

# Data Mining Final Project- Friend Mining Based on Facebook

B96502132 Pei-Hao Su (蘇培豪)

R00921033 Tsung-Hsien Wen (溫宗憲)

R00942075 Yu-Yu Chou (周宥宇)

## I. Introduction

近幾年來，在「社群網站」急速的發展之下，透過網路介面使人們更容易認識、接觸到不同的人、加入不同的社群認識新朋友進而得到新資訊。而社群網站對各功能的整合也使得使用者不再需要下載一堆的軟體程式，透過社群網站整合的介面與功能，社群網站成為了當今人們不可或缺的一項工具。正因此，各種研究油然而生，如社群網站上的摘要、即時動態、介面改良...等等，各有各的奧秘以及值得研究之內容。

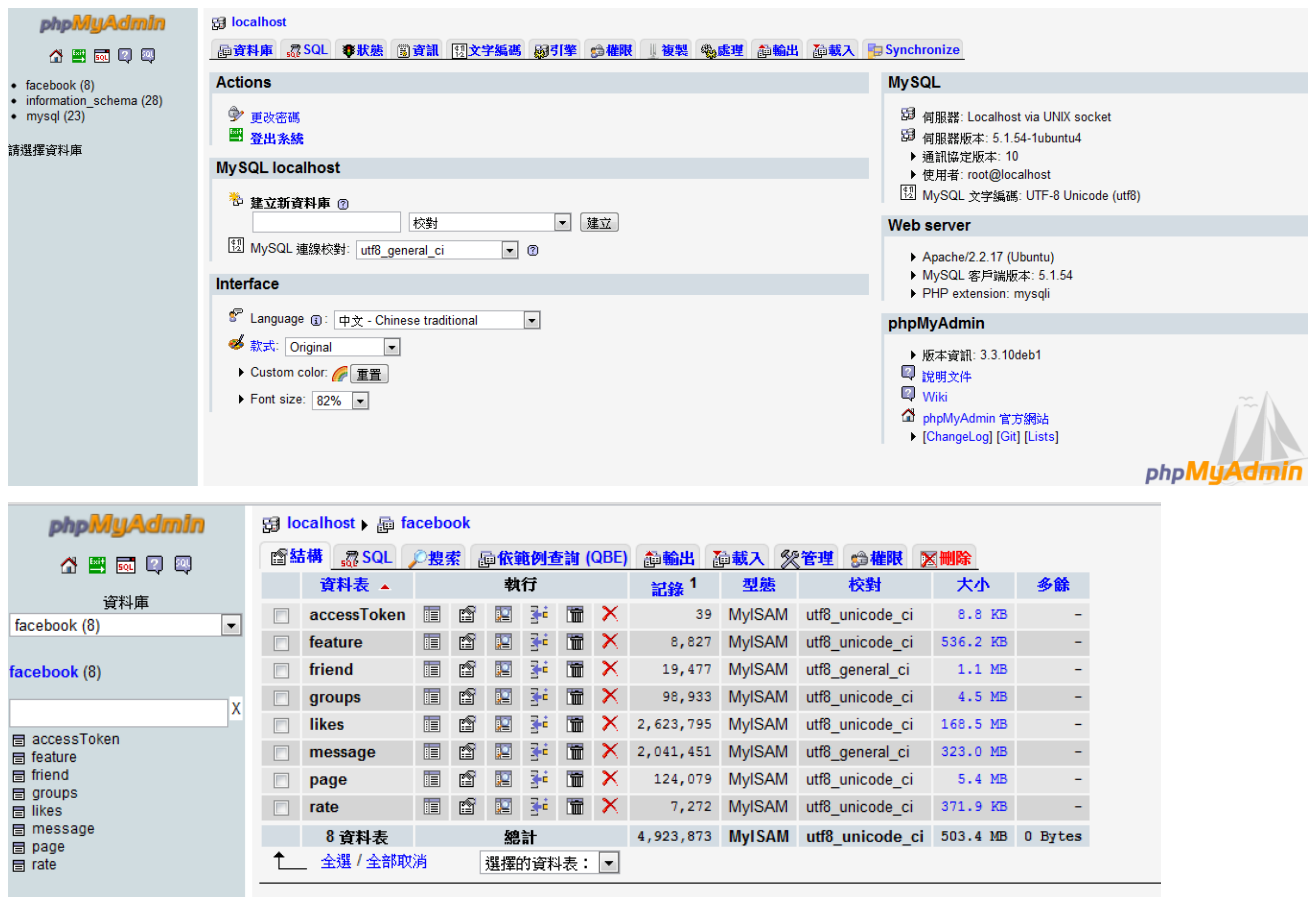
本專案專注於所謂的“friend recommendation system”，如同字面上所述，要如何在上千萬上億的使用者中推薦給使用者與其興趣相投、潛在的好友是一門相當大的學問。社群網站上的資料量相當龐大，共同好友、互相評論、使用字句...等等都是可用的資訊，在過去的好友推薦系統中，或可利用簡單的共同好友之間的聯結或共同社團等等的資訊做出一套簡單的好友推薦系統。這樣的推薦方式儘管十分簡單，對於擁有許多使用者資料的現存社群網站來說卻是一個十分有效，並且能幫助使用者找到自己朋友的工具。這類型的推薦機制建立在社群網站的推薦功能只是幫使用者在茫茫的廣大網路世界中，找到他們已經有所連結的朋友，而非推薦一個原本可能毫無關係，但可能潛在性興趣相仿，臭味相投的陌生人彼此認識，對使用者本身擴大交友圈的功能並不大。因此我們這次研究的專題聚焦在如何根據兩人在網路世界的潛在用字用語，以及他們所互相關心的一些文章主題，分析並推薦兩者互相認識與結交，進而增進網路世界資訊的互相分享與傳遞，並提供現有設群網站及交友網站更多的可能性。

