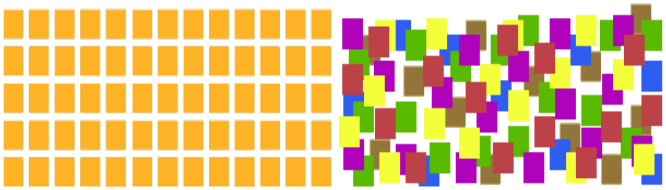
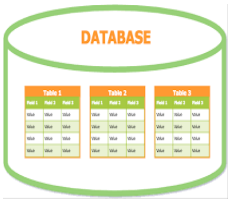
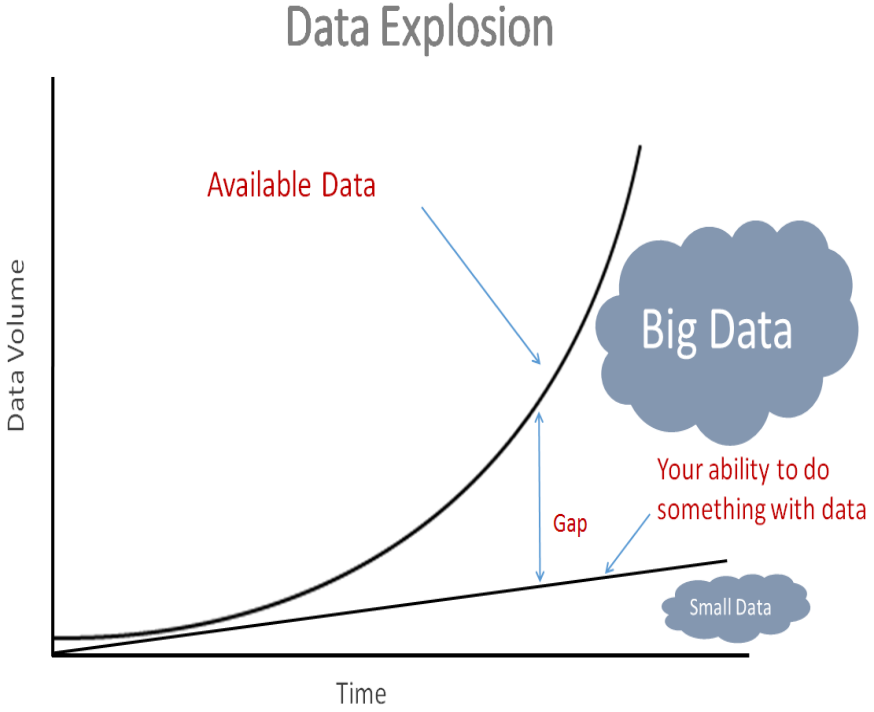
TEXT MINING WITH R

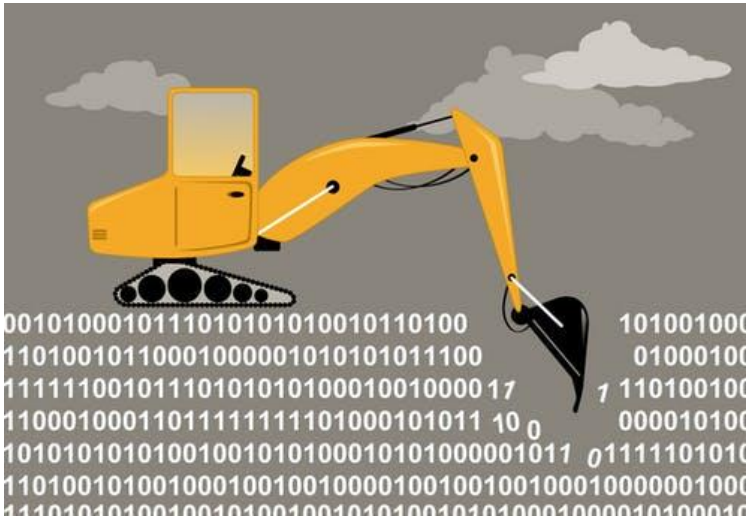
"80% of business-relevant information originates in unstructured form, primarily text."

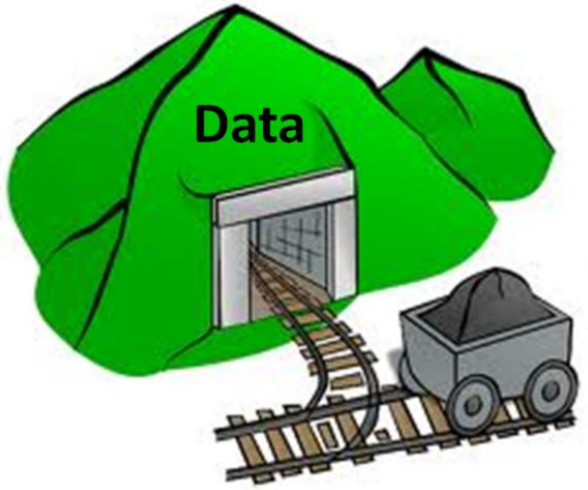


Structured Data vs. Unstructured Data





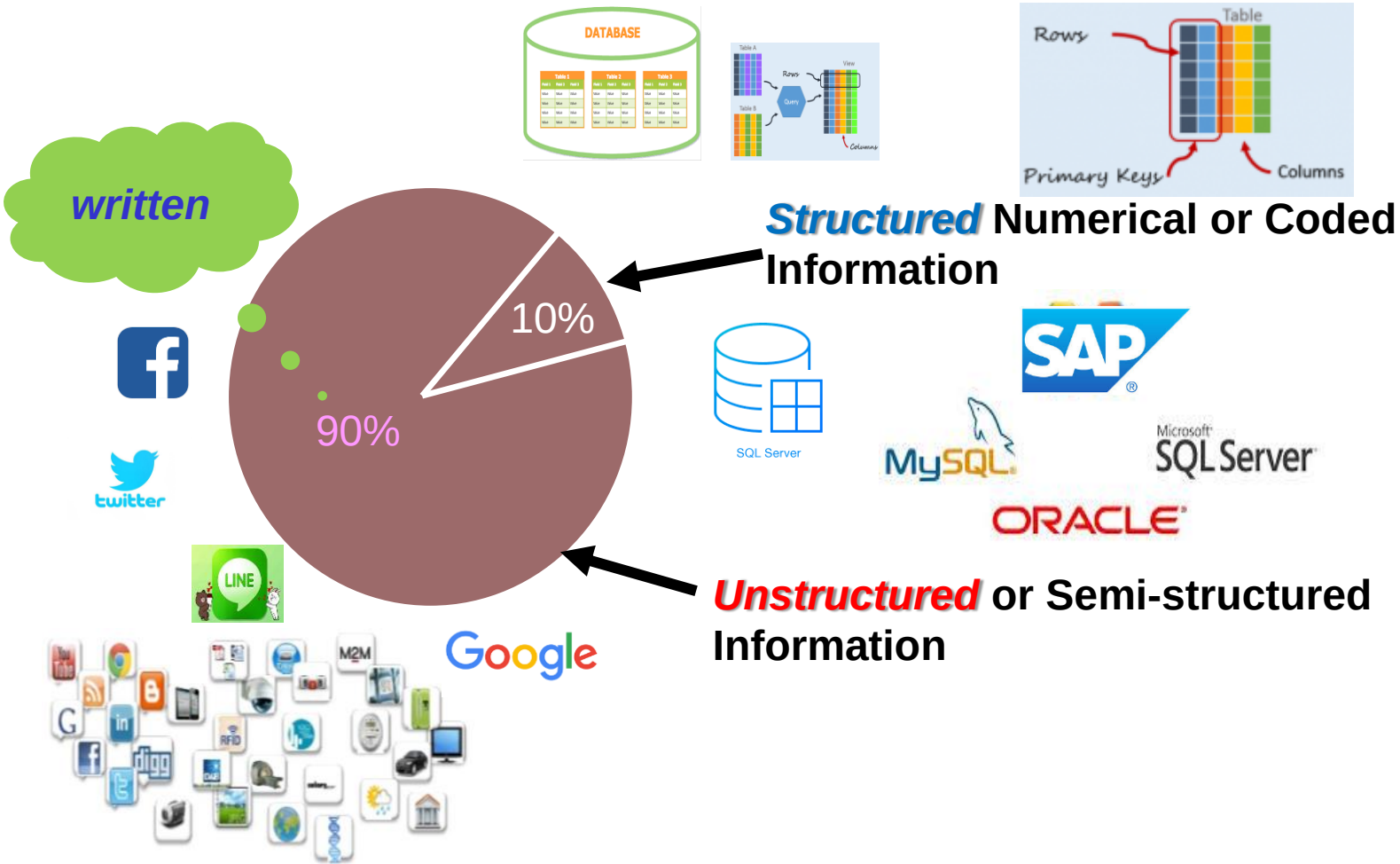




Useful knowledge (patterns)

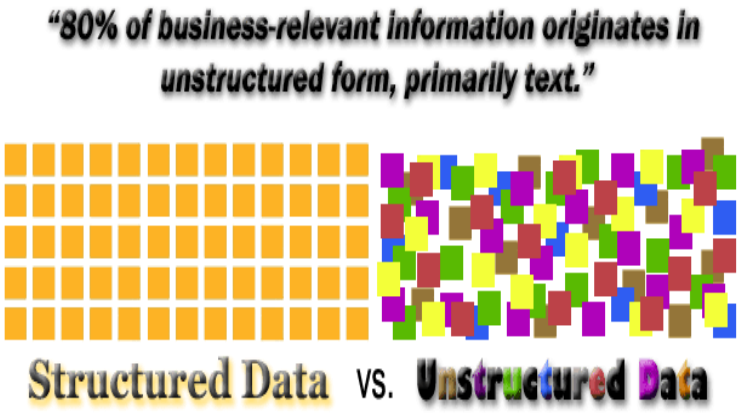
Motivation for Text Mining

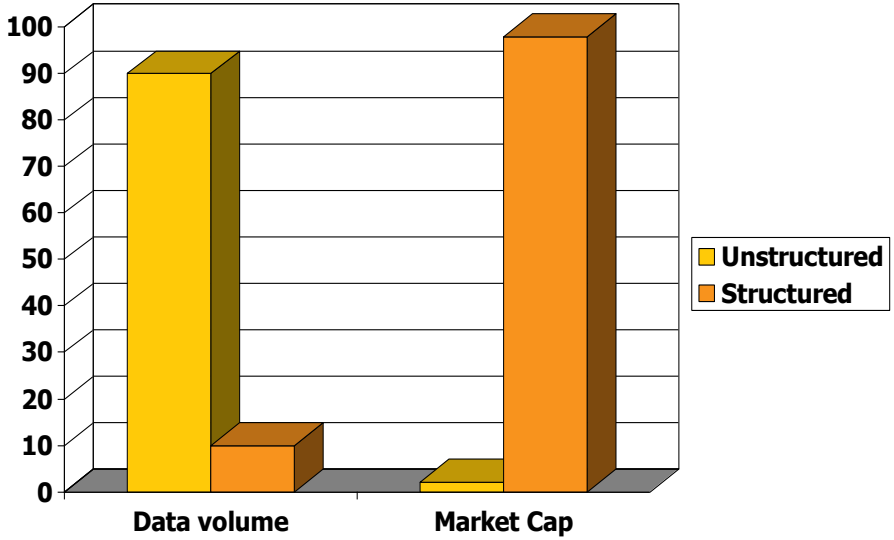
- Approximately **90%** of the world's data is held in unstructured formats (source: Oracle Corporation)



Motivation for Text Mining

- 過去有這樣的分析，Oracle(甲骨文)這些公司在處理的是**Structured Data**，資料量占比是10%，這樣的資料量與公司市值的關係比例為何?



