**Topic: Gift Recommender**

**r99922003 郭韋狄 r99922157 蘇適 r99944002 張衡**

**Weti.kuo, Penn Su , George Chang**

{r99922003, r99922157, r99944002}@csie.ntu.edu.tw

------------------------------------------------------------------------------------------------------

課程網規定報告要包含的東西如下：

4~6頁的short paper

Problem description and motivation

Literature survey

Proposed solution/design

Discussion and evaluation metrics

References (at least 8)

-----------------------------------------------------------------------------------------------------

**Gift Giver**

**ABSTRACT**

Keywords

**INTRODUCTION**

Suppose you are planning to attend a birthday party of a far relative. You may think about, "what am I going to send?" That might bother you an entire afternoon since you think this question, because you have no idea what kind of gift this relative will like. Pick a nice gift has been a hard and an interesting topic for a long time.

There are many factors to affect final decision. As a sender, budget is a big issue. If a sender spends a lot of money to prepare a gift, the receiver may be embarrassed to receive it because it’s too expensive. Under the limited budget, sender might consider the receiver's interest, religious, gender, age, and the relationship between the receiver and himself/herself. In addition, the date to gift is also a very important factor. Under normal circumstances, you do not send a coat in a hot summer.

How to find a suitable gift? In the past, you go to a department store, and try to find the ideal gift, browsing through one store after the other, but nothing seems perfect. If you are not a decisive one, the process is time-consuming. At present, there are many online website to recommend gifts, e.g. Gift ideas, Hallmark Cards, Amazon gifts, Find gift, Yahoo! Gift Finder, JCpenny, Macy's, Sears[Reference]. However, according to our observations, most of them cannot supply satisfactory service. Usually, gifts websites prefer using best sellers to be their recommendations. The gift website usually comes out the similar recommendation list even when we sent the different preferences, occasion and receiver’s age. And the list has too much detail, such as see’s toffee chocolate 16oz 20U.S.D. For non-expert user, they can hardly show their intimate part. Some people ask others opinion when they are picking a nice gift. But as the sender consults with more and more people, the gift might not be surprise because people like to gossip all day.

Based on the above reasons, we explore a system named "GiftGiver" to recommend a list of gifts to help senders. In order to figure out what should be a nice gift. We follow the idea of Scenario-Oriented Recommendation [Edward]. We analyze each gift within 8 features. Besides, gifts item are not specific product. GiftGiver come out a concept of the gift idea, just like flower, tea mug etc. In the way that sender can have their own gift choice. If sender has no idea about a particular item, our system can also give sender the detail of the gift which parses from other websites.

The uncommercial design made our system be a good helper for the people who what to send a gift. As group computer scientist, using computer technology to help people to have a better life is a dream of us.

-------------------------------------------------------------------------------------------------------

**Pre-questionnaire**

Before implementing the system, we want to figure out what is the real reason to affect people sent gifts. We design a questionnaire to ask 3 adult female and 10 adult male about what they will consider when they sent a gift.

The question is like “when you are picking a gift, is the relation between you and receiver seems important to you?” Each question has 5 levels of score. 1 score means not important and 5 score instead.

From the result table, we can separate the score in to 3 part – Red (score> 4), pink (4>score>3), and green (3>score). The red part is the most important union. Such as relationship, price, sender’s gender etc.

This result is quite reasonable. In general, the gift will finally own by the receiver than own by both sender and receiver. So the options about receivers are higher than the options about senders. In other part, price, occasion, sent date and receiver’s gender are common to affect whether the gift is reasonable.

Occasion = 4.583

Relationship = 4.917

Price = 4

Date = 4.06

Interests = 4.31

Receiver’s Career = 3.75

Receiver’s Gender = 4.583

Receiver’s Religion = 2.75

Receiver’s Age = 4.25

Sender’s Career = 2.583

Sender’s Gender = 2.583

Sender’s Religion = 1.583

Sender’s Age = 2.583

上圖需要修改!!!!(英文、五分制、字數等......)

