Improving Opinion Mining with Feature-Opinion Association and Human Computation

CHAN, Kam Tong

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Computer Science and Engineering

May 2009

Abstract of thesis entitled:

Improving Opinion Mining with Feature-Opinion Association and Human Computation

Submitted by CHAN, Kam Tong

for the degree of Master of Philosophy

at The Chinese University of Hong Kong in May 2009

Opinion mining aims at analyzing people's feeling towards a particular topic. Typical approaches in solving this problem employ various techniques from the field of natural language processing, statistical learning and information retrieval. It has been a hot research field in recent years. Although a number of studies have been conducted, current solutions have been suffering from various limitations, 1) simplistic sentiment model, 2) limited scope of analysis, and 3) lack of detailed result. Existing lexicon based models use simple aggregation functions to sum up the overall sentiment score and assign binary classes (i.e. positive or negative) to the review in question. Analyses are limited to subsets of all available information, such as words carrying a particular part-of-speech tag. Results based on these approaches are limited to simple plus or minus scores in which no distinctions could be made on the level of "positiveness" of a piece of opinion.

In the first part of this thesis, we study the effects of Feature-Opinion Association (FOA) in opinion mining. Instead of using a simple aggregation function to sum up the polarities of all opinion expressing components, we propose an FOA algorithm to prune some of the unrelated components away. Experimental result shows that FOA helps to improve opinion mining accuracy.

In the second part of the thesis, we study how the idea of human computation can be incorporated into the opinion mining process. We propose a social game framework by extracting the common components of existing games. Based on this framework, we derive guidelines for designing new social games systematically. We finally designed a new social game called FeatureGuess, to collect the feature-related information to improve the Feature-Opinion Association process.

Finally we explore the use of opinion mining in bookmark recommendation systems. We observe that tags, which contain concise information about the bookmark, may represent the taste of users. We propose a collaborative filtering based model for bookmark recommendation systems that make use of tagging information. We analyzing the sentiments expressed in tags and experimental results show that it has positive impact on recommendation results.

利用特徵意見結合及人類運算改進意見挖掘

陳錦棠

「意見挖掘」的目的是分析人對於一個特定主題的感覺。現有用於解決這個問題的方法一般都採用了自然語言處理、統計學習和信息檢索等技術。在最近幾年間，意見挖掘成爲了一個十分熱門的研究領域。雖然已進行了大量有關方面的研究工作，但目前的解決方案仍有各種限制或缺憾，如：(1) 簡單的情感模型、(2) 能進行分析的內容十分有限、和 (3) 一般意見挖掘缺乏詳細的結果給使用者參考。現有使用字典爲基礎的方法使用簡單的聚集公式來得出文章的整體情緒評分(即正面或負面)。而現有的方法只能對部分信息(如關鍵字或形容詞) 進行分析，其他信息是未能被使用。使用這些方法所得出的意見挖掘結果只局限於簡單的正負分數，使用者沒法區分意見的強弱度。

在本論文的第一部分，我們會分析「特徵意見結合」(ＦＯＡ)在意見挖掘過程中能產生的效用。對比起現有的方法一般使用句字裡所有詞彙放進簡單的聚集公式裡進行意見挖掘，我們的方法會先利用ＦＯＡ算法移除與特徵不相關的意見後才進行分析。實驗結果表明，ＦＯＡ有助於提高意見挖掘的準確性。

在第二部分的論文，我們研究如何利用人類運算以改進意見挖掘的算法。我們的目標是外包部分意見挖掘的分折過程給人類。以現時的技術，有很多對於人類雖然是很簡單的問題，但對於計算機來說是十分難以解決的。在意見挖掘的領域中自然語言處理是一個困難的部分，我們研究如何可以有效地外包這部分給人類以增加意見挖掘結果的詳細程度。

最後，我們探討如何利用意見挖掘技術來改進書籤建議系統。我們注意到在書籤系統中的標籤（ｔａｇ）含有用戶對於網頁的喜好資訊，對標籤進行情感分析將有助於改進書籤建議系統的建議結果。

Acknowledgement

I have to say Thank You to many people around me. The first person that I would like to thank is my supervisor, Prof. Irwin King. Prof. King is a person with a lot of ideas. He is also a very kind person and he always like to share many things with us. Without his careful guidance, I would not have completed my M.Phil. study so smoothly. What Prof. King has inspired me was not only on academic research, but also on how to be a world scholar by encouraging me to attend international academic events. During my two years of study, I received several opportunities to go abroad and meet with different people. These valuable experiences have transformed me to be a better-rounded people and the effect is life-long.

I would also like to thank Prof. Jimmy Lee for his fruitful advices during my study. The comments he gave me allow me to do better researches and improve my presentation skills. I am also graceful for Prof. M.C. Lee and Prof. Li Qing for sparing the time to be my examiners. The feedbacks from you are very important in finishing my final thesis.

I would also like to take this chance to express my appreciation to my fellow colleagues in CUHK. Not only do they share a lot of ideas with me on research, but they have also made my two years study joyful.

Finally, I felt an immense gratitude to my family for their support during my M.Phil study. Their support enabled me to focus on my study without worrying anything else.

To my dearest family.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Major Topic

In this chapter, we discuss the two main topics of this thesis: (1) opinion mining and (2) human computation. We go through the basic concepts of each topic and discuss the motivations of our researches. We then describe the thesis chapter organization.

1.1.1 Opinion Mining

Opinion mining is a new and important research topic in the field of web data mining [60, 10, 34]. The aim of opinion mining is to analyze the sentiments expressed by people on the web. It draws large attention in recent years because of its wide range of possible applications [9, 96, 46]. Big companies and business people can make use of opinion mining perform marketing researches [25]. Politician can better listen to the public to adjust their campaign policies. Consumers can look for the opinions of the products they want to buy before buying them [26]. In general, opinion mining helps to collect information about the positive and negative aspects of a particular topic.

Depending on the level of interest, there are two types of opinion mining. The first one is Document-level Opinion Mining, which is to determine the overall sentiment of a given review. It omits whether individual aspects of the topic are positively/negatively commented. The use is limited and it is not the focus of this paper.

The second type of opinion mining is Feature-based Summarization (FBS). The *feature* here means the aspect of a given topic that is being commented. In FBS, we identify target features mentioned in the reviews and determine sentiment expressed by the author. This type of opinion mining is more general and in fact we can consider Document-level Opinion Mining as a special case of Feature-based Summarization. Therefore, in this thesis, we focus on the latter type of opinion mining.

1.1.2 Human Computation

Web 2.0 technologies have brought new ways of connecting people in social networks for collaboration in various on-line communities. Social Computing [86, 87] is a novel and emerging computing paradigm that involves a multi-disciplinary approach in analyzing and modeling social behaviors on different media and platforms to produce intelligent and interactive applications and results.

In this thesis, we explore the possibility of incorporating social computing technique into the opinion mining process. In particular, we explore how we can design a game based on the concept of human computation to solve some of the most difficult problems in opinion mining. The idea of human computation is to outsource part of the computation process, which are difficult for computers, to human in form of a entertaining game. Tons of game based on this idea, such as ESPGame [80] and Peekaboom [84], have been developed to solve various computationally difficult problems. We believe that this idea is well-suited with the opinion mining problem in which natural language understanding is a very difficult problem for computers.

Finally, we make use of the opinion mining techniques to solve some of the problems in field of social computing. Social bookmarking [88] sites like Del.icio.us has drawn significant attention nowadays because of its ease of use and the power in connecting people. One of the most important feature in Del.icio.us to recommend new pages to user. However, the traditional way of doing recommendation [67, 74] are rather simple. Tags that are attached to the bookmarks may carry critical information about the taste of a user. In this thesis, we explore the use of opinion mining on bookmark recommendation system by analyzing the sentiment expressed in tags. This is an interesting problem that could help improving the recommendation result significantly.