我們這次的專題則在好友推薦系統中加入了 **Probability Latent Semantic Analysis (PLSA)** 這項技術所能提供的潛藏主題分析功能並同時考慮使用者語彙上的相似程度。**PLSA** 是用

期望最大化演算法(EM)，算出所謂的淺藏式資訊。每個使用者我們利用 **PLSA** 根據知名社群網站 **Facebook** 上的諸多資訊予以分析與研究，建立人與人之間所關心的潛在主題模型，並利用此模型以及其所用的語彙做與其他所有使用者比對其相似度。實驗證明這樣的推薦模式在我們所提資訊不足的好友推薦情境下能達到比純用語言分析更好的結果。而透過淺藏的資訊以及使用過的歷史字串將可以推薦使用者更加適合，更合的來的人作為朋友。以下第二、三部分將介紹實驗中所用的蒐集之資料以及使用之特徵(features)，第四部分介紹實驗中所用的兩種模型，第五部份介紹實驗中檢索系統所用的資料，第六部分介紹整體實驗的架構、七部分介紹評估的方法、第八、第九部分為實驗結果與結論。

## II. Data collection

為了擷取社群網站 Facebook 上有用的資訊，我們閱讀了 Facebook 所提供之 api，並利用實驗室電腦架設了一個 PHP-based 資料擷取伺服器，網站頁面如下圖所示，網頁有各種讀取以及存取功能，能對資料作即時更新，並選擇儲存為我們想要之格式。我們將各種資料作適當分類，為避免侵犯受測者之隱私，我們請 20 位自願的同學們登入帳號擷取他們的資料。由於 Facebook 網站的限制與條款，我們只能窺測到屬於自願使用者能觀察到的網路資訊，這樣不完整的資訊我對我們的研究增加一些困難。擷取到的資料整理如下，這裡面包含同學們參與的群組、粉絲頁、貼文資訊、共同好友、Like 資料、訊息...等等，也包含各個同學們好友的貼文和各種資訊。



The top screenshot shows the phpMyAdmin interface for MySQL localhost. The left sidebar lists databases: facebook (8), information\_schema (28), and mysql (23). The main panel shows the 'Actions' menu with options like '更改密碼' and '登出系統'. The 'MySQL localhost' section includes fields for '建立新資料庫' and 'MySQL 連線校對'. The 'Interface' section shows language settings (中文 - Chinese traditional) and font size (82%). The right sidebar displays server information: MySQL version 5.1.54-1ubuntu4, user root@localhost, and web server details (Apache/2.2.17, MySQL client version 5.1.54, PHP extension mysql).