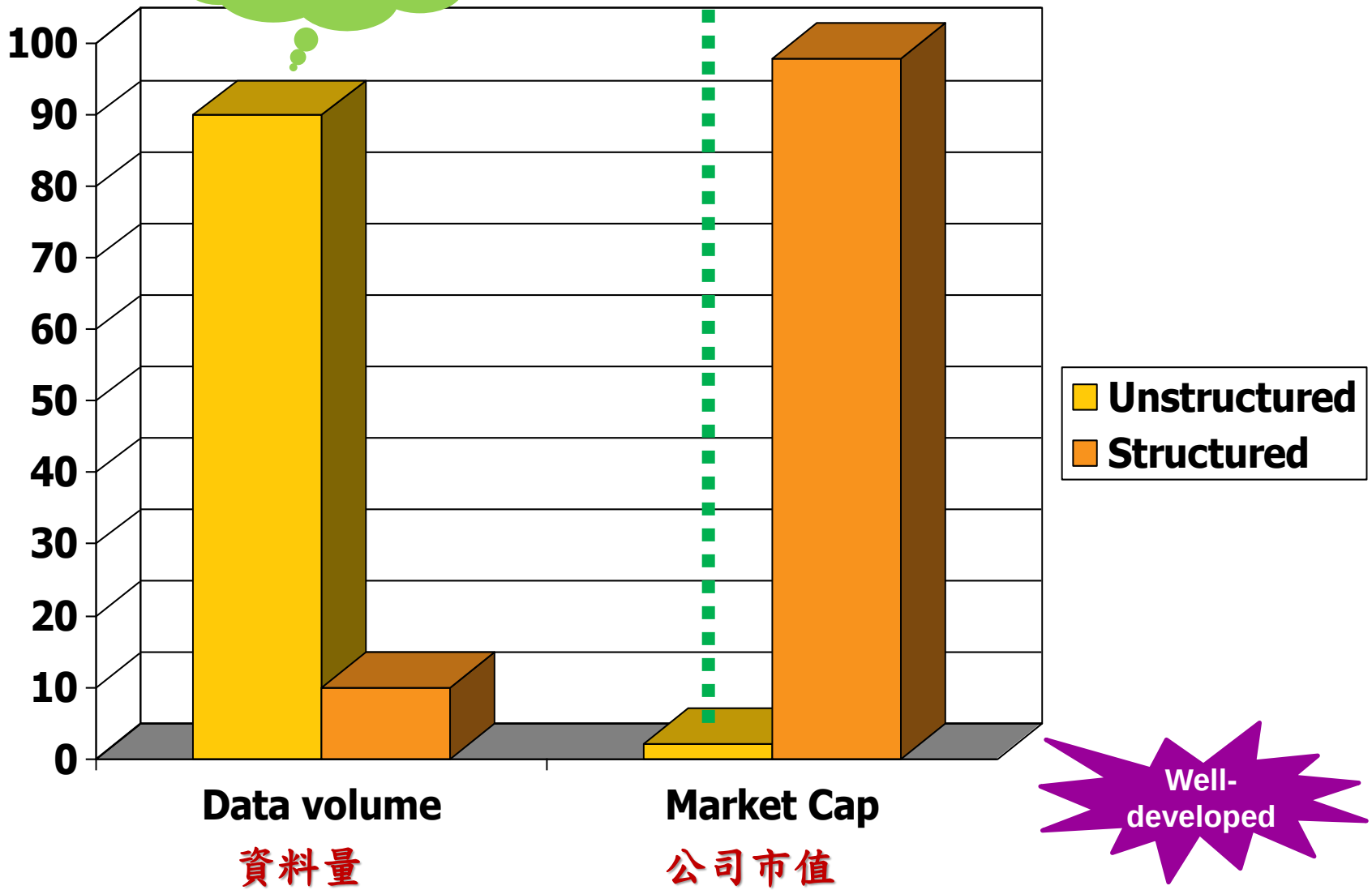


- 所以Text mining不只在學理上有趣，在經濟上面也是一個很有機會的領域。

Business Opportunity

右邊長條圖的9倍高

美國新創公司
的機會與方向



written documents
有些時候會很忠實地
表達您真正的想法
real opinion

Sample Business Applications

豐田暴衝害命 367 億和解
https://tw.news.appledaily.com/international/daily/20140321/35714931/

- Decision Support in CRM
  - What are the customers' typical complaints?(客戶抱怨)
  - Personalization in ecommerce
    - e.g. Line/FB/Website message or comment
  - Suggest products that fit a user's interest profile (based on personality information)
- Industry
  - Identifying competitors' web pages(全球競爭對手網頁資訊)
    - e.g. competing products and their prices
- Job seeking
  - Identify parameters in searching for jobs
    - e.g. 104 or 1111, how to write the resume or CV?
- Marketing
  - Discover potential buyers according to a user's text based profile(e.g. 2 years later→new iPhone)

What is text mining?

- **the discovery by computer of new, previously unknown information, by automatically extracting information from different *written* resources.**

- **First,定義中強調*written*這一個字**
  - **不只是手寫的，還包含word 、power point…等等軟體環境所打字打出來的內容都算是*written* resources ，任何被打出來的東西都算是text。**

- **Second,再來從這些 *written* resources中去找出之前沒有發現的資訊(unknown information)**

- **Third,要能夠自動(by automatically)找到，所以使用一些R或Python等工具去*written* resources中自動找到一些蛛絲馬跡，這就是text mining的精神。**

- *Well kids, I had an awesome birthday thanks to you. =D Just wanted to so thank you for coming and thanks for the gifts and junk. =) I have many pictures and I will post them later. hearts*

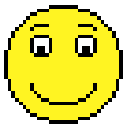
What are the characteristic words of these two moods?

Home alone for too many hours, all week long ... screaming child, headache, tears that just won't let themselves loose.... and now I've lost my wedding band. I hate this.

[Mihalcea, R. & Liu, H. (2006). A corpus-based approach to finding happiness, In *Proceedings of the AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*.]
Slide based on Rada Mihalcea's slides in the presentation.

- *Well kids, I had an awesome birthday thanks to you. =D Just wanted to so thank you for coming and thanks for the gifts and junk. =) I have many pictures and I will post them later. hearts*

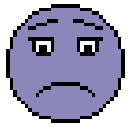
current mood: *positive*



What are the characteristic words of these two moods?

Home alone for too many hours, all week long ... screaming child, headache, tears that just won't let themselves loose.... and now I've lost my wedding band. I hate this.

current mood: *negative*



[Mihalcea, R. & Liu, H. (2006). A corpus-based approach to finding happiness, In *Proceedings of the AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*.]
Slide based on Rada Mihalcea's slides in the presentation.

Positive Vs. Negative

- 如何看出?
- 有無關鍵字呢?
- 人腦可判斷出這兩篇文章在說什麼，那電腦呢?

- 學習目標
  - 電腦有沒有辦法經由一些Preprocessing的步驟之後稍微有一點點能力可以做這樣子的判別呢?
    - 這篇文章在說什麼?這兩篇文章像嗎?
  - 人腦可以判斷出這兩篇文章不像，但是電腦能嗎?
  - 人腦可以判斷出上面那篇是Positive Sentiment，下面那一篇是Negative Sentiment，那麼電腦有辦法嗎?

- 要如何教會電腦做這些事情呢?

- *Well kids, I had an awesome birthday thanks to you. =D Just wanted to so thank you for coming and thanks for the gifts and junk. =) I have many pictures and I will post them later. hearts*

current mood: *positive*

What are the characteristic words of these two moods?

Home alone for too many hours, all week long ... screaming child, headache, tears that just won't let themselves loose.... and now I've lost my wedding band. I hate this.

current mood: *negative*

	Awesome	Thank	gift	Scream	Headache	tear	lose
D1	1/36	2/36	1/36	0	0	0	0
D2	0	0	0	1/29	1/29	1/29	1/29

Positive Or Negative

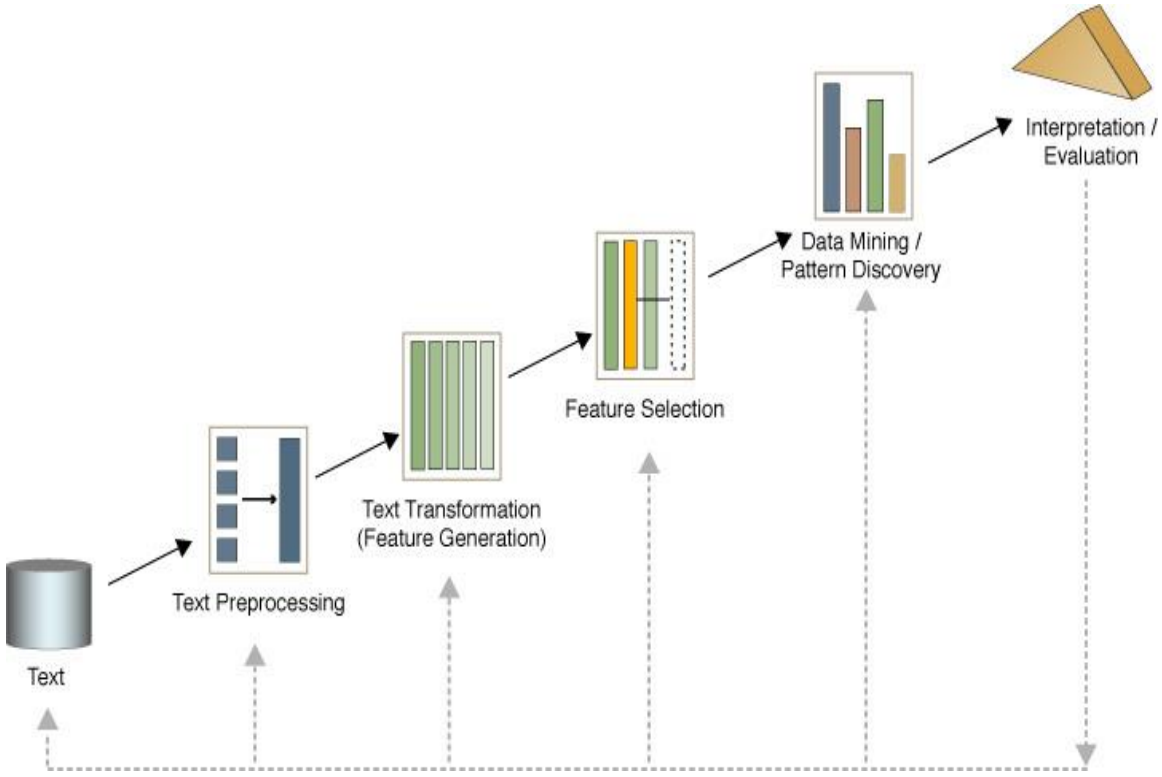
The goal: text representation

- Basic idea:
  - Keywords are extracted from texts.
  - These keywords describe topical content
  - Based on the ***vector*** *space model* of document collections:
    - Each unique word in a corpus of Web pages = one **dimension**
    - Each page is a vector with non-zero weight for each word in that page, zero weight for other words

➔ **Words** become **"features"** (in a data-mining sense)

Text mining **process**

- **Text**
  - **Document Collection**
  - **Text Characteristics**
- **Text preprocessing**
  - **Syntactic/Semantic text analysis**
- **Feature Generation**
  - **Bag of words**
- **Feature Selection**
- **Text/Data Mining**
  - **Classification**
  - **Clustering**
- **Analyzing results**



Individual articles

資料
Source of Text

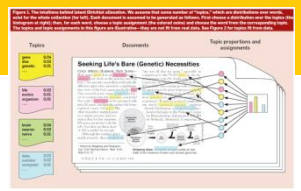
1

xxxooovvv

xxx

ooo

vvv

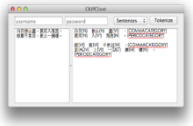


Sentiment Analysis
Topic Models

1a

中文
(Chinese)

English



Magazine corpus

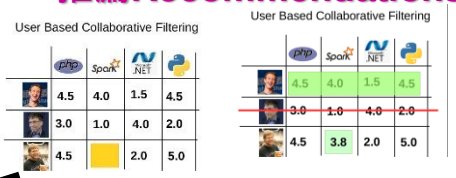
斷字/分詞
(Chinese Word Segmentation)
Ex. CKIP / R / Python

Corpus

2 語料庫
(Corpus)