1.2 Major Work and Contributions

The main work and contributions during my studies are:

1. **Feature Opinion Association Model in Sentiment Analysis.** We study the Feature Opinion Association (FOA) problem in sentiment analysis. We propose an algorithm that uses statistical and structural information of review text to perform the association between features and opinion terms. The association acts as a pruning strategy that removes irrelevant opinions for each feature and leave only relevance ones for sentiment classification. The result is published in the paper "Lets Tango – Finding the Right Couple for Feature-Opinion Association in Sentiment Analysis," in Proc. Advances in Knowledge Discovery and Data Mining 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009.

2. **Social Game for Opinion Mining.** Social games is an innovative approach to channel human abilities to solve

computational difficult problems. We study existing social games and develop a model that captures the common components of these games. We also derive guidelines for designing new social games to solve other problems. This work enables us to better understand these games and would allow new game development to be easier. This work is submitted to The 2009 IEEE International conference on Social Computing (SocialCom-09) as “Mathematical Modeling of Social Games” and is under review. Another related work is already published in the paper “A Brief Survey of Computational Approaches in Social Computing,” in Proc. Proceedings of the 2009 International Joint Conference on Neural Networks (IJCNN2009), 2009.

3. **Tag Sentiment Analysis for Social Bookmark Recommendation.** Tradition social bookmarking sites deploy collaborative filtering techniques to generation site recommendations to users. This work focus on the user's opinion expressed in tags of bookmarks. We create a model for bookmark recommendation system and analysis the major components in it. We propose a novel tag sentiment analysis algorithm that can distinguish between sentiment expressing tags and normal tags. We propose a new multi-matrix collaborative filtering algorithm for bookmark recommendation that can incorporate sentiment expressing tags into consideration. This work is prepared to be submitted to the ACM Conference on Information and Knowledge Management (CIKM 2009).

1.3 Thesis Outline

In the next chapter, we review the current progress on opinion mining. We categorize existing works based on their nature and extract important details of each work. We also go over the existing works on social bookmarking and social games. These two topics are closely related to our work and are reviewed in details. In Chapter 3, we present how Feature-Opinion Association (FOA) can be incorporated into the sentiment analysis process to improve classification accuracy. Chapter 4 presents our analysis on social games. We give details of the social game framework and design new games based on the derived guidelines. Chapter 5 describes our proposed model for social bookmarking system that makes use of tag subjectivities to improve recommendation results. Finally, we conclude the thesis and discuss the possible work that could be done in the future.

□ **End of chapter.**

Chapter 2

Literature Review

2.1 Opinion Mining

There are three major steps in opinion mining: i) extract the set of features the author is commenting on, ii) extract the related word/phrase that is used to describe the feature and iii) determine whether each of the word/phrase is expressing a positive, negative or neutral sentiment. Various approaches have been proposed to solve problems related to these processes. In this section, works related to each of the three steps in the opinion mining process will be discussed.

2.1.1 Feature Extraction

This is the first step in opinion mining. By feature extraction, we mean that given a text document, we try to extract the target object's feature commented by the author. It is different from the tradition feature extraction in the area of machine learning. Various techniques have been developed for feature extraction already, and they can be summarized in the following sections.

Association Rule Mining

Hu [31] observed that features can be divided in to explicit and implicit. By explicit it means that the feature itself appears in the text, for example in the following sentences:

> ”The battery life is long.”
> ”The camera is small enough to put in my pocket.”

The feature ”Batter Life” appears in the first sentence explicitly. However, in the second sentence, the feature ”size” is not directly mentioned but only implied by the word ”small”. Liu first deal with explicit features, he proposed to use association rule mining techniques [26] (Integrating Classification and Association Rule Mining) to identify explicit features that are frequently mentioned in the review corpus. The intuition behind this is simple: Customers may talk about a lot of things in their reviews. However, when they talk about the same target features, the term that they are using will converge. He also suggested that not all the itemsets extracted by the algorithm are real features. He further proposed two pruning method to increase the precision of feature extraction, namely, Compactness Pruning (for remove items that does not always appear in the same order) and Redundancy Pruning (to prune short features are part of the longer features).

Liu [33] followed up the work of extracting infrequent features. Base on the observation that features usually appear closely with the opinion words, he suggested to collect a set of adjective words (opinion words) that modifies the extracted frequent features. Then the infrequent feature algorithm works as follows:

Algorithm 1 Infrequent Feature Extraction

```
for each sentence in the database do
    if there are no frequent features but one or more opinion words then
        Add the nearby noun/noun phrase into the infrequent feature list
    end if
end for
```

Liu [50] continued his work on implicit feature extraction. His idea is to manually label words that implicitly related to a feature to the feature itself. For example, the sentence:

”The camera is heavy.”
”It is too big put in my pocket.”

will carry a tag $< weight >$ and $< size >$ respecitively. Then association mining can be used in similar fashion to identify implicit features in the testing corpus.

NLP-based Feature Extraction

Another research team from IBM tried to solve this problem from a very different perspective. While the techniques mentioned above are purely statistical, Yi [95] proposed to incorporate NLP (Natural Language Processing) techniques into the feature extraction process. Based on the part-of-speech (POS) tag defined by Penn Treebank [53], they defined the following:

Base Noun Phrases (BNP)

BNP allows feature terms to be in one of the following patterns:
NN, NN NN, JJ NN, NN NN NN, JJ NN NN, JJ JJ NN
where NN and JJ are POS tag for nouns and adjectives.

Definite Base Noun Phrases (dBNP)

dBNP are BNPs that are preceded by the definite article ”the”

Beginning Definite Base Noun Phrases (bBNP)

> bBNP are dBNPs that appear at the beginning of sentences followed by a verb phrase

Using bBNP's definition as a heuristic for feature term extraction, they suggested using likelihood-ratio test as follows:

Let D_+ be a collection of document focused on a topic T and, D_- be the rest, then the likelihood ratio $-2\log\lambda$ is defined as follows:

$$-2\log\lambda = -2\log\frac{max_{p1\leq p2}L(p1,p2)}{max_{p1,p2}L(p1,p2)} \tag{2.1}$$

The higher the value of $-2\log\lambda$, the more likely the phrase is relevant to the topic. Using this likelihood ratio and bBNP heuristic, the author is able to achieve 97% and 100% precision in digital camera and music domain respectively.

2.1.2 Sentiment Analysis

The opinion extraction and sentiment determination process is tightly coupled together. Sentiment Analysis is the term used in many opinion mining related research papers.

Lexicon Based Approach

Early works in this area are mostly based on lexicon based approaches. The idea is to build a lexicon of words with known sentiment for sentiment classification. Sentiment extraction techniques are very similar (base on the adjectives surrounding the feature term). They differ from each other in the lexicon building process.

Hu [31] suggested using WordNet to build bipolar clusters. Starting with a head for each cluster, e.g. fast for positive and slow for negative, synset of these words will be added to the corresponding the clusters. It is based on the assumption that words that share the same orientation are synonyms and those having opposite orientations are antonyms. By having an initial seed, the lexicon can be expanded by following the synsets in WordNet.