The bottom screenshot shows the 'facebook' database selected. The table list is displayed with columns: 資料表, 執行, 記錄, 型態, 校對, 大小, 多餘. The table list includes:

資料表	執行	記錄	型態	校對	大小	多餘
accessToken		39	MyISAM	utf8_unicode_ci	8.8 KB	-
feature		8,827	MyISAM	utf8_unicode_ci	536.2 KB	-
friend		19,477	MyISAM	utf8_general_ci	1.1 MB	-
groups		98,933	MyISAM	utf8_unicode_ci	4.5 MB	-
likes		2,623,795	MyISAM	utf8_unicode_ci	168.5 MB	-
message		2,041,451	MyISAM	utf8_general_ci	323.0 MB	-
page		124,079	MyISAM	utf8_unicode_ci	5.4 MB	-
rate		7,272	MyISAM	utf8_unicode_ci	371.9 KB	-
8 資料表	總計	4,923,873	MyISAM	utf8_unicode_ci	503.4 MB	0 Bytes

### III. Features

所擷取的特徵參數如下所示，而用途則如右下圖所示。

#### 1. User

共同好友數，即為兩使用者好友的交集。

#### 2. Likes

為對某使用者對另一貼文按 Like 之 feature。

#### 3. Comments

使用者回覆某篇貼文之 feature。

#### 4. Fan Page

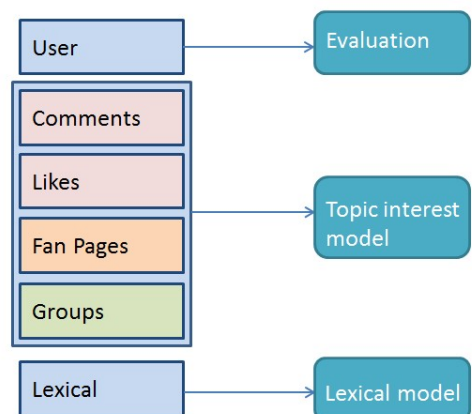
粉絲頁，使用者喜愛的偶像專頁。

#### 5. Group

社團群組，使用者參與的社團

#### 6. Lexical

此為每一使用者平常所用之語彙資訊。



👍 7 people like this.



**朱建宇** 好問題。

about an hour ago · Like · 👍 1



**Weijung Chen** 要去的話請call我，翹班都得去

about an hour ago · Like · 👍 1

You and 陳筠貞



👍 82 friends like this.



English Conversation class



台大南友會

20+



New Group...