Besides, we interview 5 people about how they think when they consider a nice gift. We find out that the interviewer has the same consideration with questionnaire answer. We also find out people consider about the relationship first, then is occasion. The other effect seems not relative to each other effect for them.

More information actually helped us to find a better gift recommendation. To collect information from the senders, we decide to form a web user interface. The UI’s content includes occasion, relationship, sender's favor, receiver's profile (which includes receiver's career, gender, age, interests), and send date

........................................................................................................................................

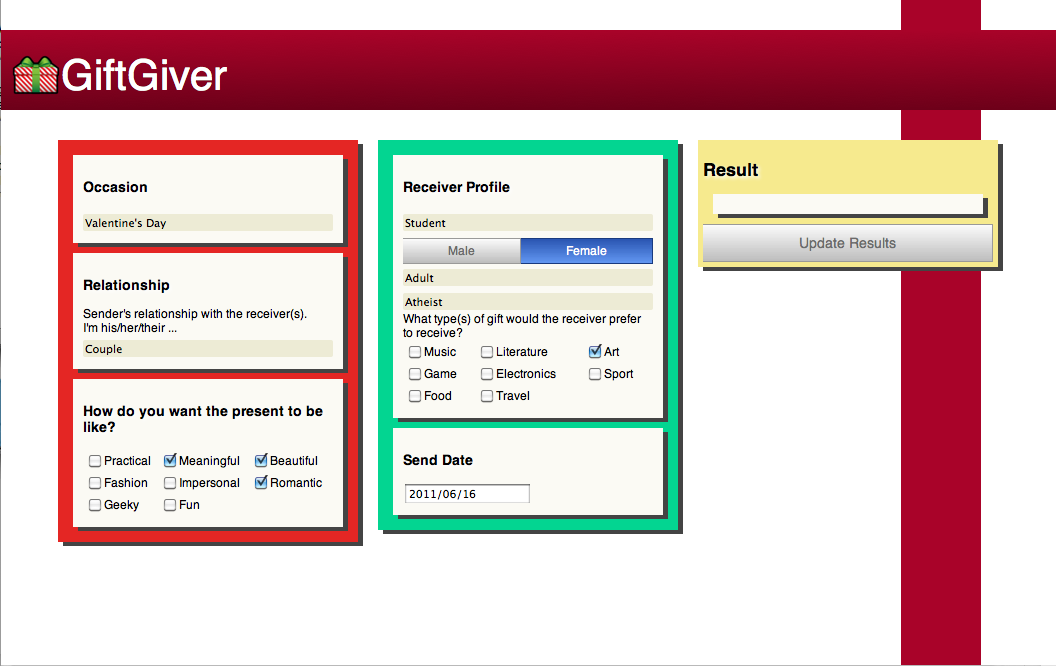
**System Architecture**

以下我們依序介紹............

Flow (introduction of the system)

First, we let senders fill in their consideration on the UI question. For reducing the loading of senders, we using check box and combo box and make a lot of common option. Second, after fill in the questions, the system will use the information from the UI questions, and generates a list of 20 recommended gifts. Senders can also click the name of specific gift. System will help sender to find the relative product form the commercial websites.

UI設計說明：



The UI of the GiftGiver is as below[picture2.1], it is a 1page website. We separate the questions into 3 big parts, the main consideration, the Receiver profile and the Result. Sender can easily fill in their consideration step by step. After fill in the answer, system will comes out 20 recommended gifts. If receiver is not satisfied, he/she can also change the answer and search again.

Compare to gifts website, our UI has a better flow but less option in some question. For example, Gifts.com has 9 ages degree, such as 0~6, 7~12, 13~19, 20~30, 31~40 etc. However, we prefer using fuzzy set to define our age instead of clear separation by years. Because nouns like child, adult etc. will help the system in the commonsense part. On the other hand, some people still looks like a child even that person has already more than 30s. The fuzzy part will come out positive result.