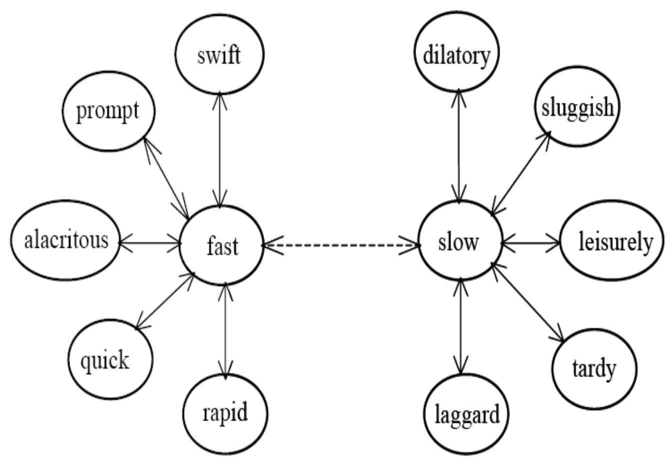

Figure 2.1: Sample Bipolar Cluster suggested by Hu [31]

After building the lexicon, sentiment can be predicted by computing the sum of sentiment of individual adjective extracted in the sentence containing the feature term. (Assume that positive words has a value of 1 and -1 for negative words) Using this method, Hu [31] is able to achieve an average of 84

Kim [40] (Determining the Sentiment of Opinions) proposed a technique similar to Hu [31]. Instead of just looking at adjectives, verbs are also good indicators of sentiment. Kim also observed that in the lexicon expansion process, some common words (”great”, ”strong”, ”take”, etc) appear in both side (positive and negative) very frequently. He, therefore, suggested a measure of strength of sentiment polarity in order to determine how strong a word is positive (and negative). The sentiment polarity is obtained by computing:

$$\arg\max_{c} P(c|w) = \arg\max_{c} P(C|syn_1, syn_2, ...syn_n,) \qquad (2.2)$$

where c is the sentiment category (i.e. positive or negative) and w is the unseen word, which is approximated by the synset of positive (or negative) words in WordNet.

Having the lexicon built, Kim [40] proposed 3 different sentiment classification model. The first model is similar to Hu [31]'s work. The second model computes the sentiment strength using harmonic mean:

$$p(c|s) = \frac{1}{n(c)} \sum_{i=1}^{n} p(c|w_i) \qquad (2.3)$$

$$\text{if } \arg\max_j p(c_j|w_i) = c$$

where n(c) is number of opinion phrases whose sentiment category is c. The third model uses the geometric mean:

$$p(c|s) = 10^{n(c)-1} x \prod_{i=1}^{n} p(c|w_i) \qquad (2.4)$$

$$\text{if } \arg\max_j p(c_j|w_i) = c$$

Using these methods, the author is able to achieve around 70% of human-machine agreement on the determined sentiment.

Machine Learning Assisted Sentiment Analysis

Pang [61] studied the performance of using traditional machine learning techniques to perform sentiment analysis in document level. That is, instead of doing feature and opinion extraction, we determine only the overall sentiment of the document. They treat sentiment analysis as a special case of topic-based categorization with positive and negative "topics". Among the three classifiers (Naive Bayesian (NB), Maximum Entropy (ME) and Support Vector Machine (SVM)), SVM performed the best in general, achieving around 80% of accuracy. They also tried different feature extraction (machine learning features) heuristics including unigrams, bigrams, adjectives and combination of them. In general, the unigram heuristics worked the best. They also concluded from their experiments that incorporating term occurrence frequency, bigram, pos tag and word position does not improve the sentiment analysis results.

Turney [78] proposed an interesting idea to perform sentiment analysis. The proposed method is an unsupervised classification method base on web search engines. The idea is that words with positive sentiment will occur more frequently with the term "excellent" and words with negative sentiment will occur more frequently with the term "poor" in the web. Turney defined the Pointwise Mutual Information (PMI) between the two words:

$$PMI(word1, word2) = \log 2 \tag{2.5}$$

base on this definition, the Semantic Orientation (SO) of a phrase can be computed as follows:

$$SO(phrase) = PMI(phrase, \text{"}excellent\text{"}) - PMI(phrase, \text{"}poor\text{"}) \tag{2.6}$$

The PMI between the phrase and the two base words can be approximated by firing web query to search engine and count the number of returned documents.

NLP-based Sentiment Analysis

Since opinion is closely related to text processing and natural languages understanding, it is not rare to see some of the techniques for sentiment analysis are based on Natural Language Processing techniques.

Yi [95] defined a Sentiment Lexicon to contains the sentiment definition of individual words in the following format:

$$< lexical\_entry >< POS >< sent\_category >$$

where lexical_entry is a term (possible multi-word), POS is the required part-of-speech tag and sent_category is the sentiment orientation (i.e. positive or negative). Such a lexicon can be built by some linguistic resources such as WordNet. The authors also defined a Sentiment Pattern Database which contains entries in the following format:

$$< predicate >< sent\_category >< target >$$

where predicate is typically a verb, target is a sentence component (Subject, Object, Prepositional Phrase) the sentiment is directed to. The database is also built by some linguistic resources with manual refinement. With these two databases, the sentiment analysis process becomes the matching of rules in the database. Using this method, the authors are able to achieve an accuracy of above 90% in the testing corpus.

Kanayama [37] proposed an automatic lexicon expansion technique for domain oriented sentiment analysis. Based on the fact that the same word may have different sentiment orientations under different domain, the authors developed the algorithm with the help of syntactic parsing. Similar to the above work, the author extracted over 100 patterns for Japanese Language. They used these patterns to relate features with opinion and assign polarity (sentiment orientation) to them. In this paper, the authors introduced an important concept called Context Coherency. The idea of Context Coherency is simple: With the exception of the appearance of adversative expressions (but, however, even though, although, etc.), polarities of phrases with the same sentence are similar (Intra-Sentential Context Coherency) and polarities of nearby sentences should be similar to each other (Inter-Sentential Context Coherency).

To measure how strong the coherency is, the authors proposed the notion of Coherent Precision, defined as follows:

$$Cp(d, L) = \frac{\#(Coherent)}{\#(Coherent) + \#(Conflict)} \tag{2.7}$$

where #(Coherent) and #(Conflict) are occurrence counts of same and opposite polarities between two nearby phrases. Depending on the definition of "nearby" (the size of the window within that are said to be in the same closure) and the topic domain, the value of cp varies. However, the authors found out that the value of cp is around $72 - 27\%$ across different domain corpus, which suggested that the idea of Coherent Context really make sense.

Similar to Kamayama's idea, Ding [15] derived several linguistic rules to relate sentiment in nearby context. They also suggested that even under the same domain, same words can have different polarity for different features. For example, "long" is a positive description for battery life, while it can be a negative description for focus delay. Taking features into account, they suggested that synonyms and antonyms of words with polarity discovered can also be assigned to the corresponding polarity. Using all these rules, the authors used an algorithm, started with initially known positive and negative sets of word, to expand the sentiment lexicon by iterative infer unknown words' polarity from currently known words.

2.2 Social Computing

Social Computing is a broad topic that covers a lot of areas in computer science. In here, we focus on the two topics that are most related to this thesis, namely, *Social Bookmarking* and *Social Games*.

2.2.1 Social Bookmarking

More and more people are using social bookmarking [88] service nowadays. The problem of bookmark recommendation has drawn significant attention in recent years. Siteseer [67] is one of the early systems in a web page recommendation. It uses users' bookmarks and the organization of bookmark (i.e., folder structures) as hints to predict users' preferences. The idea is to look at the overlapping of bookmarks among different users to determine their pair-wise similarities. Anything else such as website content and title are not being considered. Virtual communities grouping users with similar interest (i.e., high overlapping of bookmarks) are formed based on the user similarities. Finally the recommendations could be done easily by suggesting bookmarks from the same virtual community as the user. While this approach takes bookmark structures into consideration, which implicitly includes some semantic information, it is still a social network based model.

GiveALink [74] is a social bookmarking community for users to store their bookmarks. The site provides website recommendation by analyzing the semantic similarities among website contents. The proposed semantic similarity measure for bookmarks is based on how users organize their bookmark into directories. Collaborative filtering algorithm is then used for the final recommendation.