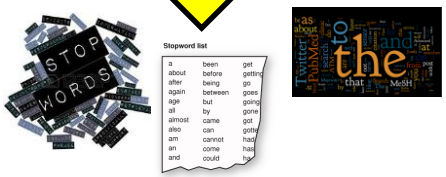
6

推薦Recommendations

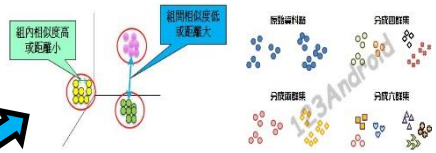


3

資料整理
(Data Organization)
1.Stop words
2.Stemming



集群分析Clustering

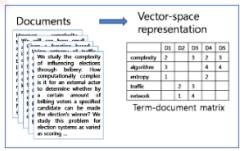


4

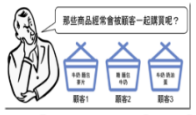
資料矩陣(Data Matrix)
次數Frequency/權重Weighting
Ex. tf / idf / tfidf / entropy

tdm

dtm



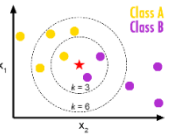
	T1	T2	T3	T4	T5	T6	T7	T8
Doc1	2	0	4	3	0	1	0	2
Doc2	0	2	4	0	2	3	0	0
Doc3	4	0	1	3	0	1	0	1
Doc4	0	1	0	2	0	0	1	0
Doc5	0	0	2	0	0	4	0	0
Doc6	1	1	0	2	0	1	1	3
Doc7	2	1	3	4	0	2	0	2



關聯規則
Association Rules

視覺化工具(Visualization tool)
Ex.文字雲(Word Cloud)

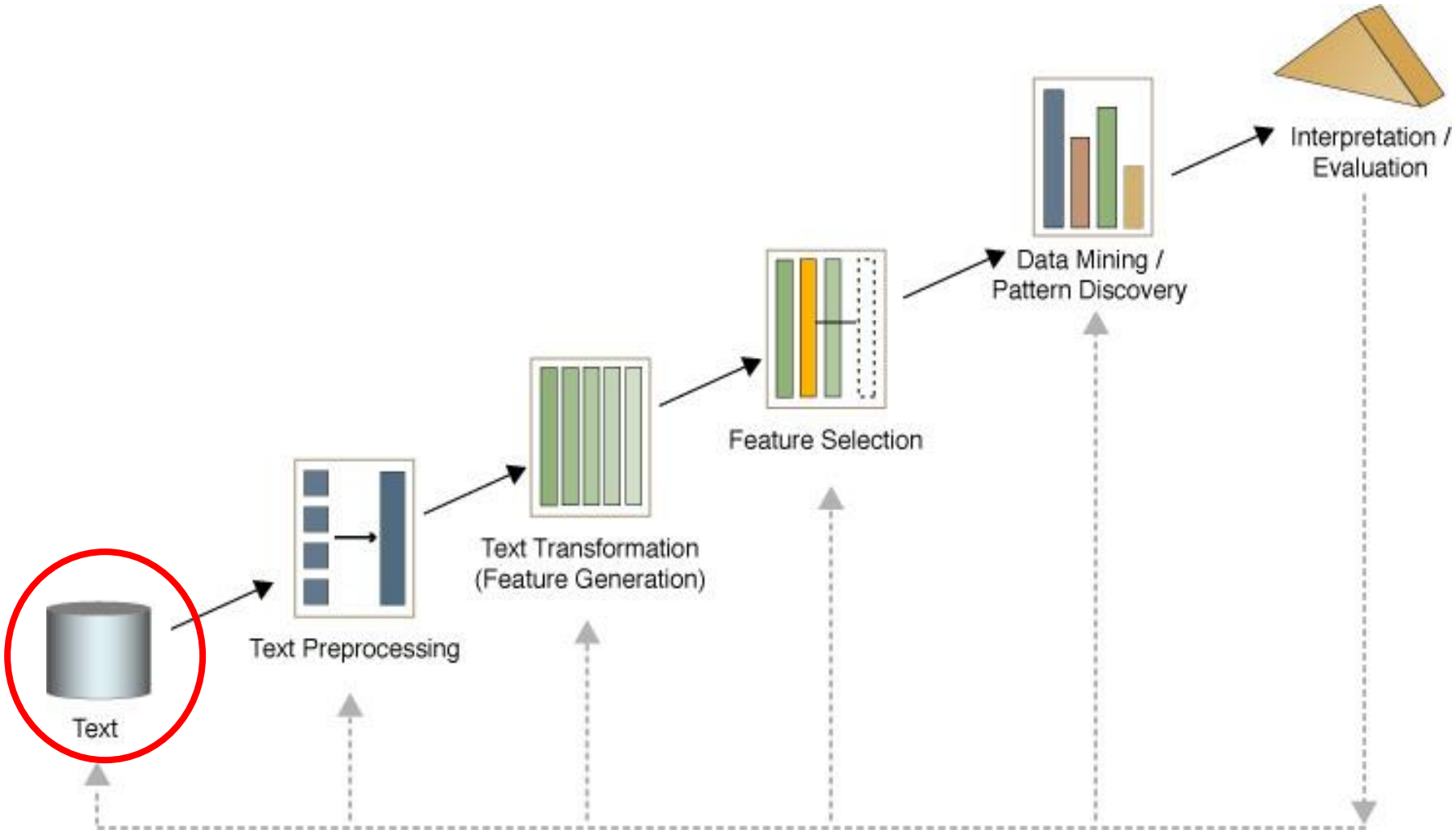
5



資料分類Classification

Sources of Text

- Document Collections



Document Collection

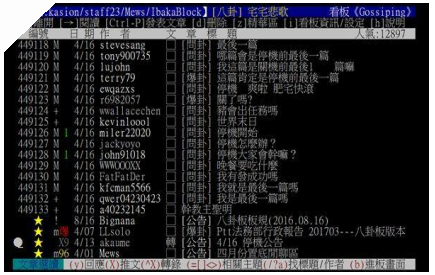
- Large volume of textual data
  - –Billions of documents must be handled in an efficient manner

Text Resources

- Email
- News Articles
- Web Pages
- Microblogs
- Scientific Articles
- Contracts
- Patent Portfolios
- Technical Documents
- Customer complaint letters
- Transcripts of phone calls with customers
- Lawsuit documents

網路爬蟲(web crawler)

- 也叫網路蜘蛛（spider）
- Ex. ptt



[爬蟲實戰] 如何爬取PTT的網頁?

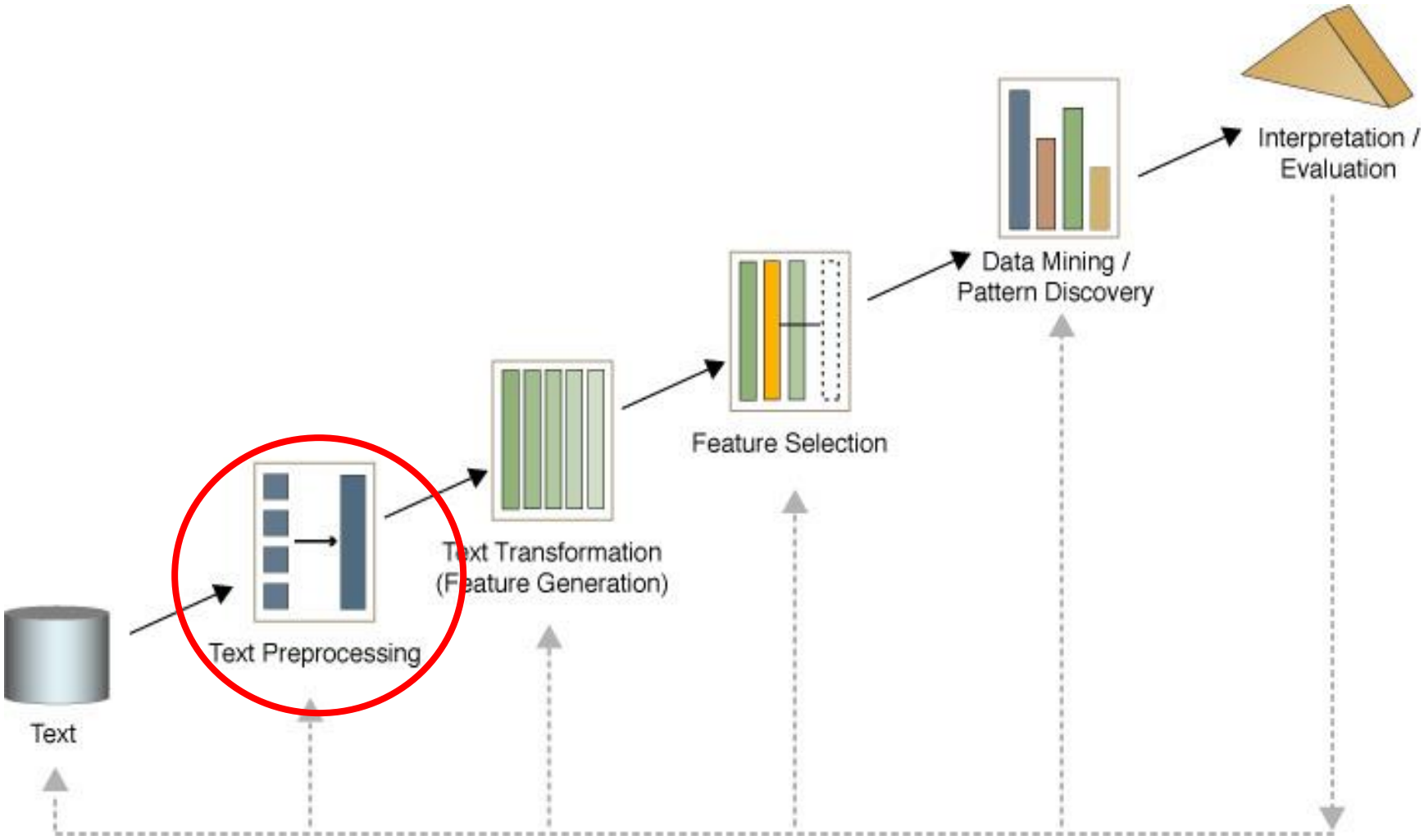
網路爬蟲、分析與視覺化

Practice 1

- 批踢踢網站說明
  - https://www.ptt.cc/bbs/index.html
- R code for web crawler
  - Open the file text_R.txt

Text preprocessing

- Syntactic/Semantic analysis



- Stop Words(停用字)
  - English: A, ABOUT, ABOVE, ACROSS, AFTER, AGAIN, AGAINST, ALL, ALMOST, ALONE, ALONG, ALREADY, ALSO,…
  - 中文:我、啊、的、曰…
- Stemming(詞幹化)
  - learn, learns, learned, learning…
  - walk, walks, walking, walked…
  - 中文沒這問題

Stop Words (停用字)

- Stop-words are words that from non-linguistic view do not carry information
  - …they have mainly functional role
  - …usually we remove them to help the methods to perform better
- Natural language dependent –examples:
  - English: A, ABOUT, ABOVE, ACROSS, AFTER, AGAIN, AGAINST, ALL, ALMOST, ALONE, ALONG, ALREADY, ALSO,
  - 我、啊

Stemming(詞幹化)

- Different forms of the same word are usually problematic for text data analysis, because they have different spelling and similar meaning(e.g. learns, learned, learning,…)

- Stemming is a process of transforming a word into its stem (normalized form)

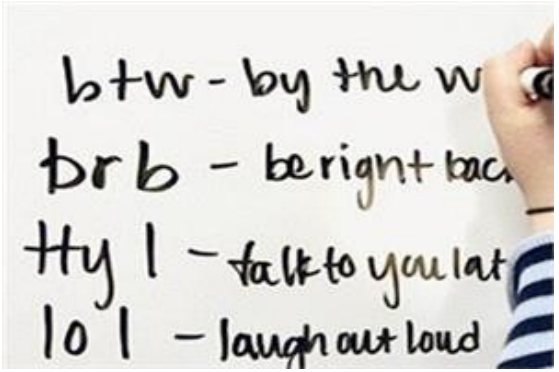
- Original text

- Information Systems Asia Web -provides research, IS-related commercial materials, interaction, and even research sponsorship by interested corporations with a focus on Asia Pacific region.
- Survey of Information Retrieval -guide to IR, with an emphasis on web-based projects. Includes a glossary, and pointers to interesting papers.