Wall

Info

Friend Activity

Photos

**241,054**

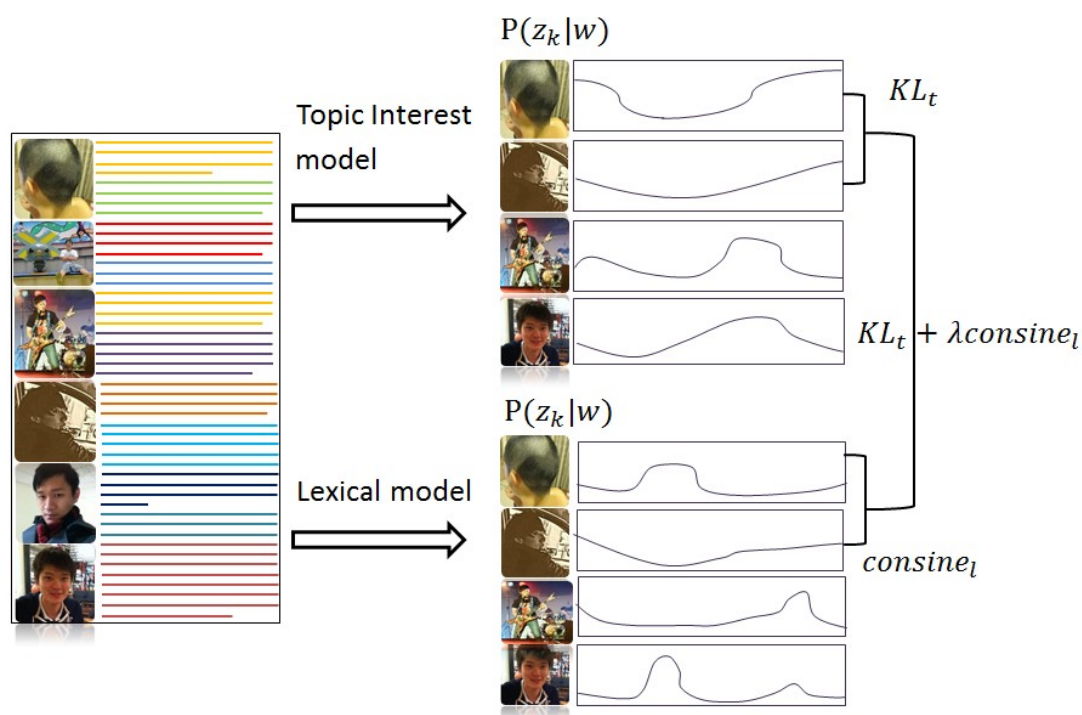
like this

**577**

talking about this

## IV. Model

實驗整體架構圖如下，主要分為兩大部分為 Topic interest model 以及 Lexical model，最後每個使用者對其他所有使用者作相似度比對並且結合兩分數，按造分數高低推薦好友，個別 model 詳細介紹如下：