The Dogear Game [18] takes advantages of the idea of human computation for collaborative bookmark recommendation

Figure 2.2: GiveALink Portal

within a social network. The game is designed to be played by people from the same organization. The game works by asking the player to guess who created the bookmark with the only hints being the title and tags of it. A recommendation is sent to the user whenever a player made a wrong guess that the user is the creator of that bookmark.

Kanawati [36] proposes a multi-agent system in which agents in the system learn users' preferences by their bookmarking behaviors and suggest bookmarks to them. Whenever a user creates bookmarks or accept a recommendation, the agents threat them as positive examples (and vice versa) to train a classifier. The classifier can be used to decide whether new bookmarks should be recommended to the user.

GroupMark [63] is another webpage recommendation system based on users' bookmarks. It is not an automated system that recommendations will not be generated without explicit user requests. This is to allow users to give specify filters and criteria

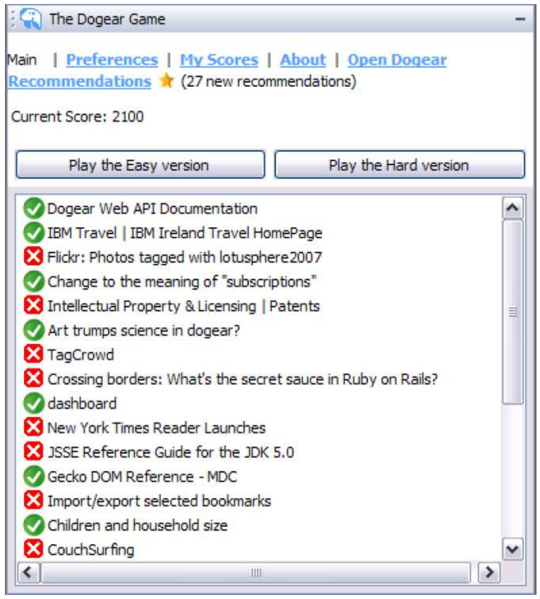

Figure 2.3: DoGear Interface

for recommendation. Although information and collaborative filtering approaches are used, this system is more like a search engine than a recommender system.

There are a few commercially available bookmark recommendation systems on the web. StumbleUpon[1] allow users to rate websites by giving thumb up or thumb down in the browser toolbar [90]. The ratings are used in collaborative filtering process to generate website recommendations. This method relies on the rating giving by users, which is not available in typical bookmarking systems.

inSuggest[2] is another website providing bookmark suggestion based on the del.icio.us bookmarking system. Swimmie[3] is a software plug-in for internet browsers that can suggests website based on users bookmarks in the browser. Both service claims that they analyze users' bookmarks and recommend websites that match their taste. However, the detailed mechanisms of

[1]http://www.stumbleupon.com/
[2]http://bookmarks.insuggest.com/
[3]http://swimmie.jp/en/

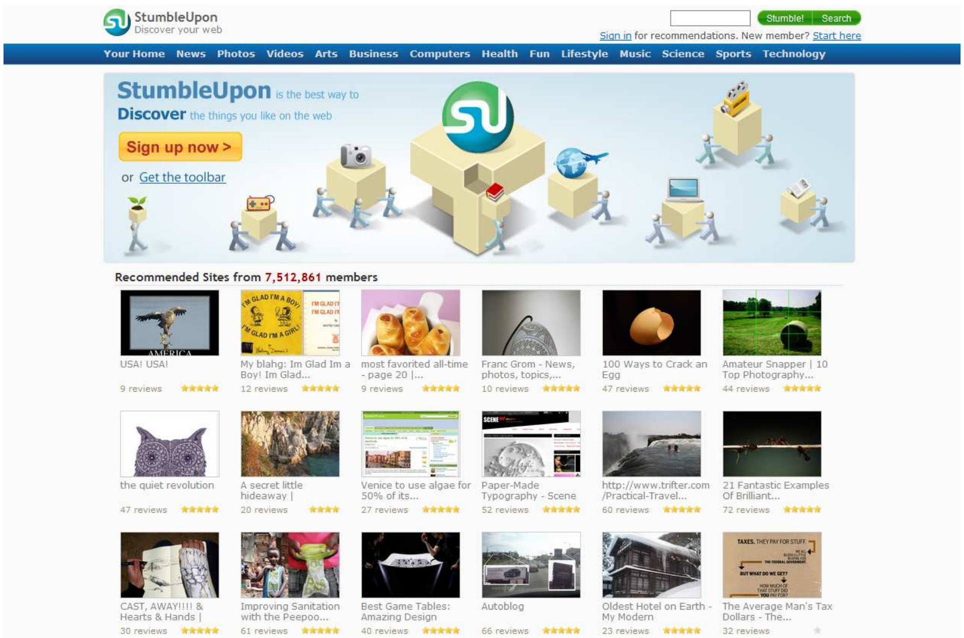

Figure 2.4: StumbleUpon Main Page

how they achieve the goal are unknown to the public.

2.2.2 Social Games

Early, there were a number of projects tried to solve many difficult AI problems through the use of computational power of computers and their users around the world. Examples for collecting commonsense knowledge are Cyc [47], Open Mind [75] [5] and Mindpixel [4], and an example for solving the maximum clique problem is Wildfire wally [62]. All these games either rely on contributions from online volunteers or pay for the engineers to enter information. Therefore, they are unable to scale up the system and cost very high. Besides, they have no mechanism to guarantee that the information collected is accurate.

To encourage more Internet users to provide accurate information to solve the difficult AI problems, social games were proposed to provide entertainment to the online game players, but as a side effect of their playing, accurate information can

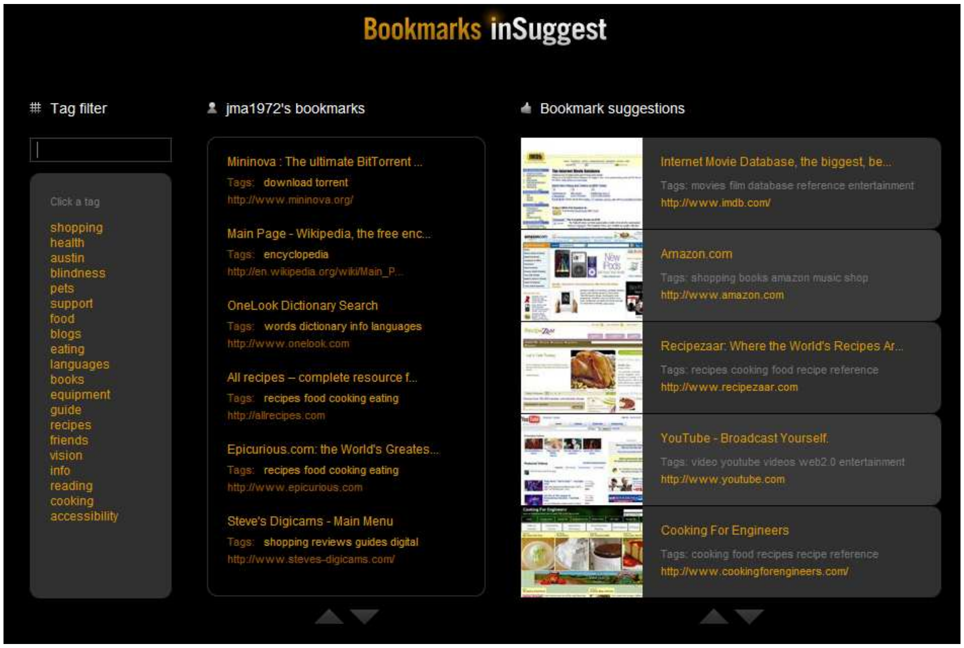

Figure 2.5: inSuggest Bookmark Recommendation

be collected from the players for solving the problems. The idea of social game can be traced back to the CAPTCHA project [79] which was to develop automated mechanisms to tell human beings apart from computer agents. The success of CAPTCHA shows that there are really some problems human can solve easily but computer cannot. This lead to the research of developing new methods, i.e., social games, to effectively utilize human computing power for difficult problems.