Feature Vector of Gift and Occasion

As a recommendation system, we use a feature matching type of the recommendation system [韋狄再common sense報過的那邊survey paper].

We analyze 60 gifts items and 3 common occasions by 8 features.

The features are:

1.[ Practical ] – Is it functional or just for decoration?

2.[ Meaningful ] – Is it has other meaning besides its function?

3.[Beautiful ] – Is it good looking?

4.[Trendy ] – Is it fashion?

5.[Impersonal ] – Is it just for him/her or other people can use it?

6.[Romantic ] – Is it present for love and sweet?

7.[Geeky ] – Is it electrical?

8.[Fun ] – Is it let you happy in a pure way?

Each gifts and occasion will have a vector of 8 features. The value is normalized to[0,1].

Then the Vector will looks like below:

Gift -> Book [ 1.0, 0.6, 0.2, 0.4, 1.0, 0.0, 0.0, 0.6]

Occasion -> Valentine’s day [ 0.2, 1.0, 0.6, 0.2, 0.0, 1.0, 0.2, 0.2]

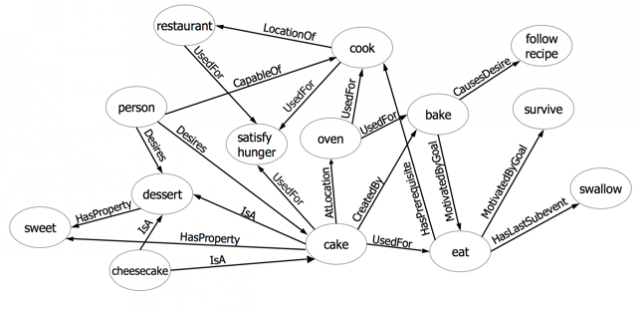
Commonsense and commonsense knowledge base

In our system, we use Open Mind Common Sense, a knowledge corpus that contains 800,000 sentences about everyday life, gathered from Web volunteers. Using this resource, we successfully built a fashion recommendation system.

# ConceptNet

# ConceptNet aims to give computers access to common-sense knowledge, the kind of information that ordinary people know but usually leave unstated. The data in ConceptNet is being collected from ordinary people who contributed it on sites like [Open Mind Common Sense](http://openmind.media.mit.edu/). ConceptNet represents this data in the form of a semantic network, and makes it available to be used in natural language processing and intelligent user interfaces.

ConceptNet is an open source project, with a Python implementation and a [REST API](http://csc.media.mit.edu/docs/conceptnet/webapi.html) that anyone can use to add computational common sense to their own project. A great tool to help you use ConceptNet in your software is [Divisi](http://csc.media.mit.edu/analogyspace/divisi).



Commonsense analyze tool - Divisi

Divisi is a library for reasoning by analogy and association over [semantic networks](http://en.wikipedia.org/wiki/Semantic_network), including common sense knowledge.

Divisi uses a sparse higher-order [SVD](http://en.wikipedia.org/wiki/Singular_value_decomposition) can help find related concepts, features, and relation types in any knowledge base that can be represented as a semantic network. By including common sense knowledge from [ConceptNet](http://conceptnet.media.mit.edu/), the results can include relationships not expressed in the original data but related by common sense.

禮物清單的來源：

在比較諸多送禮網頁的品質後，我們決定以Amazon, Costco, JCpenny[相關Reference]建立禮物清單的基底。

1. 一開始我們先用人力蒐集實體的禮物[將感情方面的過濾掉]，得到約200個禮物的清單。

2. 接著我們將這些禮物包在一個category裡，利用set expansion得到約1000個禮物的set.