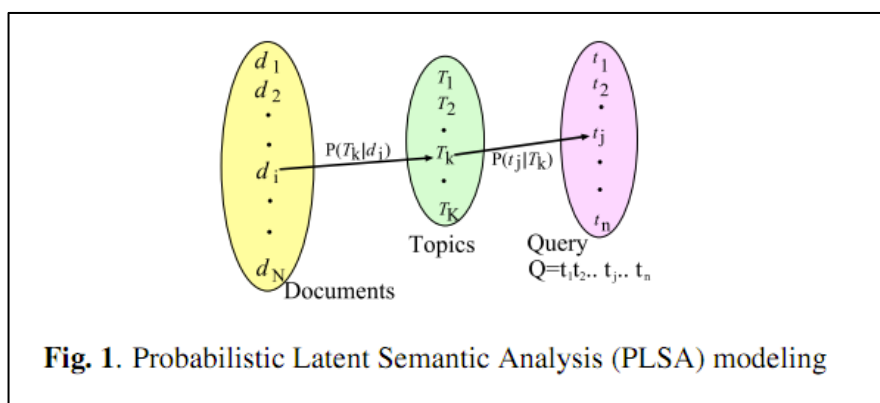


### 1. Topic interest model

#### Probability Latent Semantic Analysis (PLSA)

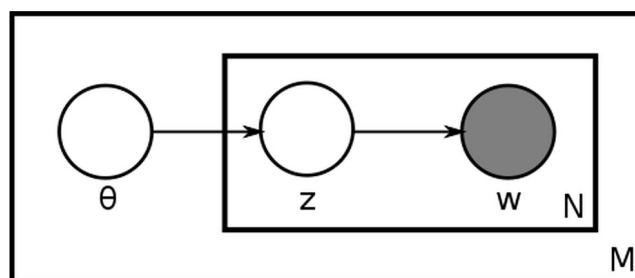
PLSA 傳統作法是個將 EM 演算法運用在文字資料上的技術，在有很多的文件 (document)，每個文件中有若干個文字 (term)，PLSA 假設這每一篇的文件產生的方式是先從給定的主題 (Topic) 中選出其中一個主題，再從該主題對應的詞典中選出一個文字 (term)，按造此循環做  $N$  次而產生一篇  $N$  個 terms 的文件 (如下圖所示)。PLSA 則是在此假設之下運用 EM 演算法，對每一個文件求出給定文件產生主題的機率  $P(T|d)$  和對每一個主題產生某個 term 的機率  $P(t|T)$ ，而透過 EM 演算法這兩組機率會逼近所有文件組成的 set。而本實驗不同於以往用文字上的資訊，我們利用社群網站上各種不同的 features 代替文件，且每個人則當作一個 term 去表示。我們將每位使用者所張貼之訊息當作一個文件，

而曾經在此訊息互動之使用者 id 作為 **terms** 形成一組文件組。再將所有社團每一個當作一個文件，且有參與該社團中的使用者 id 當作 **terms** 形成第二組文件組。還有社群網站中的粉絲頁也當作一個文件，參與該粉絲頁的使用者 id 當作 **terms** 形成第三組文字組。將三組文字組全部結合在一起 **train** 出一組 **PLSA**，最後對每位使用者則可知道給定使用者 (**terms**)的情況下出現某個主題(**topic**)的機率分布圖形。



上圖左為文件(documents)中為主題(topic)，右則為文字(term)

下圖也為 PLSA 之示意圖



N : 文字數量

M : 文件數量

w : 文字(term)

z : 主題(topic)

θ : 文件(document)

## 相似度計算 (KL divergence)

在經過 PLSA 的 training 步驟之後，我們對兩兩使用者必須計算其淺藏意涵中的相似程度。而這裡我們使用了 KL divergence 計算兩兩特徵向量的相似程度，KL divergence 的式子為  $D_{KL}(P||Q) = \sum_i \log \frac{P(i)}{Q(i)}$ 。P、Q 分別為兩特徵向量，而  $P(i)$ 、 $Q(i)$  則為特徵向量中第  $i$  項的值。KL divergence 類似於亂度的概念，若兩特徵向量越相近則算出來的 KL divergence 的值應該越小。此外 KL divergence 多用於計算兩特徵向量 P、Q 個別的特徵參數總和為 1 且不為負號。

## 2. Lexical model

語彙模型上，對每一位使用者我們蒐集他過去所評論、張貼之貼文所使用的語言。但若單純用每一個文字所出現的次數形成一組特徵向量(feature vector)，往往會有一些問題，像是不重要的文字如「的」、「我」、「你」...這些所謂的 function word 往往不代表有什麼意義，但是出現的頻率卻特別高。因此我們使用了一個再處理文字資訊上常用的方法，就是對每一個文字計算其 term frequency (TF)以及 inverse document frequency (IDF)，式子列如下。 $n_{i,j}$  是該詞在文件  $d_j$  中的出現次數，而分母則是在文件  $d_j$  中所有字詞的出現次數之和。 $|D|$ ：語料庫中的文件總數， $|\{j: t \in d_j\}|$ ：包含詞語  $t_i$  的文件數目（即  $n_{i,j} \neq 0$  的文件數目）如果該詞語不在語料庫中，就會導致被除數為零，因此一般情況下使用  $1 + |\{j: t \in d_j\}|$ 。而對每一個人  $j$  所使用過的每一個文字  $i$  就可以計算其  $tf_{i,j}idf_{i,j}$ 。因此每一個人就可以用一個由各個文字  $tf_{i,j}idf_{i,j}$  所形成的 feature vector 所表示。

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad idf_{i,j} = \log \frac{|D|}{|\{j: t \in d_j\}|}$$