Existing social games aim at collecting text information for images or sounds, collecting commonsense knowledge, collecting players' selection from given choices, collecting information for semantic web, collecting bookmark which is an url tagging with descriptive text, collecting personal relationships, collecting assessment results for e-recruiting or collecting human behavior patterns for social robots or for diplomacy. Recently, social games for collecting information through mobile devices and sensors are developed.

Figure 2.6: GWAP Portal

ESP game [80], Peekaboom [84], Squigl [3] and Phetch [82] were proposed to collect text information for images. The objective of ESP game [80] is to collect labels for images on Web. In 2006, Google brought a commercialized online version of the ESP game, the Google Image Labeler [2]. Peekaboom [84] and Squigl [3] aim to label images with all fully annotated with information about what objects are in the given image, where each object is located, and how much of the image is necessary to recognize it. Phetch [82] is to collect explanatory descriptions and sufficient detailed information for images.

Since the commonsense knowledge is so obvious that no one has bothered to record it and the knowledge collected by using search engine may be incorrect and in unstructured format. Verbosity [83], Common Consensus [48] are the social games to

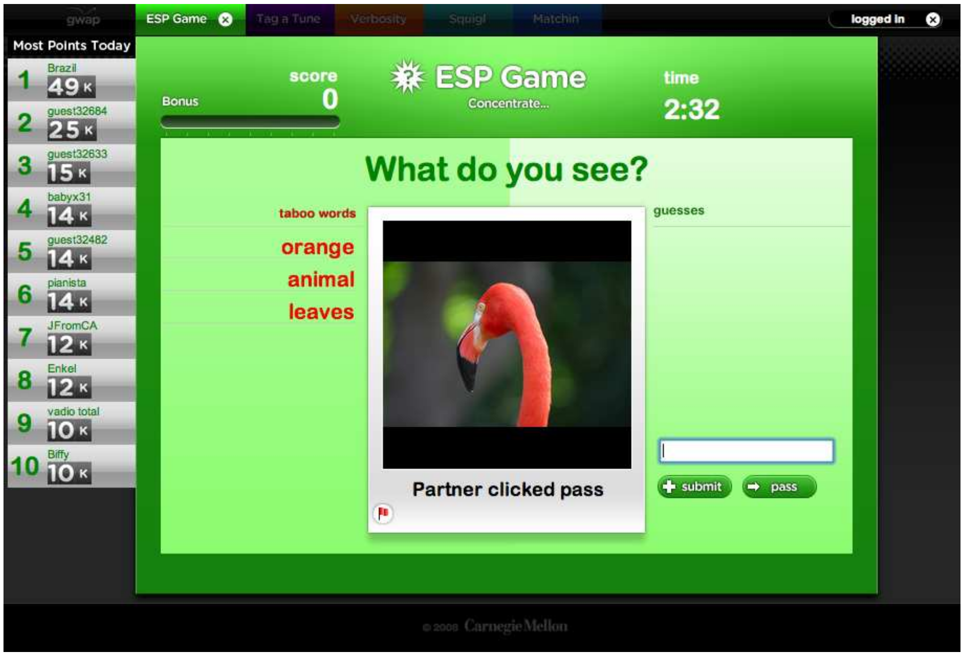

Figure 2.7: ESPGame Game Interface

collect commonsense knowledge in game play. Verbosity [83] aims to collect common-sense statements or facts related to the given word, while Common Consensus [48] both collects and validates common sense knowledge about everyday goals.

Tagatune [45] [44] is an audio-based game that aims to extract subjective descriptions of sounds and music from players. Matchin [3] is a game for collecting players' preference or taste. The two players are shown two images. Each player chooses the image that he thinks his partner will prefer. If the images chosen by two players are matched, they both will gain marks. In [73], it applies human computation to ontology alignment and web content annotation for the Semantic Web using various games of OntoGame, such as OntoPronto, SpotTheLinks, OntoTube, and OntoBay.

There currently exist many social bookmark sites on the Internet, such as del.icio.us [1] [28]. The Dogear Game [19] is a social game that aims to achieve organizational goals which play-

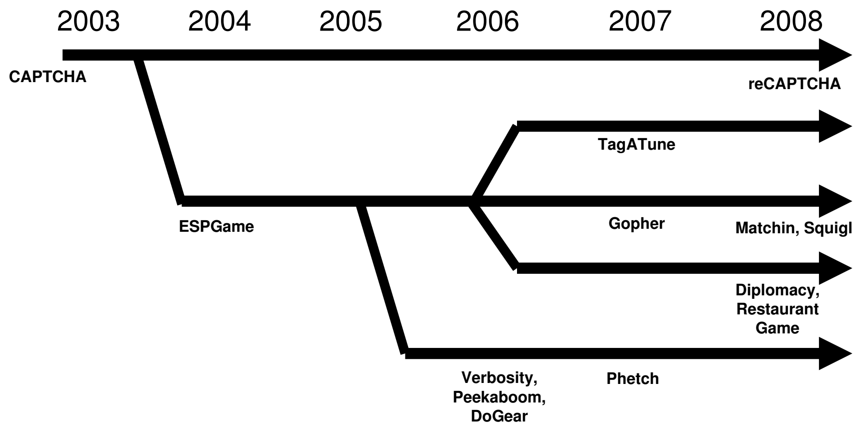

Figure 2.8: History of Human Computation

ers can learn about their colleagues' bookmarks. The Dogear Game uses the bookmarks in an enterprise social-bookmarking system called Dogear [56], which provides a clear association of author to bookmark, to determine the correctness of players' input. Social Heroes [71] is a pervasive social game in which players trade points by tagging each other using Twitter. Social Heroes provides an interface for surrounding personal relationships, identity and communication. CyPRESS [43] is used for e-recruiting, online games to apply for jobs. CyPRESS combines the two approaches of self- and e-assessment. It leads to an improvement of the overall short listing process.

Restaurant Game [59] presents a method of learning human behavior patterns through online gaming. It is a game that players collaborate to create a salad through selecting and discussing available salad items, and the collected data is intended for learning behavior models for autonomous social robots. Diplomacy [41] is a strategic board game with strong emphasis on cooperation and strategizing with opponents for ultimate victory. Players are required to make deals and plan together with their opponents - creating and dissolving alliances from round to round.

The Gopher system [12] employs mobile social gaming for geospatial tagging. Gophers are in-game agents that act as carriers for tasks and proxies to carry information from one player to another. The nature of a task is completely open-ended and predetermined by the player who created the gopher. The Gopher Guessing Game was an early concept prototype that aimed to tag locations in the real world through gameplay within Gophers. The design of the Gopher Guessing Game allows asynchronous matches (so players did not have to be connected at the same time).

The Context-Aware Recognition Survey (CARS) system [91] uses ubiquitous sensors to monitor activities in the home. The contextual information gathered by sensors is used to help users label a multitude of anonymous activity episodes. The CARS system is a game-like computer program in which users attempt to correctly guess which activity is happening after seeing a series of symbolic images that represent sensor values generated during the activity. It allows anyone to label the data at any time, without requiring additional hardware (beyond sensors) or causing additional interruption to daily routine.

Existing social games are casual games. Casual games are designed to be easy to learn, have simple game play, short playing time, and are intended for use by a wide player demographic [14]. There are comparatively low production and distribution costs for the producer. Moreover, they can have any type of game play, and fit in any genre. Since the current social games are developed on an ad-hoc basis without a systematic approach, a formal framework does not exist for designing a social game in general. von Ahn et al. [81] summarized some common properties of current social games and listed out the design principles of current social games. Their study is description-based, but not in a formal framework. In addition, it only considers the existing social games but not social games in general.