3. 接著我們再以人力將這1000個禮物過濾到約250個左右。

-------------------------------------------------------------------------------------------------------

Gift Giver Algorithm:

baseline: (法一)

一開始我們直接將user所有的選擇當作concept，各別和禮物清單裡的所有禮物算divisi分數，排出前20名的禮物。效果非常的差！

In the beginning, we used to let commonsense play the main role of the recommendation. We pick up nearly 300 gifts from the website like Costco, Amazon, and JCpenny’s as gifts set. Then build a prototype of a system. Calculate the spreading activation score in the conceptNet- Divisi. Each gift will calculate the spreading activation score to each kind of considerations. Such as birthday to “book”. In the end, sort by the sum of the spreading activation score.

We soon find out that the result is so weird. The gift-Nuts has appeared in any kind of situation. And “condoms” show up with “coupe”. For improve this kind of Awkward problem. We decide to build another system which commonsense play the helper role.

Gift Giver Algorithm: (法二)

前置作業：

我們挑出60個禮物，每個禮物以一個八維的向量表示[practical, polysemy, appearance, trendy, impersonal, romantic, geeky, fun], 當中每個value會界於0~1，0代表極不相關，1代表極相關。另外我們定義了三個Occasion的向量，分別是

Christmas day [0.8, 0.8, 0.8, 0.2, 0.6, 0.6, 0.4, 0.8]、Valentine's day[0.2, 1, 0.6, 0.2, 0, 1, 0.2, 0.2]、Birthday[0.4, 0.4, 0.4, 0.6, 0.4, 0.2, 0.4, 0.6]。

step 1:

diff = {}

for name, features in gifts.items():

diff[name] = 0

for feature in features:

diff[name] += pow( (float(occasions[occasion][feature]) - float(gifts[name][feature])),2)

sorted\_gifts = sorted (diff.items(), key=lambda x: x[1])

result\_gifts = []

for item in sorted\_gifts:

result\_gifts.append((item[0], gifts[item[0]]))

從使用者的問卷中，我們發現Occasion是一個影響禮物性質很大的因素，所以在step 1時，我們將使用者所選的Occasion向量和每個禮物向量分別算the sum of squared difference(平方差)，公式如下：

定義禮物為G，Occasion為O，i為第幾維的向量，diff(G,O) =

其值越小，代表該禮物越符合該Occasion，我們依該分數做排序。

step 2:

inferior\_gifts = []

for gift in result\_gifts:

for input\_feature in input\_features:

if float(gift[1][input\_feature]) < 0.4:

inferior\_gifts.append(gift)

break

for gift in inferior\_gifts:

for index, old\_gift in enumerate(result\_gifts):

if old\_gift[0] == gift[0]:

del result\_gifts[index]

ˋ

在step2時，我們利用sender想要送的禮物feature (有打勾的那些欄位)，設定這些feature的threshold為0.4，只要該禮物的某一項feature value(有打勾的那些欄位)小於該值的話，即將其從result\_gifts移到inferior\_gifts。

step3:

在step3時，我們將user選的Occasion、Relationship合為一個category(命名為scenario)，接著利用candidates = spread.left\_category(scenario).top\_items(20)找出最有關係的前20個items。最後我們比對禮物清單中是否有和這20個item一模一樣的禮物，如果有的話，我們將其在category的分數\*10；反之，我們將其設為0。

results = []

# Occasion, Relationship => Category

if relationship == 'none':

scenario = divisi2.category(occasion)

else:

scenario = divisi2.category(occasion,relationship)

# find similarity between a gift and a category

candidates = spread.left\_category(scenario).top\_items(20)

for gift in result\_gifts:

score = 0.0

for candidate in candidates:

bait = float(candidate[1])

# First level

if candidate[0] == gift:

score += bait\*10

# Because Part 3 is very important, so we multiply 10.

results.append ( (gift[0], score) )

for gift in inferior\_gifts:

score = 0.0

for candidate in candidates:

bait = float(candidate[1])

# First level

if candidate[0] == gift:

score += bait\*10

# Because Part 3 is very important, so we multiply 10.

results.append ( (gift[0], score) )

step4:

