the original form

Stop words

- With a stop list, you exclude from dictionary entirely the commonest words. Intuition:
  - They have little semantic content: *the, a, and, to, be*
  - They take a lot of space: ~30% of postings for top 30
- But the trend is away from doing this:
  - You need them for:
    - Phrase queries: “King of Denmark”
    - Various song titles, etc.: “Let it be”, “To be or not to be”
    - “Relational” queries: “flights to London”

Thesauri and soundex

- Handle synonyms 同義字 and homonyms 同音字
  - Hand-constructed equivalence classes
    - e.g., ***car*** = ***automobile***
    - ***color*** = ***colour***
- Rewrite to form equivalence classes
- 原則：兩種方式，在索引時處理？或在查詢時處理？
  - (1) Index such equivalences
    - Ex. When the document contains ***automobile***, index it under ***car*** as well (usually, also vice-versa)
  - (2) expand query
    - Ex. When the query contains ***automobile***, look under ***car*** as well

Soundex

- Traditional class of heuristics to expand a query into phonetic equivalents
  - Language specific – mainly for names
  - E.g., ***chebyshev*** $\rightarrow$ ***tchebycheff***

Lemmatization

- Reduce inflectional/variant forms to base form
- E.g.,
  - *am, are, is* → *be*
  - *car, cars, car's, cars'* → *car*
- *the boy's cars are different colors* →

  the boy car be different color
- Lemmatization implies doing "proper" reduction to dictionary headword form

Stemming

- Reduce terms to their "roots" before indexing
- "Stemming" suggest crude affix chopping
  - 很粗略地將字首字尾去除
  - language dependent
  - e.g., ***automate(s), automatic, automation*** all reduced to ***automat***.

for example compressed and compression are both accepted as equivalent to compress.

for exampl compress and compress ar both accept as equival to compress

Porter's algorithm

- Commonest algorithm for stemming English
  - Results suggest at least as good as other stemming options
- Conventions + 5 phases of reductions
  - phases applied sequentially
  - each phase consists of a set of commands
  - sample convention: *Of the rules in a compound command, select the one that applies to the longest suffix.*

    由一群合併或判斷規則所組成 → 挑選適用最長字尾的規則

Typical rules in Porter

- *sses* → *ss*
- *ies* → *i*
- *ational* → *ate*
- *tional* → *tion*

- Weight of word sensitive rules
- *(m>1) EMENT* →
  - *replacement* → *replac*
  - *cement* → *cement*

Other stemmers

- Other stemmers exist, e.g., Lovins stemmer
  http://www.comp.lancs.ac.uk/computing/research/stemming/general/lovins.htm
  - Single-pass, longest suffix removal (about 250 rules) 計算代價高

- Do stemming and other normalizations help?
  - help recall for some queries
    有利找到資料 (檢出率上升)
    Ex. 找 意大利, 同時找出義大利
  - but harm precision on others
    但可能會找到錯的資料 (精確率下降)
    Ex. 找 Steve Jobs 卻找到 steve's job