In the literature, some frameworks related to human computation problems were studied. The first general human computation framework was proposed in [94]. By adopting a Web 2.0 approach, the framework binds its human computation system, problems providers, participating Web sites and Internet users together to label images and video efficiently but it is not for solving other large-scale human computation problems. Besides, the framework does not convert an AI problem into a game because the conversion is nontrivial and needs to be designed case by case. In [24], the concept of secure distributed human computation was studied. It used basic probability tools to analyze how many malicious parties such a system can tolerate. It also derived design principles for a secure distributed human computation system framework, but it does not consider about the social games for solving problems.

□ **End of chapter.**

Chapter 3

Feature-Opinion Association for Sentiment Analysis

3.1 Motivation

One of the early approaches [31] in sentiment classification depend on a predefined lexicon which contains sentiments of commonly appeared opinion words. This lexicon is then expanded iteratively by adding synonyms and antonyms of the currently known set. The problem of this approach is that sentiments of opinion words are context sensitive. Table 3.1 shows an example of how sentiment of opinion word changes in different context:

Sentence	**Sentiment**
The *picture quality* is **low**!	negative
The *CCD noise* is **low**!	positive

Table 3.1: Context Sensitivity of Opinion Words

Low is a positive term for *CCD noise* while it is a negative term when is used to describe *picture quality*. Therefore, some researchers have proposed to build domain-specific [37] or feature-specific [16] lexicons for sentiment classification. These approaches automatically generate a sentiment lexicon for a particular domain by utilizing some linguistic rules together with a predefined set of sentiment words to infer sentiments of other opinion words.

An accurate and robust Feature-Opinion Association (FOA) method is crucial for both lexicon generation and sentiment classification. That is because more than one feature and opinion word may have mentioned in a sentence. It is not necessary that all opinion words appeared in the sentence are used to describe every feature. An accurate FOA allows us to know what feature an opinion word is describing and thus the correct *sense* of that word can be used to perform sentiment analysis.

Due to the complexity of natural language processing, Feature-Opinion Association is not a trivial task. While the simplest method is to associate the nearest opinion words to each feature, this method can produce wrong results:

"Pictures taken from this compact camera are vivid!"

In this sentence, there are two features, "Picture" and "Camera" (Camera is regarded as one of the feature because people may comment only on camera in some sentence). The opinion words are "compact" and "vivid". However, if we use the above association method, "vivid" can never be matched to "Picture", which is not a desired result.

To solve the above problem, we propose to use a function to compute the relevance score between features and opinion words. We observe that some opinion words are more related to a particular feature than the others. For example, the word "beautiful" is frequently used to describe picture. It is odd to see something like "beautiful battery life". So if we see the sentence "The camera takes beautiful pictures and has a long battery life.", we do not associate the term "beautiful" to "battery life". By performing statistical analysis on a set of topic related documents, we are able determine whether an opinion word is related to the feature in a sentence and thus associating it to the correct feature.

The proposed FOA algorithm makes use of the above observation to match features and opinion words by maximizing the sum of the relevance scores of sentences. We used the algorithm to generate a sentiment lexicon and the built lexicon is used together with our FOA algorithm again to perform sentiment analysis. Experiment results show that our method is useful in improving the sentiment classification accuracy.

3.2 Problem Definition

3.2.1 Definitions

In this section, we give the definitions of *Feature* and *Opinion* in product reviews.

Feature can be a component of the product (e.g. Flash, Lenses), or it can be an attribute of the product (e.g. Weight, Size). In product reviews, people comment on one or more product(s) usually of the same category (e.g. Camera). Products in the same category share a similar set of common features. Since our main focus is on the effect of incorporating Feature-Opinion Association into the sentiment analysis process, we assume that a list of frequently appeared feature of the interested product category is already discovered either manually or by some previous proposed methods [95, 31].

Opinion words in general can be anything that is used to describe a feature. However, due to the difficulties in natural language understanding, it is not easy to perform sentiment analysis on all types of opinion. As with most of the existing works, we limit our scope to handle opinion words that are in the from of adjectives and adverbs.

3.3 Closer look at the problem

To perform sentiment analysis at sentence level, we consider sentences that contain at least one feature and one opinion word. With the features and opinion words defined as above, we can divide them into 6 categories:

Category 1 – Exact Match: This is the simplest case where one opinion word is matched exactly to one corresponding feature:

"The focus is correct and the picture is clear."

In this sentence, "correct" is associated to "focus" and "clear" is associated to "picture".

Category 2 – No Associated Opinion: It is not unusual that some sentences are not directly commenting on a feature and thus, there are no associated opinion word for it:

"The picture quality is affected by the bright flash."

In this sentence, "picture quality" does not have any opinion words associated to it.

Category 3 – No Associated Feature: On the other hand, some opinion words are not associated to any features:

"The camera is easy to use even for young kids."

The word "young" is not associated to any camera features.

Category 4 – Multiple Opinion Words for One Feature: Some features have more than one opinion words describing it:

"I like this great little camera."

The "camera" is described by "great" and "little".

Category 5 – Sharing Opinion Words: Multiple features share the same opinion word:

"Excellent zoom lenses and flash!"

The word "Excellent" is used to describe both "lenses" and "flash".

Category 6 – Combined Case: This is the most general case where any combination of the above situations can occur together:

"Except the large and heavy battery, everything like zoom lenses and flash are excellent!"

3.3.1 Discussion

From the above examples, we can see that there are no fixed patterns between the appearance of features and opinion words. An ideal Feature-Opinion Association algorithm should allow sentiments of each feature predicted under all of the above cases.

3.4 Proposed Approach

We define the Feature-Opinion Association Problem (FOA) as follows:

Given a sentence that contains a non-empty set of features $F = \{f_0, f_1, ..., f_n\}$ and opinion words $W = \{w_0, w_1, ..., w_m\}$, match the product feature with its related opinion words that the following function is maximized:

$$foa(F, W) = (\sum_{i=0}^{n} \sum_{j=0}^{|matched(fi)|} rel(f_i, matched(f_i)(j)) \qquad (3.1)$$

In the equation, $matched(f)$ is the set of opinion words matched to feature f and $matched(f)(i)$ is the i-th opinion word in the set. The function $rel(f, w)$ returns the relevance score of opinion word w to feature f. The key to this association problem is to define a good $rel(f, w)$ function.

Under this model, each opinion word will be matched to the feature that has the highest rel score with it. That means each opinion word must be matched to one of the feature. From the section above, we know that some of the opinion words are not associated to any features *(Category 3 – No Associated Feature)*, therefore in the actual FOA algorithm, we set a threshold value th so that opinion words and features with rel score lower that th will not be matched together.

For the case where multiple feature shares the same opinion word *(Category 5)*, the FOA algorithm will still assign the shared opinion words to only one of the feature (i.e. the feature with the highest rel score). Since the opinion words used in this case will be more general (e.g. Good, Excellent) rather than feature-specific (e.g. vivid, colorful), we can obtain the sentiment of the associated feature easily. As for the remaining features, their sentiments can be inferred by the overall sentiment of the sentence.

3.4.1 Nearest Opinion Word (DIST)

The simplest solution to the FOA problem is to associate opinion words that are nearest to features. In this case, $rel(f, w)$ is defined as the inverse of the distance between the opinion word w and feature f, as follows:

$$rel(f, w) = \frac{1}{dist(f, w)}. \tag{3.2}$$

Where $dist(f, w)$ is the distance between word f and w.