- After the stop-words removal

- Information System Asia Web provide research IS-relate commercial material interaction research sponsorship interest corporation focus Asia Pacific region
- Survey Information Retrieval guide IR emphasis web-base project Include glossary pointer interest paper

Noise



- Noisy data
  - Spelling mistakes : Aple
  - Abbreviations: e-commerce
  - Acronyms: EC

- Not well structured text
  - Email/Chat rooms
    - "r u available ?"
    - "Hey whazzzzzz up"
    - orz
  - Multilingual

Chinese Word segmentation

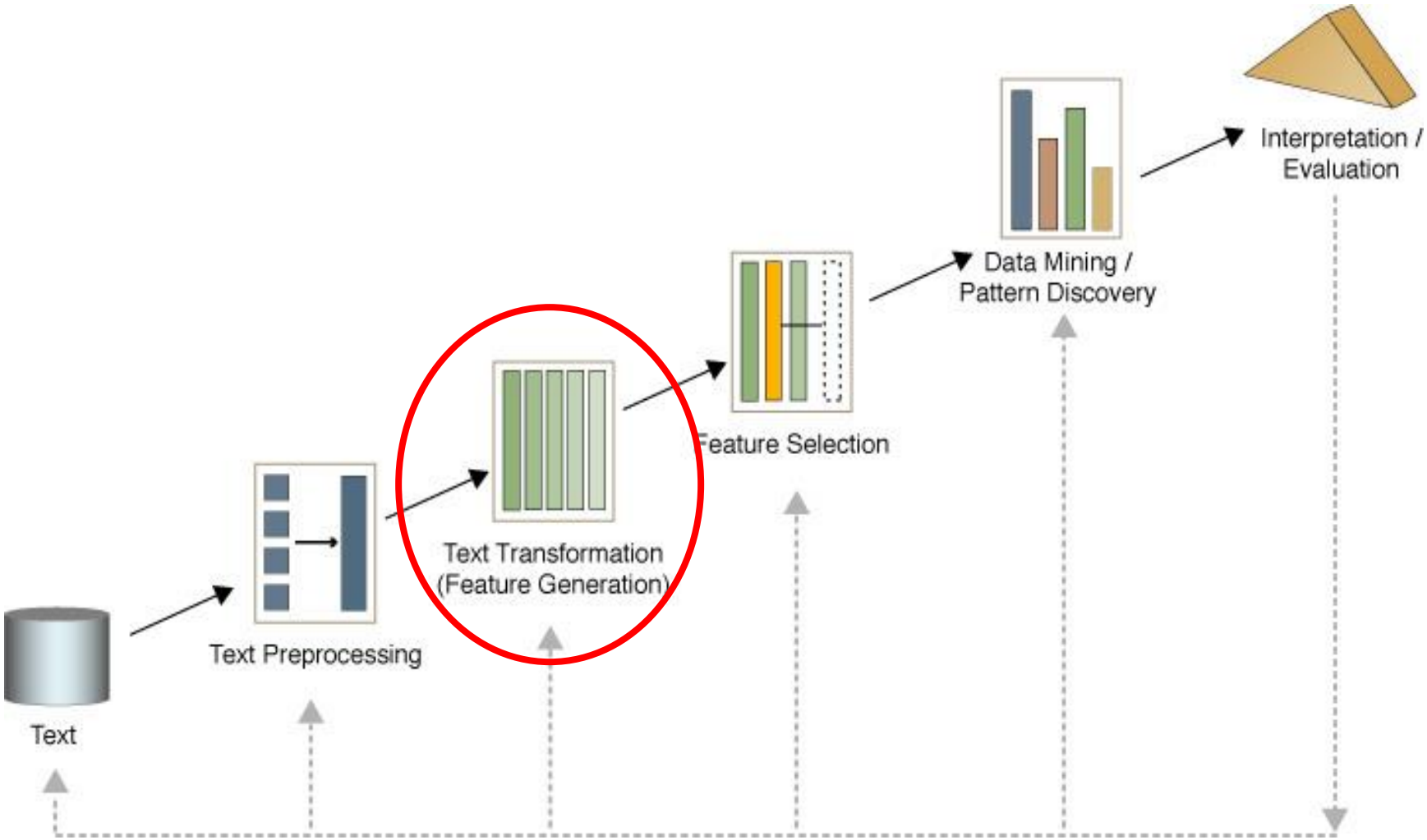
- Chinese has word segmentation issues
  - English uses space to separate words, but Chinese words are linked together
  - 下雨天留客天留我不留
  - 下雨，天留客。天留，我不留
  - 下雨天，留客天。留我不?留。
- Word segmentation is a serious issue for Chinese
  - 中研院斷詞系統:http://ckipsvr.iis.sinica.edu.tw/
- R
  - jiebaR package

Practice 2

- R code
  - Open the file vectorTM_R.txt

Feature Generation

- Bag of words



Feature Generation: Bag of words

- Text document is represented by the words it contains (and their occurrences)
  - e.g., “The Lord of the rings” → {“the”, “the”, “Lord”, “rings”, “of”}
  - Order of words is not that important
  - Repitition counts

An example

中油吸收5毛漲幅，汽油可望降2元

原先預估要降1.5元的汽油，傳出經濟部要求中油繼續吸收0.5元的漲幅之後，可能可以降到2元，經濟部長也證實，吸收0.5元的漲幅，原則一直沒變，結果加油站上午冷冷清清，不然駕駛人就是只加1百、2百，要等到降價再

中油(Nc) 吸收(VC) 5(FW) 毛(Nb) 漲幅(Na) ，(COMMACATEGORY) 汽油(Na) 可望(VK) 降(VC) 2(FW) 元(Nf) 原先(D) 預估(VE) 要(D) 降(VC) 1(FW) ·(PERIODCATEGORY) 5(FW) 元(Nf) 的(DE) 汽油(Na) ，(COMMACATEGORY) 傳出(VC) 經濟部(Nc) 要求(VF) 中油(Nc) 繼續(VF) 吸收(VC) 0(FW) ·(PERIODCATEGORY) 5(FW) 元(Nf) 的(DE) 漲幅(Na) 之後(Ng) ，(COMMACATEGORY) 可能(D) 可以(D) 降到(VJ) 2(FW) 元(Nf) ，(COMMACATEGORY) 經濟部長(Na) 也(D) 證實(VE) ，(COMMACATEGORY) 吸收(VC) 0(FW) ·(PERIODCATEGORY) 5(FW) 元(Nf) 的(DE) 漲幅(Na) ，(COMMACATEGORY) 原則(Na) 一直(D) 沒(D) 變(VH) ，(COMMACATEGORY) 結果(Dk) 加油站(Nc) 上午(Nd) 冷冷清清(VH) ，(COMMACATEGORY) 不然(Cbb) 駕駛人(Na) 就是(Cbb) 只(Da) 加(VC) 1百(Neu) 、(PAUSECATEGORY) 2百(Neu) ，(COMMACATEGORY) 要(D) 等到(P) 降價(VH) 再(D) 去(D) 加油(VB) 。(PERIODCATEGORY)

An example

中油吸收5毛漲幅，汽油可望降2元

原先預估要降1.5元的汽油，傳出經濟部要求中油繼續吸收0.5元的漲幅之後，可能可以降到2元，經濟部長也證實，吸收0.5元的漲幅，原則一直沒變，結果加油站上午冷冷清清，不然駕駛人就是只加1百、2百，要等到降價

中油 吸收 5 毛 漲幅 汽油 可望 降 2 元 原先 預估 要 降 1 · 5 元 的 汽油 傳出 經濟部 要求 中油 繼續 吸收 0 · 5 元 的 漲幅 之後 可能 可以 降到 2 元 經濟部長 也 證實 吸收 0 · 5 元 的 漲幅 原則 一直 沒 變 結果 加油站 上午 冷冷清清 不然 駕駛人 就是 只 加 1百 2百 要 等到 降價 再 去 加油

An example

中油 吸收 5 毛 漲幅 汽油 可望 降 2 元 原先 預估 要 降 1 · 5 元 的 汽油 傳出 經濟部 要求 中油 繼續 吸收 0 · 5 元 的 漲幅 之後 可能 可以 降到 2 元 經濟部長 也 證實 吸收 0 · 5 元 的 漲幅 原則 一直 沒 變 結果 加油站 上午 冷冷清清 不然 駕駛人 就是 只 加 1百 2百 要 等到 降價 再 去 加油

Stop word		
原先	可以	就是
要	也	只
的	一直	等到
之後	結果	再
可能	不然	去

中油 吸收 5 毛 漲幅 汽油 可望 降 2 元 預估 降 1 · 5 元 汽油 傳出 經濟部 要求 中油 繼續 吸收 0· 5 元 漲幅 降到 2 元 經濟部長 證實 吸收 0 · 5 元 漲幅 原則 沒 變 加油站 上午 冷冷清清 駕駛人 加 1百 2百 降價 加油

Chinese stop word

Stop word	Stop word	Stop word
一 Neu 58388	我 Nh 40332	不 D 39014
了 Di 31873	他 Nh 30025	也 D 29646
就 D 29211	人 Na 24269	都 D 20403
說 VE 19625	我們 Nh 18152	你 Nh 17298
要 D 15955	會 D 14066	很 Dfa 13013
大 VH 11577	能 D 11125	著 Di 11026
她 Nh 10776	還 D 9698	可以 D 9670
最 Dfa 9416	自己 Nh 9069	來 D 8992
所 D 8869	他們 Nh 8818	兩 Neu 8692
可 D 8508	為 VG 8369	好 VH 8304
又 D 8037	將 D 7858	更 D 7298
才 Da 7266	已 D 7256	...

▲ Source: Chapter 7 Text Operations in Hsin-Hsi Chen's lecture slides

An example

中油 吸收 5 毛 漲幅 汽油 可望 降 2 元
預估 降 1 · 5 元 汽油 傳出 經濟部 要求
中油 繼續 吸收 0 · 5 元 漲幅 降到 2 元
經濟部長 證實 吸收 0 · 5 元 漲幅 原則 沒
變 加油站 上午 冷冷清清 駕駛人 加 1百 2百
降價 加油

Collocations

before	after	before	after
5 毛	5毛	2 元	2元
2 元	2元	0 · 5 元	0·5元
1 · 5 元	1·5元	沒 變	沒變
0 · 5 元	0·5元		

中油 吸收 5毛 漲幅 汽油 可望 降 2元 預估 降 1·5元 汽油 傳出 經濟部 要求 中油 繼續 吸收 0·5元 漲幅 降到 2元 經濟部長 證實 吸收 0·5元 漲幅 原則 沒變 加油站 上午 冷冷清清 駕駛人 加 1百 2百 降價 加油

An example

Term	Freq.	Term	Freq.
中油	2	降到	1
吸收	3	經濟部長	1
５毛	1	證實	1
漲幅	3	原則	1
汽油	2	沒變	1
可望	1	加油站	1
降	2	上午	1
２元	2	冷冷清清	1
預估	1	駕駛人	1
１·５元	1	只加	1
傳出	1	１百	1
經濟部	1	２百	1
要求	1	等到	1
繼續	1	降價	1
０·５元	2	加油	1

An example

Term	F.	Term	F.
中油	2	降到	1
吸收	3	經濟部長	1
5毛	1	證實	1
漲幅	3	原則	1
汽油	2	沒變	1
可望	1	加油站	1
降	2	上午	1
2元	2	冷冷清清	1
預估	1	駕駛人	1
1·5元	1	只加	1
傳出	1	1百	1
經濟部	1	2百	1
要求	1	等到	1
繼續	1	降價	1
0·5元	2	加油	1

Term	F.	Term	F.
降價	4	經濟部長	1
吸收	3	證實	1
漲幅	3	原則	1
5毛	3	沒變	1
中油	2	加油站	1
汽油	2	上午	1
2元	2	冷冷清清	1
預估	1	駕駛人	1
1·5元	1	只加	1
傳出	1	1百	1
經濟部	1	2百	1
要求	1	等到	1
繼續	1	加油	1
可望	1		