## 相似度計算(Cosine similarity)

Cosine similarity 用於計算兩向量之相似程度，式子為 $\cos(\theta) = \frac{A \cdot B}{|A||B|}$ 。A、B分別為兩向特徵向量，在這個實驗中就是使用者語彙上 TF-IDF 所形成的特徵向量，若兩向量相似程度越高，則所算出之 cosine similarity 的分數則越高，物理意義上就是兩向量的夾角較小，方向一樣。且從式子中也可看出此分數的值介於-1~1之間。

## V. Data set

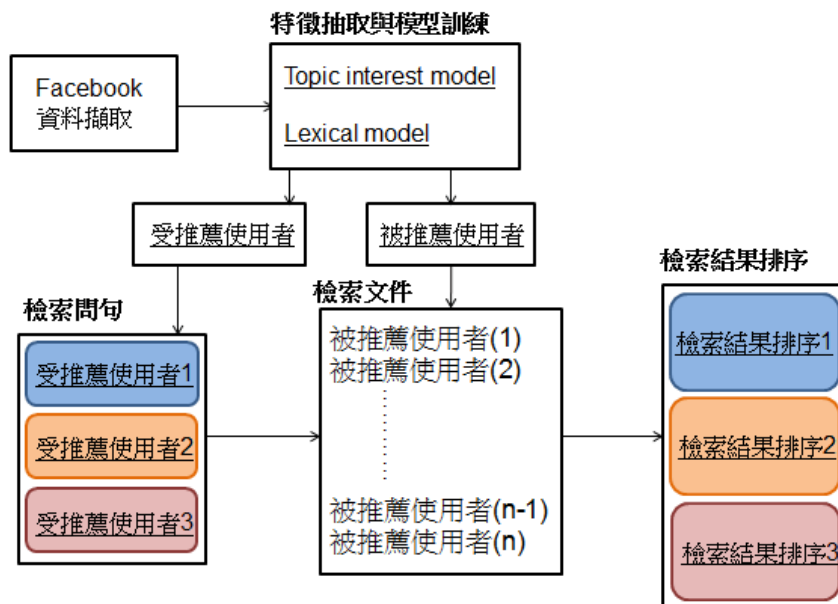
	id	owner_id	friend_id	common_friend_count	common_page_count	common_groups_count	owner_like_count	friend_like_count	owner_comment_count	friend_comment_count	rate
<input checked="" type="checkbox"/>	1	100000104623106	510233811	71	0	0	0	0	0	0	
<input checked="" type="checkbox"/>	2	100000104623106	520861499	77	0	0	0	0	0	0	
<input type="checkbox"/>	3	100000104623106	533158933	80	0	0	25	0	0	0	
<input checked="" type="checkbox"/>	4	100000104623106	541938630	81	0	0	0	0	0	0	
<input type="checkbox"/>	5	100000104623106	544087307	14	0	0	0	0	1	1	
<input type="checkbox"/>	6	100000104623106	545376935	1	0	0	4	0	0	0	
<input type="checkbox"/>	7	100000104623106	558283958	69	1	0	0	1	0	0	
<input type="checkbox"/>	8	100000104623106	558387917	12	0	0	0	0	0	0	
<input type="checkbox"/>	9	100000104623106	573905305	34	0	0	0	0	0	0	
<input type="checkbox"/>	10	100000104623106	594136394	73	0	0	0	0	0	0	
<input type="checkbox"/>	11	100000104623106	603042004	56	0	0	5	0	0	0	
<input type="checkbox"/>	12	100000104623106	605827866	26	0	0	4	0	1	1	

我們找了二十位同學當作實驗中接受推薦使用者，而前處理時我們可得一組特徵表格如上所示。特徵表格分別有兩使用者之間「共同好友數」、「共同評論數」、「共同群組數」、「共同粉絲頁數」...等等。而所有擷取到的使用者數量共約有一萬兩千名使用者。

對每一位接受推薦的使用者，我們希望從一萬兩千名使用者中推薦好友。因此我們將此系統化為檢索系統的架構，二十位接受推薦者視為檢索系統中的問句，其他將近一萬兩千個使用者則當作被搜尋的文件。而對受推薦使用者 A，其檢索結果答案則為所有從未與 A 有任何互動且與 A 有共同好友的人所形成之集合。