3.4.2 Co-Occurrence Frequency (COF)

Another approach to the FOA problem is to define $rel(f, w)$ as the Co-Occurrence Frequency (COF) between feature f and opinion word w. The Co-Occurrence Frequency between word a and b is defined as the number of times a and b appears in the same sentence in the training corpus. The intuition is that the higher the COF score, the more likely that the opinion word is related to the feature. To illustrate the idea, we have selected three features of camera, namely "Picture Quality", "Feature" and "Zoom" and list out the top 10 opinion words with the highest COF scores from the corpus data set (Section 3.5.1):

Feature	**Opinion Word**
Photo Quality	good, excellent, great, better, really, best, superb, low, poor, just
Feature	great, good, easy, excellent, really, nice, just, digital, better, best
Zoom	optical, good, digital, great, better, wide, really, just, long, excellent

Table 3.2: Opinion Words sorted by COF scores

The problem of COF is that words like "good", "great", "excellent" dominate the top of the rank. They are ranked so high because they are very commonly used as opinion and appear very frequently in the corpus. While we cannot say that they are totally irrelevant, they are by no means specific to the feature.

3.4.3 Co-Occurrence Ratio (COR)

To get a more feature-specific ranking, we should also take into accounts the corpus frequency (CF) (i.e. the number of times a term appears in the corpus) of opinion words. For example, if an opinion word w appears in the corpus for 100 times ($CF = 100$) and its COF with feature f is 80 ($COF(f, w) = 80$) then we say that the Co-Occurrence Ratio (COR) between f and w is 0.8. In this case, the rel function becomes:

$$ref(f, w) = \begin{cases} \frac{COF(f,w)}{CF(w)} & \text{if COF(f,w)} > \text{threshold} \\ 0 & \text{otherwise} \end{cases}$$

It is possible that the corpus contains some terms that have a very low corpus frequency (e.g. rare terms or misspelled terms). They can get a high COR rank even if they just occasionally appeared with the feature for 1 or 2 times. Therefore, a threshold is used to filter out this type of terms. Table 3.3 summarizes the results using the COR measure, opinion words that appeared at the top of the rank are more feature-specific.

3.4.4 Likelihood-Ratio Test (LHR)

Using method proposed by [95], assuming the association between feature and opinion word is a Bernoulli event and it follows a binomial distribution, we can compute the likelihood ratio

Opinion Word	COF	CF	COR
equal	21	88	0.239
superb	152	741	0.205
outstanding	95	472	0.201
satisfactory	12	64	0.188
superior	45	250	0.18
absolute	19	107	0.178
comparable	27	167	0.162
excellent	522	3661	0.143
highest	31	244	0.128
poor	138	1090	0.127
Opinion Word	COF	CF	COR
rich	57	121	0.471
unique	34	103	0.330
neat	27	107	0.252
wise	20	102	0.196
interesting	21	108	0.194
cool	54	279	0.194
accessible	16	85	0.188
rapid	11	69	0.159
favorite	17	108	0.157
extensive	16	105	0.152
Opinion Word	COF	CF	COR
optical	654	1175	0.557
wider	29	132	0.217
mechanical	23	109	0.211
distant	13	62	0.210
equivalent	41	196	0.209
maximum	37	186	0.199
smooth	28	172	0.163
wide	191	1205	0.159
focal	41	261	0.157
quite	15	101	0.159

Table 3.3: Opinion Words sorted by COR scores with “Photo Quality”, “Feature” and “Zoom”

as follows:

$$-2\log\lambda = \begin{cases} -2 * lr & \text{if } r_2 < r_1 \\ 0 & \text{if } r_1 < r_2 \end{cases}$$

where

$$\begin{aligned} C_{11} &= COF(f,w) \\ C_{12} &= CF(w) - COF(f,w) \\ C_{21} &= CF(f) - COF(f,w) \\ C_{22} &= SentenceCount - C_{11} - C_{12} - C_{21} \end{aligned}$$

$$\begin{aligned} r_1 &= \frac{C_{11}}{C_{11} + C12} \\ r_2 &= \frac{C_{21}}{C_{21} + C_{22}} \\ r &= \frac{C_{11} + C_{21}}{C_{11} + C_{12} + C_{21} + C_{22}} \\ lr &= (C_{11} + C_{21})\log(r) + (C_{12} + C_{22})\log(1 - lr) \\ &- C_{11}\log(r_1) - C_{12}\log(1 - r_1) - C_{21}\log(r_2) \\ &- C_{22}\log(1 - r_2) \end{aligned}$$

The likelihood ratio $-2\log\lambda$ is used as rel score in this case. i.e.,

$$ref(f,w) = -2\log\lambda \tag{3.3}$$

.

A larger value represents a higher chance that opinion word w should be associated with feature f.

3.4.5 Combined Method

The measures discussed above can be roughly divided into two types. Nearest Opinion Word is a context based distance mea-

surement while the others try to perform associate by computing the relevance score using the corpus based statistics. Although distance measurement is an intuitive method, it does not account for the feature and opinion relationship. Relevance measurements do take this relationship into account but does not care about the actual sentence structure. Therefore we combine these two types of method together as follows:
Co-Occurrence Ratio and Nearest Opinion Word (COR+DIST):

$$rel(f,w) = \frac{COF(f,w)}{CF(w) * dist(f,w)}. \tag{3.4}$$

Likelihood-Ratio Test and Nearest Opinion Word (LHR+DIST):

$$rel(f,w) = \frac{-2\log\lambda}{dist(f,w)}. \tag{3.5}$$

3.4.6 Feature-Opinion Association Algorithm

For each review sentence S containing at least one feature and one opinion word, the Feature-Opinion Association Algorithm (FOAA) associate opinion words to features subject to the *rel* function as follows:

In general, each opinion word will be to associate to the feature with the highest *rel* score except in the following scenario:

"Good lenses, good pictures!"

Depending on the *rel* function used, it is possible that the score for the same pair of feature and opinion word is the same for different appearances of the same opinion word. The algorithm will check if the same opinion word is already associated to a feature, and if so, it tries the next feature until it eventually

Algorithm 2 Feature-Opinion Association Algorithm

$F \leftarrow$ features in S
$W \leftarrow$ opinion words in S
for each w in opinion word list W **do**
  $score \leftarrow$ highest $rel(f, w)$ for all $f \in F$
  if $score \geq threshold$ **then**
    if the same word is already assign to f **then**
      Try another f with the next highest foa score
    else
      associate w to f
    end if
  end if
end for

finds one. A threshold value is used to prune opinion words that have low *rel* scores to all features appeared in the sentence.

3.4.7 Sentiment Lexicon Expansion

With the Feature-Opinion Association algorithm, steps for sentiment lexicon expansion become straightforward. Two sets of opinion words (positive and negative) are defined initially as seeds. The FOA algorithm is used to associate features and opinion words for each sentence appeared in the training corpus. Opinion words are attached with a tag indicating their associated features. Same opinion words carrying different feature tags (meaning that they are of different *sense*) are treated as two different words in the sentiment analysis process. Using the linguistic rules proposed by [15], we count, for each uniquely tagged-opinion words, the number of times it is in conjunction with the two known sets. Then we compute the orientation score:

$$orientation(w) = \frac{c_{+ve} - c_{-ve}}{CF(w)}, \tag{3.6}$$

where c_{+ve} and c_{-ve} is the number of times word w is in conjunction with the known positive and negative set respectively.

The score is normalized by its corpus frequency. The higher the score, the more likely that the word should be in the positive set and vice vera. Words with absolute score smaller than a threshold T should not be treated as either polarity. In each iteration, words with the highest and lowest score are added to the two sets respectively. The algorithm terminates when there are no more opinion words left or none of the remaining opinion words meet the threshold requirement.

3.5 Evaluation

In order to verify our ideas, we collected two data sets from the internet. These two data sets are used to conduct our experiments. NLTK [58] is used perform to natural language processing tasks such as sentence splitting and part-of-speech (POS) tagging.

3.5.1 Corpus Data Set

User reviews of all cameras of popular brands are crawled from Digital Photography Review [17]. This data set contains 400+ different camera models, 17000+ user reviews and 250000+ sentences. Each review contains two parts, namely, opinion and problems. Both parts are extracted to form this data set. We name it as the Corpus Data Set because it is used as a statistical database for computing the relevance scores discussed in Section 5.3.