An example

d1
Term
降價
吸收
漲幅
5毛
中油
汽油
2元
…

d2
Term
沙拉油
麵粉
漲幅
降價
…

d3
Term
柴油
油電
轎車
…

dn
Term
…

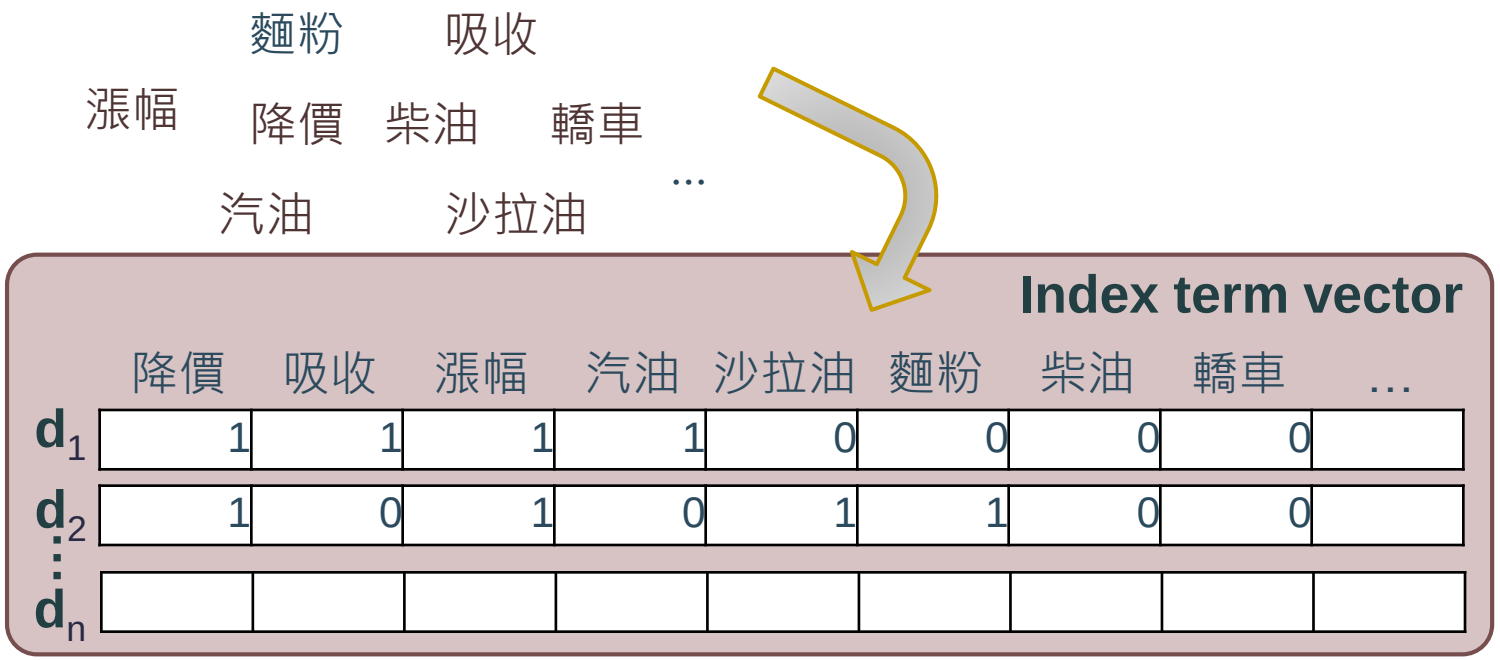
Choose selected index terms form several documents

麵粉 吸收

漲幅 降價 柴油 轎車 …

汽油 沙拉油

An example

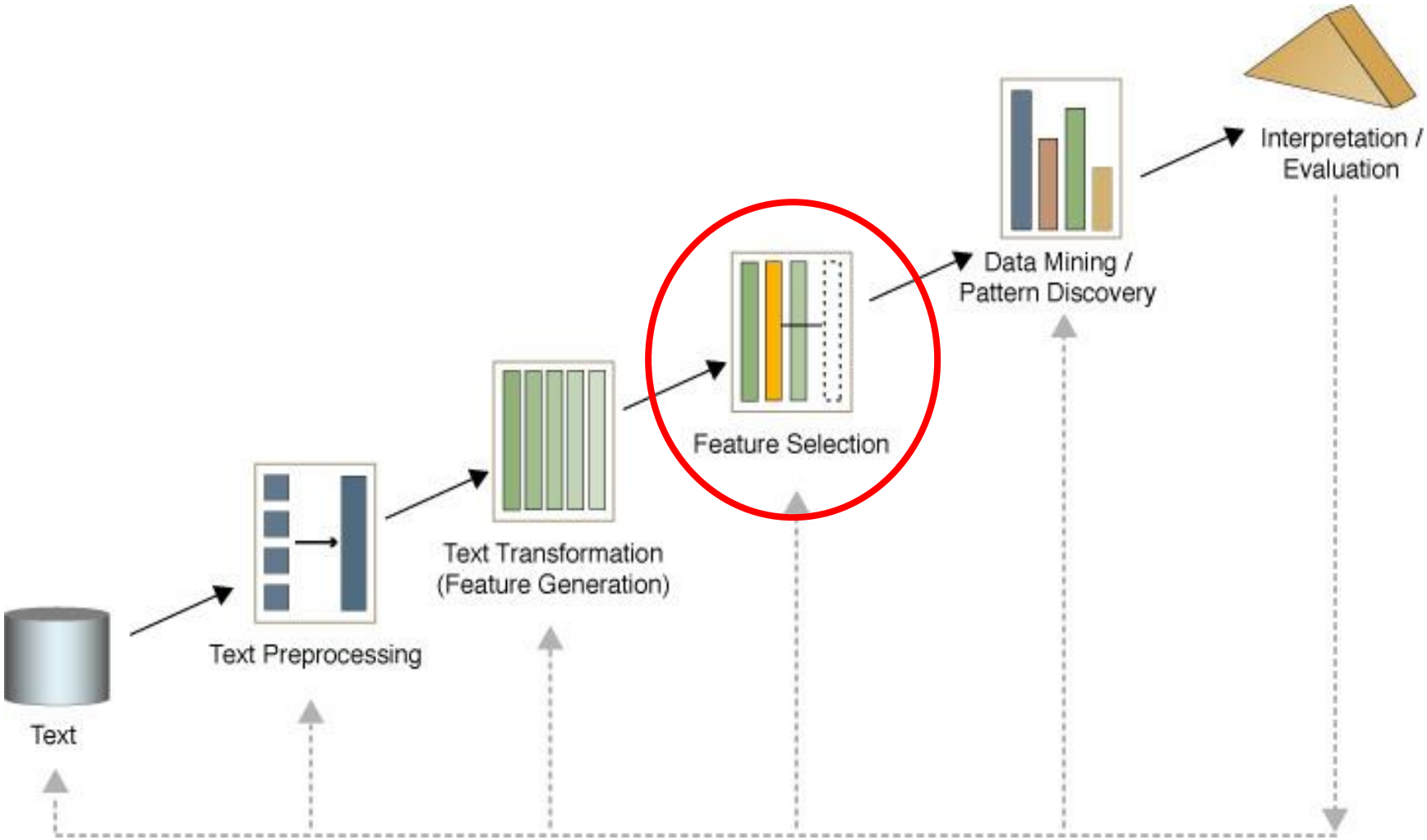


- **In vector representation**
  - **1 doc = 1 vector**
  - **Each word has its own weights**

    weights = 1, if doc1 contains word A

    weights = 0, if doc1 doesn't contain word B

Feature Selection



Feature selection

- Reduce dimensionality
  - Learners have difficulty addressing tasks with high dimensionality
- Irrelevant features
  - Not all features help!
    - e.g., the existence of a noun in a news article is unlikely to help classify it as “politics” or “sport”

Assigning Weights

- tf idf measure:
- term frequency (tf) * inverse document frequency (idf)
  - Want to weight terms highly if they are frequent in relevant documents … BUT infrequent in the collection as a whole
- Goal: assign a tf * idf weight to each term in each document

$$w_{ik} = tf_{ik} * \log(N / n_k)$$

T_k = term k in document D_i

tf_{ik} = frequency of term T_k in document D_i

idf_k = inverse document frequency of term T_k in C

N = total number of documents in the collection C

n_k = the number of documents in C that contain T_k

$$idf_k = \log\left(\frac{N}{n_k}\right)$$

$$\log\left(\frac{10000}{10000}\right) = 0$$

$$\log\left(\frac{10000}{5000}\right) = 0.301$$

$$\log\left(\frac{10000}{20}\right) = 2.698$$

$$\log\left(\frac{10000}{1}\right) = 4$$

Computing tf-idf -- An Example

Given a document containing terms with given frequencies:

A(3), B(2), C(1)

Assume collection contains 10,000 documents and document frequencies of these terms are:

A(50), B(1300), C(250)

Then:

A: tf = 3/3; idf = $\log_2(10000/50)$ = 7.6; tf-idf = 7.6

B: tf = 2/3; idf = $\log_2(10000/1300)$ = 2.9; tf-idf = 2.0

C: tf = 1/3; idf = $\log_2(10000/250)$ = 5.3; tf-idf = 1.8

- D1: "a **dog barks** at a **cat** and it **fell** from a **tree**"
- D2: "a **dog watches ants** on the **bark** of a **tree**"
- D3: "a **dog watches** another **dog watches** a **cat**"
- D4: "a **dog barks** at a **cat watches** another **cat**"
- D5: "the **bark fell** from the **tree** as a **cat watches**"

	Dog	Bark	Cat	Fell	Tree	Watch	Ant
D1	0.2	0.2	0.2	0.2	0.2	0	0
D2	0.2	0.2	0	0	0.2	0.2	0.2
D3	0.4	0	0.2	0	0	0.4	0
D4	0.2	0.2	0.4	0	0	0.2	0
D5	0	0.2	0.2	0.2	0.2	0.2	0
DF	4	4	4	2	3	4	1
IDF	0.097	0.097	0.097	0.398	0.223	0.097	0.699