在step 4時，我們利用receiver的職業、性別、年齡、興趣分別和每個禮物算spread.entry的分數，接著再依據我們附件一問卷所得的weight對應相乘，將其加總後，除以total weight做normalize，得到receiver特性的分數。若該值大於0.1時，我們將其放到禮物清單的前面；反之，不變動它所在的位置。

# receiver profile: occupation, age, gender

receiver\_results = []

normalization\_coefficient = 5.0

receiver\_occupation\_weight = 3.75 / normalization\_coefficient if receiver\_occupation != 'none' else 0.0

receiver\_age\_weight = 4.583 / normalization\_coefficient if receiver\_age != 'none'

else 0.0

receiver\_gender\_weight = 4.25 / normalization\_coefficient if receiver\_gender != 'none'

else 0.0

types\_weight = 4.44 / normalization\_coefficient if types != []

else 0.0

sum\_weight = receiver\_occupation\_weight + receiver\_age\_weight

+ receiver\_gender\_weight + types\_weight \* len(types)

# calculate the occupation score, age score, gender score separately.

for gift in result\_gifts:

receiver\_occupation\_score = receiver\_occupation\_weight \* spread.entry\_named(gift[0], receiver\_occupation)

receiver\_age\_score = receiver\_age\_weight \* spread.entry\_named(gift[0], receiver\_age)

receiver\_gender\_score = receiver\_gender\_weight \* spread.entry\_named(gift[0], receiver\_gender)

types\_score = 0

for type in types:

types\_score += types\_weight \* spread.entry\_named(gift[0], type)

receiver\_total\_score = (receiver\_occupation\_score + receiver\_age\_score + receiver\_gender\_score + types\_score)/sum\_weight

if receiver\_total\_score >= 0.1:

receiver\_results.append ( (gift[0], receiver\_total\_score) )

else:

receiver\_results.append ( (gift[0], 0) )

step 5:

整合上述四步驟後，推薦出前20名的禮物清單。

**------------------------------------------------------------------------------------------------------**

**EVALUATION**

Usability test

Questionnaire

由於時間因素，我們請台大資工系五位教職員幫忙測試我們的系統，測試結果如下表所示。

下表需要修改：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | test\_1 | test\_2 | test\_3 | test\_4 | test\_5 |
| **UI使用是否順暢** | 3 | 4 | 3 | 3 | 4 |
| **選項項目是否充足** | 3 | 4 | 3 | 4 | 3 |
| **選項是否包含我的考量** | 4 | 3 | 2 | 3 | 3 |
| **理想禮物是否在清單內** | 4 | 4 | 3 | 4 | 4 |
| **禮物意象夠明確** | 5 | 3 | 3 | 5 | 4 |
| **給與價格會對我有幫助** | 4 | 5 | 5 | 5 | 4 |
| **下次選購禮物會參考本站** | 5 | 4 | 4 | 5 | 4 |

**Results**

**RELATED WORK**

Traditionally, recommender systems are usually classified into the following three categories:

1. **User-based Collaborative Filtering Systems[Reference 1篇]:**

The user will be recommended items similar to the ones the user preferred in the past.

1. **Item-based Collaborative Filtering Systems[Reference 3篇]:**

The user will be recommended items that people with similar tastes and preferences liked in the past.

1. **Hybrid Recommender Systems[Reference 2篇]:**

These methods combine the above two methods.

They usually suggest books, music albums, from a set of input parameters, possibly including user profiles, purchase history, etc.