## VI. Experiment structure



實驗整體架構圖如上，主要分為前端的特徵抽取、模型訓練部分，與後端的檢索部分。前端特徵抽取、模型訓練部分為第二部分致第四部分所述，對社群網站 Facebook 上的資料作整理與分類，並用抽取出來的特徵(Features)去訓練兩組不同的模型(Topic interest model、Lexical model)，因此對每一個使用者都會有屬於其自己的兩組特徵向量。訓練完模型後將所有使用者分為兩群，受推薦使用者以及被推薦使用者，分別當作檢索系統中的問句(Query)以及文件(Document)。檢索部分則利用先前訓練好的特徵向量作檢索，可能兩種模型中的其中一種抑或是兩種模型的疊加，檢索過程中，每一個受推薦使用者會與被推薦使用者比較其相似度分數，相似度分數的計算方式依照不同的模型，Topic interest model 可用第四部分所述 KL divergence 計算，Lexical model 則可用 cosine similarity 計算相似度。最後對每一個受推薦使用者，依據與其相似度分數的高底可得其檢索結果排序，與事先知道的檢索答案比較則可知道系統成效之好壞。

## VII. Evaluation

使用者推薦系統的結果通常較為主觀，而本實驗選擇資訊檢索中常使用的評估機制，Mean Average Precision(MAP)做為評估標準，而此部分將會介紹 Precision、Recall、Mean Average Precision 三項標估機制。

### 1) 準確率 Precision 與召回率 Recall

準確率的定義為  $\text{準確率} = \frac{\text{檢所到的相關朋友數}}{\text{檢索到的朋友數}}$

召回率的定義為  $\text{召回率} = \frac{\text{檢所到的相關朋友數}}{\text{所有的相關朋友數}}$

準確率越高代表檢索結果的正確性越高，召回率越高代表系統找到越多的相關朋友。

在本實驗中若要得到準確率或召回率的值，通常必須對系統設定一個閾值，相關分數超過此值的人則判斷為好友，反之則忽略。系統的召回率與準確率會隨著不同的閾值而有不同的變化。一般而言準確率與召回率的變化是相反的，召回率的上升常伴隨著準確率的下降，反之亦然，兩者常有反向變化的趨勢。因準確率與召回率並無基於系統檢索排序結果評估，單純以準確率和召回率評估系統無法準確評估相關分數的優劣。

### 2) 平均準確率 Mean Average Precision

平均準確率的定義為  $\text{MAP} = \frac{1}{|Q|} \frac{\sum_{D_T^Q} \text{precision}(D_T^Q)}{|D_T^Q|}$