TF* IDF

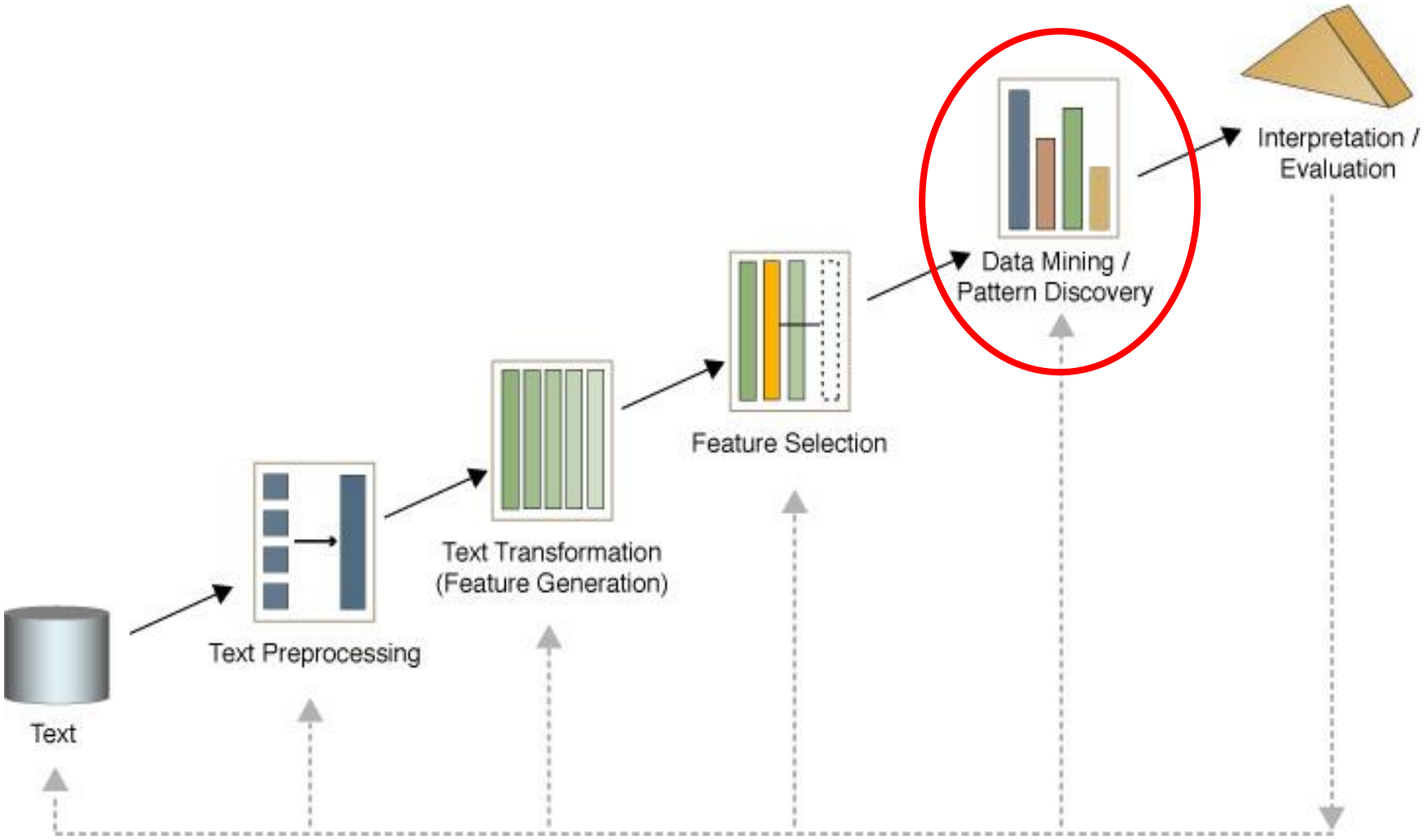
	Dog	Bark	Cat	Fell	Tree	Watch	Ant
D1	0.0194	0.0194	0.0194	0.0796	0.0446	0	0
D2	0.0194	0.0194	0	0	0.0446	0.0194	0.1398
D3	0.0388	0	0.0194	0	0	0.0388	0
D4	0.0194	0.0194	0.0388	0	0	0.0194	0
D5	0	0.0194	0.0194	0.0796	0.0446	0.0194	0

Practice 3

- R code
  - Open the file fileTM_R.txt

Text/Data Mining

- Document Similarity
- Sentiment of documents



Computing Similarity Among Documents

- Advantage of representing documents as vectors is that it facilitates computation of document similarities
- Example (Vector Space Model)
  - the dot product of two vectors measures their similarity
  - the normalization can be achieved by dividing the dot product by the product of the norms of the two vectors
  - given vectors X = <x_1, x_2, …, x_n> and Y = <y_1, y_2, …, y_n>
  - the similarity of vectors X and Y is:

$$sim(X,Y) = \frac{\sum_i (x_i \times y_i)}{\sqrt{\sum_i x_i^2 \times \sum_i y_i^2}}$$

Note: this measures the cosine of the angle between two vectors

	Dog	Bark	Cat	Fell	Tree	Watch	Ant
D1	0.0194	0.0194	0.0194	0.0796	0.0446	0	0
D2	0.0194	0.0194	0	0	0.0446	0.0194	0.1398
D3	0.0388	0	0.0194	0	0	0.0388	0
D4	0.0194	0.0194	0.0388	0	0	0.0194	0
D5	0	0.0194	0.0194	0.0796	0.0446	0.0194	0

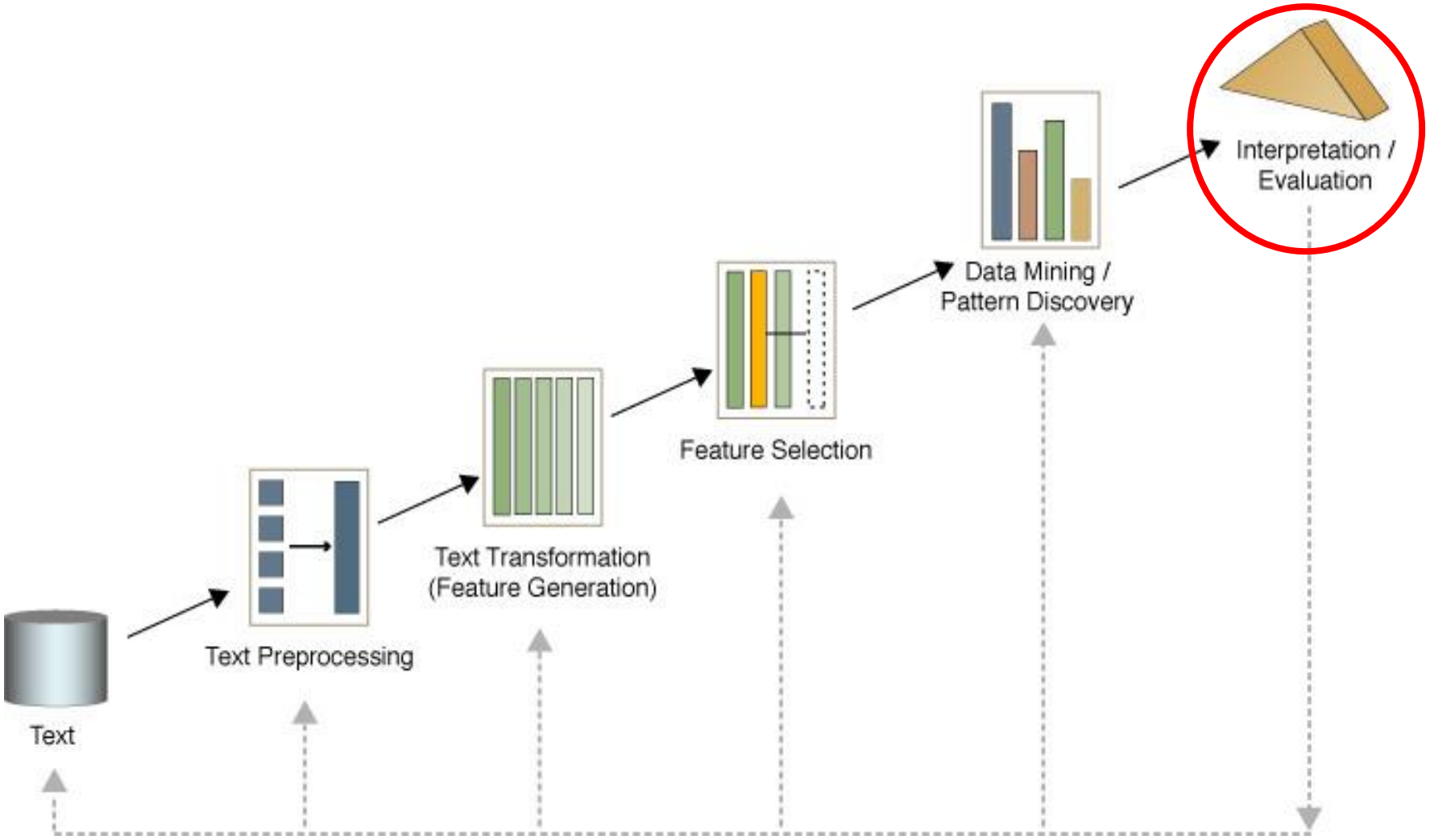
$$sim(D1,D2)=\frac{0.0194^2+0.0194^2+0.0194*0+0.0796*0+0.0446^2+0*0.0194+0*0.1398}{\sqrt{(0.0194^2+0.0194^2+0.0194^2+0.0796^2+0.0446^2)*(0.0194^2+0.0194^2+0.0446^2+0.0194^2+0.1398^2)}}$$

$$=0.29$$

Practice 4

- R code
  - Open the file fileTM_R.txt

Analyzing results



Outline

- What is text mining
- Sources of Text
- Text Preprocessing
- Feature Generation
- Feature Selection
- Text Data mining
- Analyzing Result
- **Topic Identification**

範例一(Example1)

- 英文text mining，五篇文章
- D1:A dog barks at a cat and it fell from a Tree.
- D2:A dog watches ants on the bark of a Tree.
- D3:A dog watches another dog watches a Cat.
- D4:A dog barks at a cat watches another Cat.
- D5:The bark fell from the tree as a cat Watches.

	dog	bark	cat	fell	tree	watch	ant
D1	1	1	1	1	1	0	0
D2	1	1	0	0	1	1	1
D3	2	0	1	0	0	2	0
D4	1	1	2	0	0	1	0
D5	0	1	1	1	1	1	0
df							
idf							

範例一(Example1)

- D1: "a **dog barks** at a **cat** and it **fell** from a **tree**"
- D2: "a **dog watches ants** on the **bark** of a **tree**"
- D3: "a **dog watches** another **dog watches** a **cat**"
- D4: "a **dog barks** at a **cat watches** another **cat**"
- D5: "the **bark fell** from the **tree** as a **cat watches**"

dog在D1的tfidf值

0.0194

	dog	bark	cat	fell	tree	watch	ant
D1	0.2	0.2	0.2	0.2	0.2	0	0
D2	0.2	0.2	0	0	0.2	0.2	0.2
D3	0.4	0	0.2	0	0	0.4	0
D4	0.2	0.2	0.4	0	0	0.2	0
D5	0	0.2	0.2	0.2	0.2	0.2	0
df	4	4	4	2	3	4	1
idf	0.097	0.097	0.097	0.398	0.223	0.097	0.699

tf

idf

tf-idf

dog在D1的tf值

	dog	bark	cat	fell	tree	watch	ant
D1	0.2	0.2	0.2	0.2	0.2	0	0
D2	0.2	0.2	0	0	0.2	0.2	0.2
D3	0.4	0	0.2	0	0	0.4	0
D4	0.2	0.2	0.4	0	0	0.2	0
D5	0	0.2	0.2	0.2	0.2	0.2	0
df	4	4	4	2	3	4	1
idf	0.097	0.097	0.097	0.398	0.223	0.097	0.699

相乘

dog的idf值

dog的tf-idf值

	dog	bark	cat	fell	tree	watch	ant
D1	0.0194	0.0194	0.0194	0.0796	0.0446	0	0
D2	0.0194	0.0194	0	0	0.0446	0.0194	0.1398
D3	0.0388	0	0.0194	0	0	0.0388	0
D4	0.0194	0.0194	0.0388	0	0	0.0194	0
D5	0	0.0194	0.0194	0.0796	0.0446	0.0194	0

R有預設的tf-idf公式(有加權過不太一樣)

- 就是R程式碼中的DTM_Rtfidf 變數內容

Terms							
Docs	ant	bark	cat	dog	fell	tree	watch
1	0.0000000	0.06438562	0.06438562	0.06438562	0.2643856	0.1473931	0.00000000
2	0.4643856	0.06438562	0.00000000	0.06438562	0.0000000	0.1473931	0.06438562
3	0.0000000	0.00000000	0.06438562	0.12877124	0.0000000	0.0000000	0.12877124
4	0.0000000	0.06438562	0.12877124	0.06438562	0.0000000	0.0000000	0.06438562
5	0.0000000	0.06438562	0.06438562	0.00000000	0.2643856	0.1473931	0.06438562

R的預設tfidf計算方法的結果與DM理論上有差距

	dog	bark	cat	fell	tree	watch	ant
D1	0.0194	0.0194	0.0194	0.0796	0.0446	0	0
D2	0.0194	0.0194	0	0	0.0446	0.0194	0.1398
D3	0.0388	0	0.0194	0	0	0.0388	0
D4	0.0194	0.0194	0.0388	0	0	0.0194	0
D5	0	0.0194	0.0194	0.0796	0.0446	0.0194	0

範例一(Example1)

- myDTM
- **Terms**

Docs	bark	cat	dog	fell	tree	ant	watch
1	0.019382	0.01938200	0.01938200	0.079588	0.04436975	0.000000	0.00000000
2	0.019382	0.00000000	0.01938200	0.000000	0.04436975	0.139794	0.01938200
3	0.000000	0.01938200	0.03876401	0.000000	0.00000000	0.000000	0.03876401
4	0.019382	0.03876401	0.01938200	0.000000	0.00000000	0.000000	0.01938200
5	0.019382	0.01938200	0.00000000	0.079588	0.04436975	0.000000	0.01938200

myDTM是透過R程式寫出來的，其結果跟理論上一樣

	dog	bark	cat	fell	tree	watch	ant
	0.0194	0.0194	0.0194	0.0796	0.0446	0	0
	0.0194	0.0194	0	0	0.0446	0.0194	0.1398
	0.0388	0	0.0194	0	0	0.0388	0
D4	0.0194	0.0194	0.0388	0	0	0.0194	0
D5	0	0.0194	0.0194	0.0796	0.0446	0.0194	0

範例二(Example2)

- D1:The sky is 1-blue.
- D2:The 2sun is bright today.
- D3:The sun in the sky is 3*bright.
- D4:We can see the 4/shining sun, the bright sun.