In recent years, many researchers use novel technology to recommend, e.g. Edward use Commonsense to recommend clothing[ ]. Furthermore, some researchers use recommender system in interesting domains, e.g. Cosley’s Movie Recommendation[ ], and Koutrika’s Course Recommendation[ ]. Moreover, the display style of Top 10 is popular again recently []. We follow the latest trend, and use the novel technology in interesting domain. In short, we are the first one that combine traditional recommendations and Commonsense, and apply them in Gift Recommendation

In recent years, many researchers 以新穎的技術來做推薦，像是Edward學者利用Commonsense做衣服推薦[Reference What am I]。此外，也有學者將Recommender System用在有趣的領域上，像是Cosley學者的Movie Recommendation[Reference SuggestBot], Koutrika學者的Course Recommendation[Reference FlexRecs:

], and so on. 此外, Top 10 呈現方式最近又重新流行起來[Goal-oriented web-site navigation for on-line shoppers]。我們follow最新的流行趨勢，將新穎的技術用在有趣的domain上。In short, 我們是第一個結合傳統推薦、commonsense，並且應用在禮物推薦上的研究。

**CONCLUSION**

不論是哪個國度的人，送禮是社交活動中常見的行為。在不同的場合、不同的時間點裡，該如何挑選出合適的禮物呢？我們的Gift Giver顯然是一個方便的工具。

**Future Work**

禮物推薦是一個非常新穎的推薦系統，過去沒有人做這方面的推薦，所以我們算這個方面的先驅；但相對地，我們能參考的文獻也非常有限，故在系統設計上還有許多可以進步的空間。第一點，目前每個禮物的feature value是我們人工打上去的，非常辛苦。未來我們希望能透過web mining、Machine Learning等……相關技術，自動得到這些feature value。第二點，目前我們的演算法還未將religious、date等……feature考慮進來，未來我們可以嘗試運用這些feature，看是否能讓效果更好。第三點，等我們的系統運作的夠穩健時，我們會提供online服務，讓更多的user使用，並請他們給予我們回饋，讓我們能更加進步。第四點，目前系統中禮物數仍不足夠，未來我們會開放使用者推薦禮物，我們再透過投票表決機制決定該禮物是否加到我們的資料庫中。第五點，在系統online後，我們希望它能充分應用Commonsense的優點，在不同的location運用不同的文化背景知識，做出適性化的推薦。Last but not the least，由於目前Commonsense的資料庫尚未齊全，故我們將它運用為輔助性的角色。在未來，如果Commonsense的資料庫足夠齊全的話，我們可運用machine learning技術調整它和feature value的weight。

**REFERENCE (先不按順序，把有的先放上來)**

1. What am I gonna wear?: Scenario-Oriented recommendation

2. Personal Choice Point: Helping users visualize what it means to buy a BMW

3. Beating Common Sense into Interactive Applications

4. A goal-oriented interface to consumer electronics using planning and commonsense reasoning

5. User Interface Goals, AI Opportunities

6. An Algorithmic Framework for Performing Collaborative Filtering

7. Amazon.com Recommendations: Item-to-Item Collaborative Filtering

8. Hybrid Systems for Personalized Recommendations

9. Evaluating Recommender Systems: An Evaluation Framework to Predict User Satisfaction for Recommender Systems in an Electronic Programme Guide Context

10. A new approach to evaluating novel recommendations

11. SuggestBot: Using Intelligent Task Routing to Help People Find Work in Wikipedia

12. FlexRecs: Expressing and Combining Flexible Recommendations

13. Goal-oriented web-site navigation for on-line shoppers

14. Item-based collaborative filtering recommendation algorithms

// 以下網址不知道該如何放到.bib檔裡，且可顯示出來。

15. Gift ideas: http://www.gifts.com

16. Hallmark Cards: http://www.hallmark.com/online/

17. Amazon gifts: http://www.amazon.com/gp/gift-central

18. Find gift: http://www.findgift.com/Categories/Toys/

19. Yahoo! Gift Finder: http://shopping.yahoo.com/

20. JCpenny: http://www.jcpenney.com/jcp/default.aspx

21. Macy's: http://www1.macys.com/index.ognc

22. Sears: http://www.sears.com/

附件一 (實作系統前的問卷)

可放網址????

附件二 (使用者問卷回饋)

可放網址????