Q 為檢索系統中的查詢問句，在此則為某位接受推薦的使用者。

|Q|為受推薦使用者的個數。

$D_T^Q$ 為和查詢問句 Q 相關的文件，此為應推薦給受推薦使用者 Q 的好友。

$|D_T^Q|$  代表真正應推薦給使用者 Q 的好友數。

$\text{precision}(D_T^Q)$  為 $D_T^Q$ 在檢索結果排序位置前的準確率。

用MAP判別系統效能好壞，因考慮的排序前後的相關性，評估上較為適合，且不需要事先訂定一個閾值，MAP也是在搜尋系統中常用的評估方式。

## VIII. Experiment Result

Facebook ID	Baseline	PLSA - T16	PLSA – T32	PLSA – T64	PLSA – T128	LM	Interp
100000765935791	0.0108	0.0177	0.0218	0.0269	0.0316	0.0698	0.0603
100000104623106	0.0057	0.0141	0.0205	0.0383	0.0243	0.0141	0.0287
100000103194445	0.0058	0.0152	0.0410	0.0319	0.0208	0.0088	0.0564
100000014253155	0.0102	0.0253	0.0342	0.0421	0.0353	0.0236	0.0334
100000137599908	0.0031	0.0135	0.0176	0.0226	0.0160	0.0028	0.0234
1558816722	0.0048	0.0078	0.0095	0.0132	0.0105	0.0110	0.0096
1842812921	0.0102	0.0117	0.0217	0.0180	0.0173	0.0129	0.0280
1516894462	0.0131	0.0516	0.0661	0.0597	0.0545	0.0240	0.0434
100000567394401	0.0034	0.0137	0.0169	0.0168	0.0135	0.0072	0.0193
703655553	0.0075	0.0122	0.0110	0.0436	0.0294	0.0180	0.0118
1669483824	0.0029	0.0145	0.0168	0.0126	0.0112	0.0037	0.0563
1290116813	0.0030	0.0136	0.0162	0.0141	0.0186	0.0039	0.0106
1410142953	0.0064	0.0095	0.0199	0.0233	0.0173	0.0153	0.0180
100000217507802	0.0057	0.0200	0.0193	0.0215	0.0203	0.0081	0.0190
100000727624247	0.0112	0.0440	0.0453	0.0554	0.0541	0.0175	0.0448
100001152735256	0.0017	0.0034	0.0037	0.0037	0.0062	0.0033	0.0032
1433547786	0.0048	0.0198	0.0250	0.0226	0.0252	0.0109	0.0217
100000132459238	0.0084	0.0216	0.0177	0.0213	0.0241	0.0124	0.0185
100000159008393	0.0063	0.0115	0.0563	0.0387	0.0224	0.0070	0.0433
100000286638903	0.0071	0.0272	0.1040	0.0359	0.0566	0.0254	0.1087
avgMAP	0.0063	0.0175	0.0278	0.0268	0.0242	0.0143	0.0314

實驗結果如上表所示，縱軸第一列為二十個受推薦使用者之 ID，橫軸方向則為各種不同模型所得之結果，評估方式為之前所述 Mean Average Precision。Baseline 為隨機從使用者中推薦好友所得之 MAP，PLSA-T16 則利用第四部份中 Topic interest model 所做出的檢索結果，而 T16 代表主題數量為 16。而 T32 則為主題數量為 32。主題數量我們由 16 做到 128 個主題，而結果可看出在主題數量為 32 時所得的結果最為良好。而 LM 則為第四部份所述之 Lexical Model 所得之檢索結果，當純考慮語彙上的資訊所做出的檢索結果雖較 PLSA 的結果為差，但我們將兩結果做內差可得到另一個結果最為良好的檢索結果，因此我們可以知道同時考慮使用者在文件或群組的共同出現率的隱含變數以及使用者語彙上相似度的資訊，可以對推薦好友系統有更精確，更良好的影響。

## IX. Conclusion

以上是本組的專題報告內容。有別於現存的「好友推薦系統」，我們加入了一些「興趣」、「字彙」上的特性，來使「好友推薦」更為多元化。我們利用 PLSA、KL divergence、TFIDF、cosine similarity 等等工具與方法來實作本專題報告，並且在實驗結果中驗證我們的方向以及方法是有其效果的。

未來若是有機會，我們將會加入更多的使用者特性、使用者彼此間的互動等等交流，相信更精確更好用的好友推薦系統是指日可待的！此外，針對我們研究的議題與方向，此應用層面也可以拓展到一些市面上常見的交友網站、求職網站等等，相信加入這些資訊與特性，人們可以更順利、又效率地滿足自己的需求。「科技始終來自人性，也終將導入人性」我們始終相信這句話，也冀望著這世界能因科技而更加美好。

## X. Reference

- [1] Probabilistic Latent Semantic Analysis. Thomas Hofmann
- [2] Introduction to modern retrieval. G.salton, M.J.Mcquill.
- [3] <http://en.wikipedia.org/wiki/Tf%E2%80%93idf>
- [4] <http://developers.facebook.com/>
- [5] <http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-ranked-retrieval-results-1.html>