範例三(Example3)

- D1:Data Mining and(&) Social Media Mining.
- D2:Social 3Network Analysis.
- D3:Data 4-Mining.
- D4:Big Data and Data Science.

範例四(Example4)

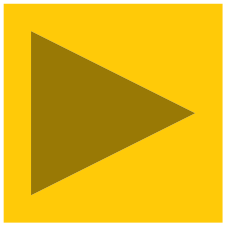
- D1:Two for tea and tea for two.
- D2:Tea for me and tea for you.
- D3:You for me and me for you.

Computing Similarity Among Documents

- Advantage of representing documents as vectors is that it facilitates computation of document similarities
- Example (Vector Space Model)
  - the dot product of two vectors measures their similarity
  - the normalization can be achieved by dividing the dot product by the product of the norms of the two vectors
  - given vectors X = <x_1, x_2, …, x_n> and Y = <y_1, y_2, …, y_n>
  - the similarity of vectors X and Y is:

$$sim(X,Y) = \frac{\sum_i (x_i \times y_i)}{\sqrt{\sum_i x_i^2 \times \sum_i y_i^2}}$$

Note: this measures the cosine of the angle between two vectors



相似度(Similarity)

$$sim(X,Y)=\frac{\sum_i (x_i \times y_i)}{\sqrt{\sum_i x_i^{\,2} \times \sum_i y_i^{\,2}}}$$

	Dog	Bark	Cat	Fell	Tree	Watch	Ant
D1	0.0194	0.0194	0.0194	0.0796	0.0446	0	0
D2	0.0194	0.0194	0	0	0.0446	0.0194	0.1398
D3	0.0388	0	0.0194	0	0	0.0388	0
D4	0.0194	0.0194	0.0388	0	0	0.0194	0
D5	0	0.0194	0.0194	0.0796	0.0446	0.0194	0

$$sim(D1,D2)=\frac{0.0194^2+0.0194^2+0.0194*0+0.0796*0+0.0446^2+0*0.0194+0*0.1398}{\sqrt{(0.0194^2+0.0194^2+0.0194^2+0.0796^2+0.0446^2)*(0.0194^2+0.0194^2+0.0446^2+0.0194^2+0.1398^2)}}$$

$$=0.29$$

有沒有辦法可以縮小這一張表格呢?

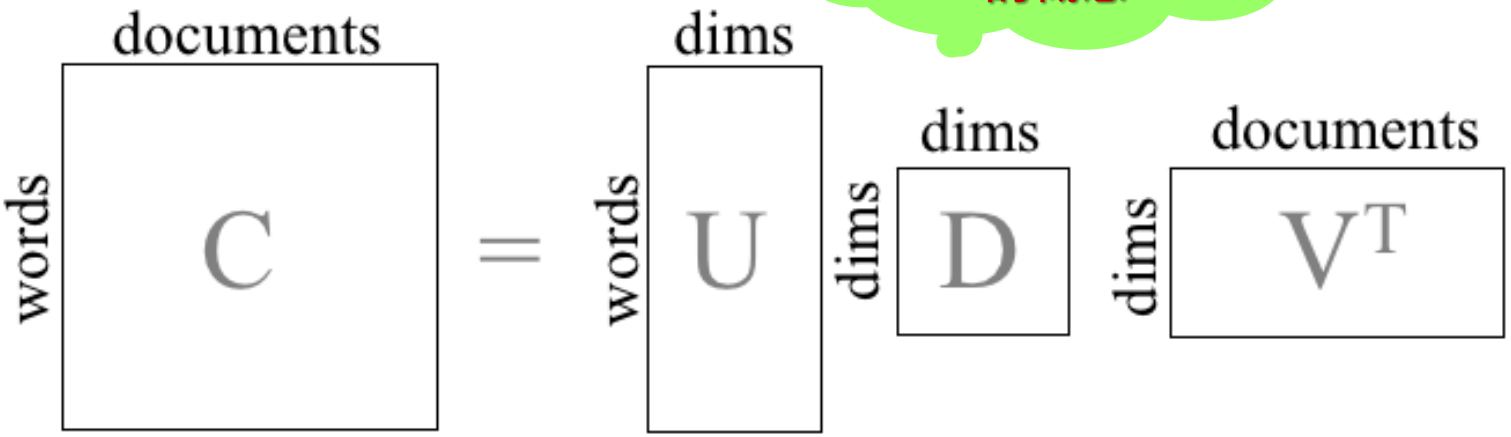
	dog	bark	cat	fell	tree	watch	ant	….
D1	0.0194	0.0194	0.0194	0.0796	0.0446	0	0	
D2	0.0194	0.0194	0	0	0.0446	0.0194	0.1398	
D3	0.0388	0	0.0194	0	0	0.0388	0	
D4	0.0194	0.0194	0.0388	0	0	0.0194	0	
D5	0	0.0194	0.0194	0.0796	0.0446	0.0194	0	
…								

假如有幾萬篇文章，則欄位會變得太多，實務上有些難處理。

Latent Semantics Analysis

想像dims是段落(paragraph)的概念

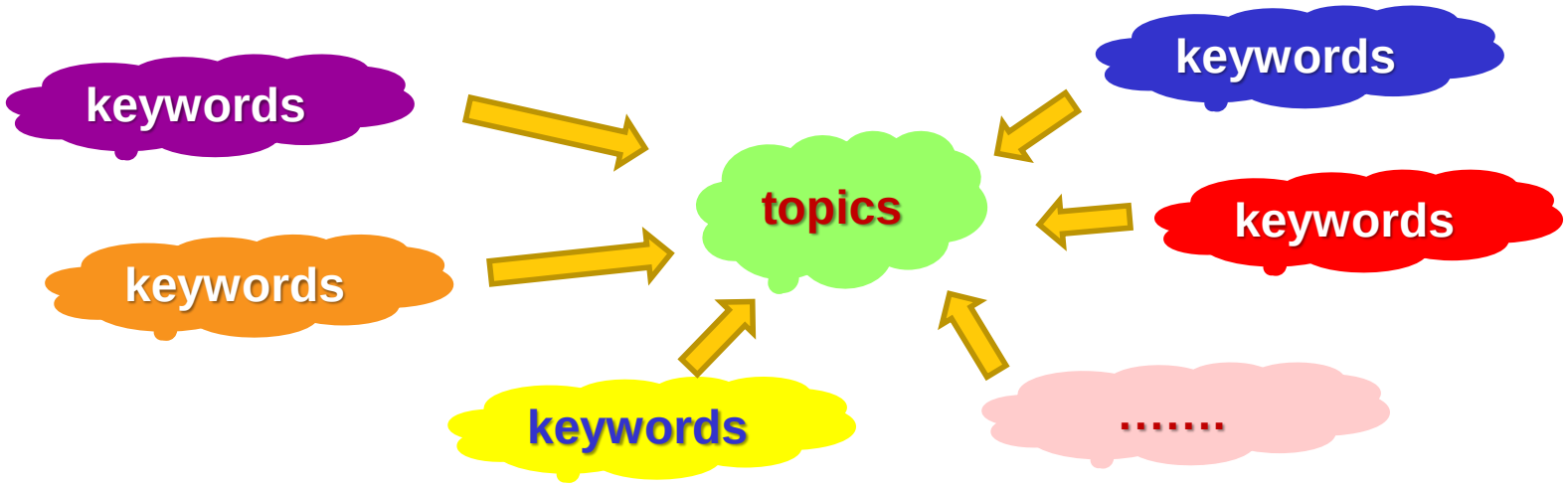
LSA



奇異值分解法 (singular value decomposition，SVD)

Keywords Vs. Topic

	dog	bark	cat	fell	tree	watch	ant
D1	0.0194	0.0194	0.0194	0.0796	0.0446	0	0
D2	0.0194	0.0194	0	0	0.0446	0.0194	0.1398
D3	0.0388	0	0.0194	0	0	0.0388	0
D4	0.0194	0.0194	0.0388	0	0	0.0194	0
D5	0	0.0194	0.0194	0.0796	0.0446	0.0194	0



範例一(Example1)

- 英文text mining，五篇文章
- D1:A dog barks at a cat and it fell from a Tree.
- D2:A dog watches ants on the bark of a Tree.
- D3:A dog watches another dog watches a Cat.
- D4:A dog barks at a cat watches another Cat.
- D5:The bark fell from the tree as a cat Watches.

Find the topics directly.

	dog	bark	cat	fell	tree	watch	ant
D1	0.0194	0.0194	0.0194	0.0796	0.0446	0	0
D2	0.0194	0.0194	0	0	0.0446	0.0194	0.1398
D3	0.0388	0	0.0194	0	0	0.0388	0
D4	0.0194	0.0194	0.0388	0	0	0.0194	0
D5	0	0.0194	0.0194	0.0796	0.0446	0.0194	0

Selecting Topics directly

- Representing Concept with words require too many features
- Animals: Dog, Cat, Ant

	Dog	Bark	Cat	Fell	Tree	Watch	Ant
D1	0.0194	0.0194	0.0194	0.0796	0.0446	0	0
D2	0.0194	0.0194	0	0	0.0446	0.0194	0.1398
D3	0.0388	0	0.0194	0	0	0.0388	0
D4	0.0194	0.0194	0.0388	0	0	0.0194	0
D5	0	0.0194	0.0194	0.0796	0.0446	0.0194	0

- Action: Fell, watch
- Plant : Tree

Assumption

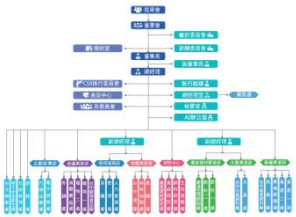
另外有一群學者認為我們不要再找keywords直接找topics

- **A document is related to only limited number of topics**
  - (很少有人寫一篇文章會包含30個topics，所以正常人文章沒以幾個topics)
- **Given a topic, a word is chosen from a limited number of vocabularies in a topic**
  - (大部分的人會先決定大部分的人會先決定topic然後才開始寫，譬如topic為環保Environmental protection，接著所使用的字會環繞著這個topic有關的字，不會超出。)
- When writing a document, the author thinks of topics before writing a word. When the topic is decided, a vocabulary is then picked

有沒有辦法直接去計算topic呢?

Detect Topics directly

邏輯上已經整理好兩個主題
Given two topics



TOPIC 1

organize
lead
committee
direct

TOPIC 2

mercury
pollute
arsenic
lead

© Can Stock Photo

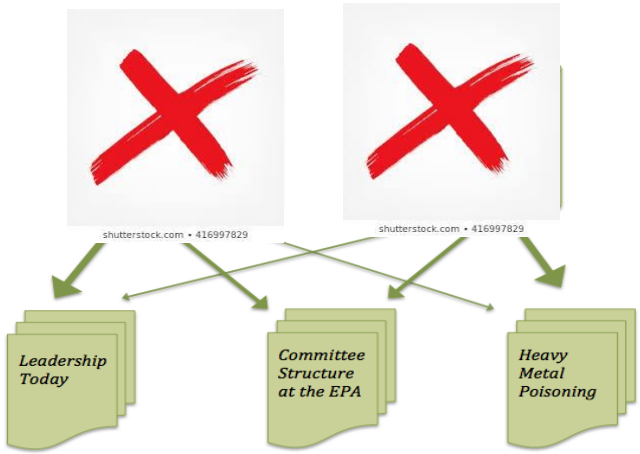
Leadership Today

Committee Structure at the EPA

Heavy Metal Poisoning

成立一個委員會來處理汙染

- However, we don’t know what are the topics and the exact distribution of documents to topics and topics to words



guess…guess…guess…

Key point:
從document回頭推算有哪些topics，而那些topics包含那些words

- If you find a **"lead"** in a document, how do you guess the topic?
  - Organization
  - **Poisonous**

- How often does lead related to organizations in other occurrence
- How common is topic organization in the rest of the document

$$P(Z|W,D) = \frac{\#\ of\ word\ W\ in\ topic\ Z + \beta_w}{total\ tokens\ in\ Z + \beta} * (\#\ words\ in\ D\ that\ belong\ to\ Z + \alpha)$$

至少一千篇文章，則判讀會比較準

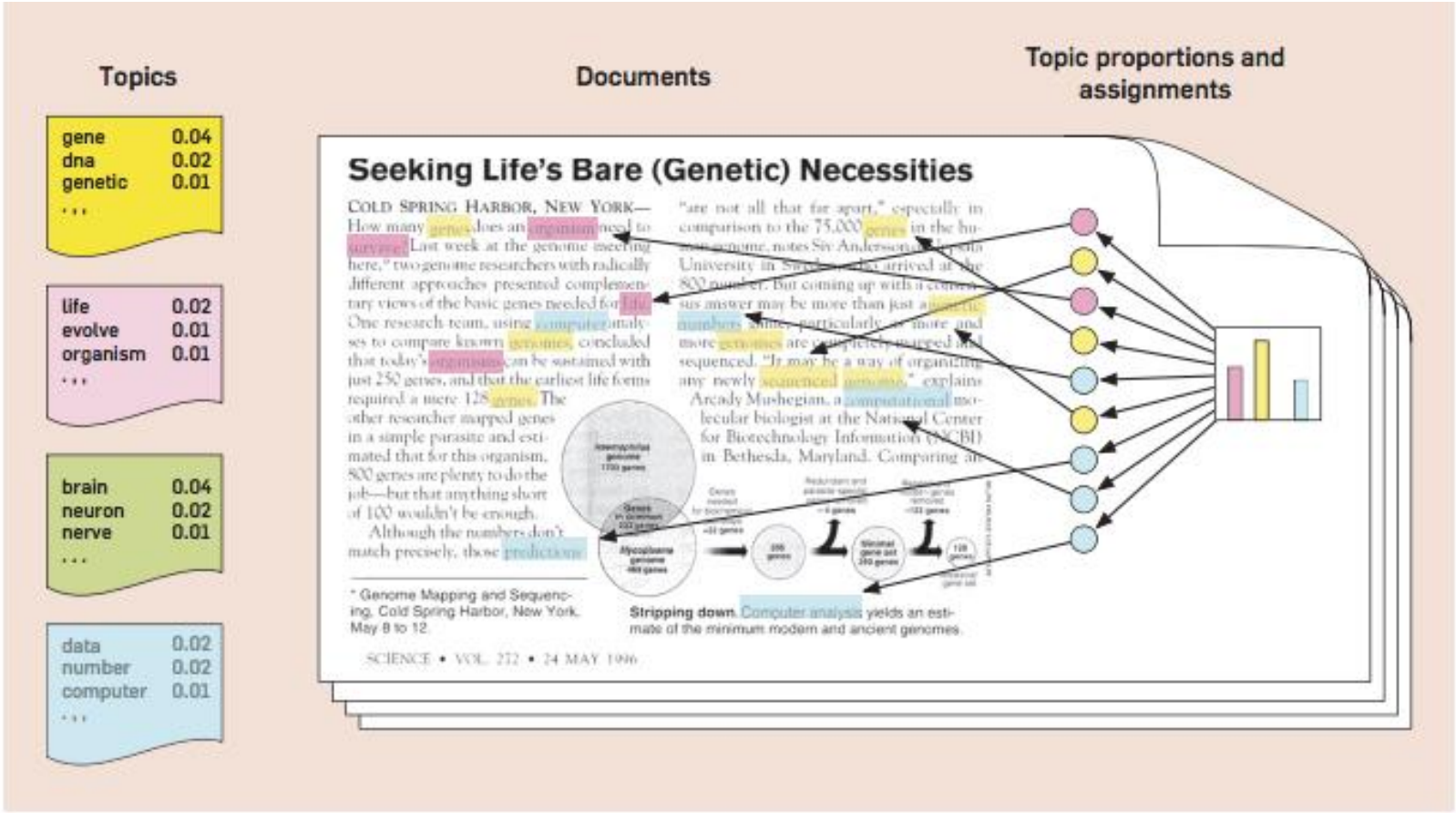
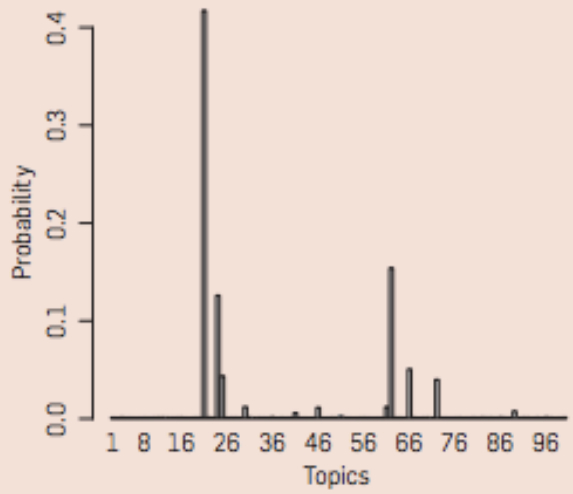
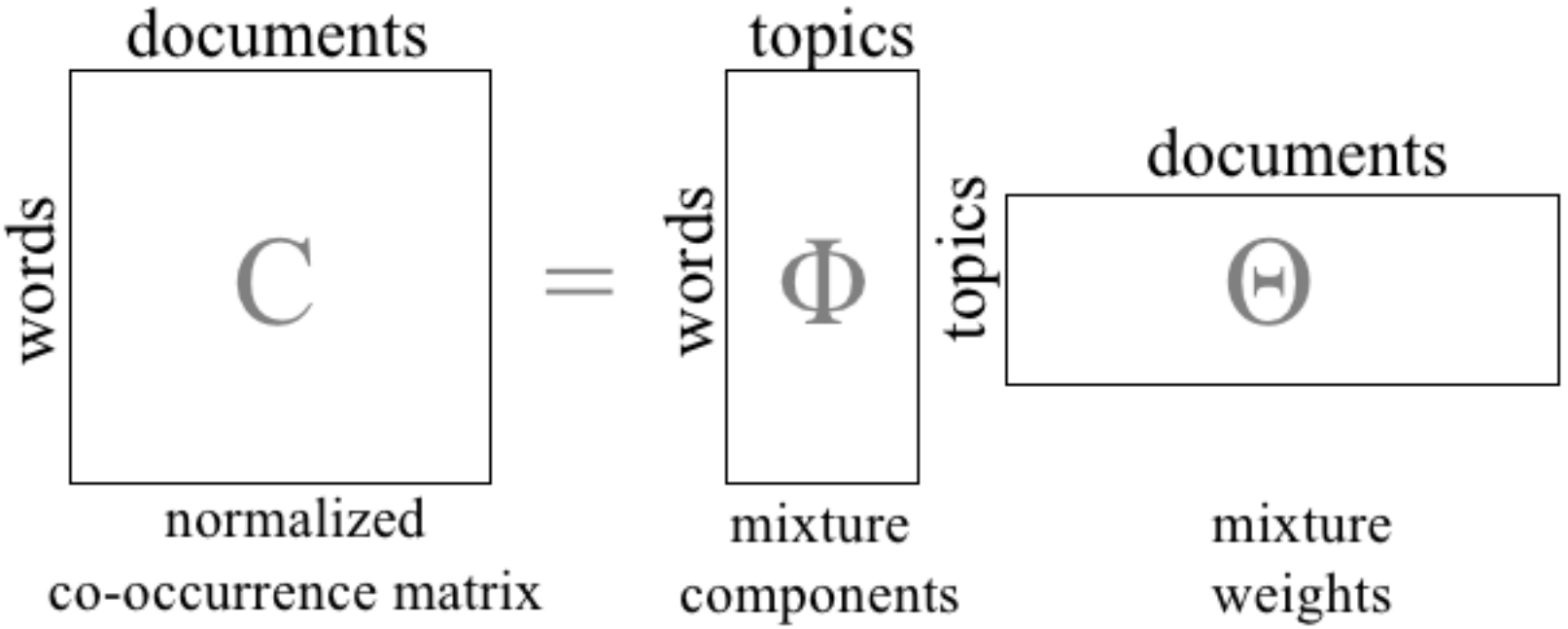


Figure 2. Real inference with LDA. We fit a 100-topic LDA model to 17,000 articles from the journal *Science*. At left are the inferred topic proportions for the example article in Figure 1. At right are the top 15 most frequent words from the most frequent topics found in this article.



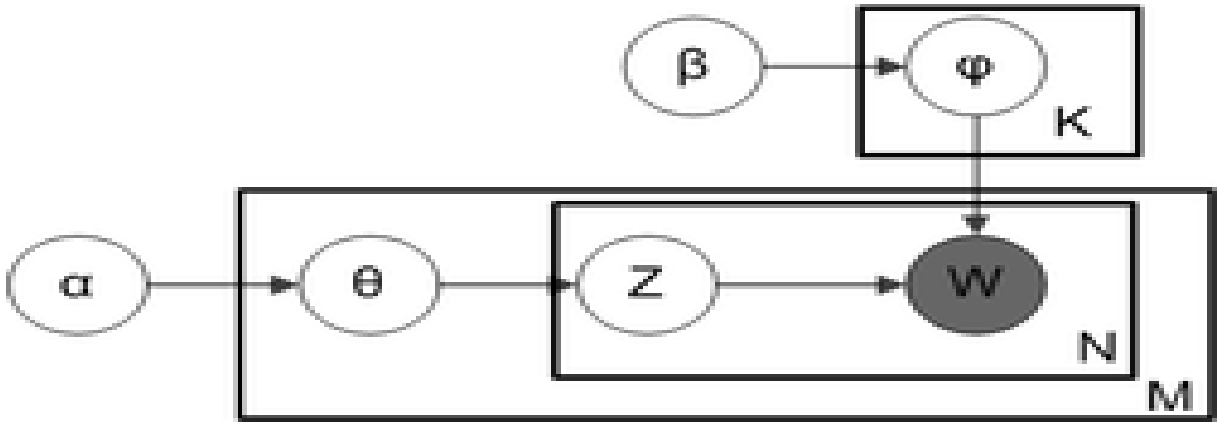
"Genetics"	"Evolution"	"Disease"	"Computers"
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

Topic Model


documents
words
C
normalized
co-occurrence matrix
=
topics
words
Φ
mixture
components
documents
topics
Θ
mixture
weights

LDA(Latent Dirichlet Allocation)

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})$$
$$= \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \left(\prod_{n=1}^{N} p(z_{d,n} \mid \theta_d) p(w_{d,n} \mid \beta_{1:K}, z_{d,n}) \right)$$



$$\propto \left(n_{m,(\cdot)}^{k,-(m,n)} + \alpha_k\right) \frac{n_{(\cdot),v}^{k,-(m,n)} + \beta_v}{\sum_{r=1}^{V} n_{(\cdot),r}^{k,-(m,n)} + \beta_r}$$

$$P(Z|W,D) = \frac{\#\ of\ word\ W\ in\ topic\ Z + \beta_w}{total\ tokens\ in\ Z + \beta} * (\#\ words\ in\ D\ that\ belong\ to\ Z + \alpha)$$

Conclusion

- TF-IDF is the most commonly utilized method to identify keywords.
- To reduce the number of Keywords, Topic models are popular
- Among all the topic models, LSA and LDA are the most popular
- Chinese language posts new challenge to text mining due to its word ambiguity and grammar.

FROM ZERO TO HERO IN R

~Go Go Go Fighting~