3.5.2 Test Data set

We used another publicly accessible data set [31, 15] for testing. Only the 4 camera reviews are used. The reviews are re-tagged manually based on our own feature list. Each camera review sentence is attached with the mentioned features and their associated opinion words. For example, the following sentence:

"Very comfortable camera, easy to use, and the best digital photos you're going to get at this price."

will receive the tags: [+Camera(comfortable,easy)] [+Picture(best,digital)] [+Price()]. That means this sentence contains 3 features that exist in our pre-discovered feature list. It is important to note that we group synonyms (i.e. picture and photo) together so that they are treated as the same feature. The "+", "/" or "?" sign in front of the feature name indicates its opinion orientation. Words in the brackets are those we found to be associated with the corresponding features.

3.5.3 Feature-Opinion Association Accuracy

For each tagged sentence in the test data set, we use the FOA algorithm to match appeared features and opinion words. The association results are compared to human tags. To favor the evaluation process, we have defined a feature-based comparison matrix as shown in Table 3.4.

		Human Tags	
		+	-
FOA	+	C_{AA}	C_{NA}
	-	C_{AN}	C_{NN}

Table 3.4: Feature-based Comparison Matrix

For each feature, there will be a list of opinion words tagged by human and the FOA algorithm respectively. In the table, the "+" sign indicates an opinion word is associated to that feature by Human / FOA algorithm and the "-" sign carries the opposite meaning. C_{AA} counts the number of opinion words that are both tagged by human and the FOA algorithm to the same feature. C_{AN} counts the number of opinion words that are tagged by human but not by the FOA algorithm. The meaning of C_{NA} and C_{NN} are defined in similar manner.

Under this comparison matrix, we can see that the goal of FOA is to maximize C_{AA} and C_{NN} and minimize the C_{AN} and C_{NA}. This is because C_{AA} and C_{NN} represent human-machine agreement and C_{AN} and C_{NA} represent disagreement. However, for the purpose of sentiment classification, association errors introduced by high C_{NA} counts are more acceptable than high C_{AN} counts. That is because the words that are mis-associated by the FOA algorithm should be less relevant to the feature and thus they are likely to carry neutral sentiment. This will have less impact on the sentiment analysis results. On the other hand, high C_{AN} counts indicates that the FOA algorithm fails to associate many important opinion words to features. That will seriously affect the sentiment classification process as it is highly dependent on the existence of opinion words.

The association accuracy can be computed by the traditional precision, recall and F-score:

$$Precision = \frac{C_{AA}}{C_{AA} + C_{NA}}, \tag{3.7}$$

$$Recall = \frac{C_{AA}}{C_{AA} + C_{AN}}, \tag{3.8}$$

$$FScore = \frac{2 * Precision}{Precision + Recall}. \tag{3.9}$$

In terms of FOA, precision is the percentage of opinion words that are correctly associated to features and recall is the percentage of correct opinion words that can be find by the FOA algorithm.

We test the accuracy of FOA using each *rel* function with a range of possible threshold *th* values. For COR, DIST and COR+DIST, the range is from 0 to 1 with step size 0.01. Range

Data Set	DIST				COF				COR			
	th	precision	recall	F-score	th	precision	recall	F-score	th	precision	recall	F-score
Camera 1	0.34	0.72	0.56	0.63	4.00	0.45	0.77	0.57	0.01	0.46	0.74	0.57
Camera 2	0.05	0.48	0.90	0.63	0.00	0.48	0.77	0.59	0.01	0.49	0.71	0.58
Camera 3	0.13	0.50	0.77	0.60	2.00	0.44	0.71	0.54	0.02	0.45	0.70	0.54
Camera 4	0.06	0.47	0.89	0.62	1.00	0.47	0.71	0.57	0.01	0.47	0.63	0.54
Average	0.20	0.58	0.64	**0.61**	1.00	0.45	0.75	**0.57**	0.01	0.46	0.71	**0.56**

Data Set	LHR				COR+DIST				LHR+DIST			
	th	precision	recall	F-score	th	precision	recall	F-score	th	precision	recall	F-score
Camera 1	46.00	0.60	0.61	0.61	0.03	0.64	0.65	0.64	9.00	0.65	0.67	0.66
Camera 2	5.00	0.53	0.76	0.62	0.02	0.65	0.66	0.65	23.00	0.80	0.61	0.70
Camera 3	0.00	0.47	0.74	0.58	0.03	0.61	0.60	0.60	1.00	0.54	0.74	0.62
Camera 4	0.00	0.52	0.77	0.62	0.00	0.42	0.84	0.56	0.00	0.53	0.77	0.63
Average	0.00	0.49	0.79	**0.60**	0.02	0.59	0.65	**0.62**	9.00	0.66	0.64	**0.65**

Table 3.5: Accuracy of FOA using different *rel* functions

for other functions is 0 - 100 with step size 1. The results of individual data set with the best F-Score is presented in Table 3.5.

Non-Combined *rel* Functions: We first analyze the FOA results that use non-combined *rel* functions. The results show that most *rel* functions, when used to perform FOA, are capable of achieving good recalls (around $65-79\%$). However, their precisions are generally quite low. This suggests that using these functions alone are not effective in pruning away the non-feature related opinion words. Among all the non-combined *rel* functions, LHR and DIST performed the best, reaching an average F-Score of 60%. An important observation is that DIST has the highest average precision and LHR has the highest average recall. That means opinion words are usually associated to features that are nearest to them. But there are also considerable amounts of opinion words appear far away from features they describe.

Combined *rel* Functions: The combined *rel* functions, in general, perform better in terms of the F-Score measurement when used in the FOA algorithm. The LHR+DIST approach is able to reach a precision of over 80% and a F-Score of 70% in data set 2. The average precision, recall and F-Score are maintained at around 65% which are definitely improvements over other the non-combined methods. COR+DIST is slightly inferior to LHR+DIST but still achieved an improved overall F-Score in our experiment. The results here indicate the importance of using both types of measurement in the feature-opinion association process. Missing either information will lead to worse association results.

Sentiment Classification Accuracy: Using the method discussed in section 3.4.7, we generate 3 sentiment lexicons. They represent the cases where LHR, COR+DIST and LHR+DIST are used in the FOA part of the lexicon generation process. The reason for choosing these 3 is that COR+DIST and LHR+DIST performed the best in terms of Precision and F-Score while LHR achieved the best average recall in the FOA process. For each of these methods, we use the FOA threshold that produces the best average F-Score in Table 3.5. Opinion words {"excellent", "good"} and {"poor", "bad"} are used as the initial seed words for the positive and negative sentiment respectively. These words are chosen because their sentiments are less sensitive with respect to the feature they describe. They can be safely assumed to be always positive (or negative) in the lexicon. In our experiment, the orientation threshold T for lexicon generation is set to 0.2.

We conduct two set of experiments with identical settings except that one includes FOA while the other does not. The sentiment classification process is as follows: For each tagged sentences, all the appeared features and opinion words are extracted. The algorithm computes the sentiment score for each feature mentioned in the sentence solely based on the associated opinion words. Features that are not associated to any opinion words will have their sentiments inferred using two different methods. The first method (Human and FOA) uses the majority sentiment of other features that appeared at the same sentence. The second method (Human* and FOA*) falls back to use all opinion words of the same sentence to infer sentiments of these features. This is the same as the case where FOA is not used. Under both cases, we use the opinion aggregation function [15] for sentiment scoring. Positive words have a score of +1 and negative words have a score of -1. Sentiments of opinion words are retrieved from sentiment lexicons generated in the above steps. The predicted sentiment orientations are compared against human tags to calculate accuracies. Table 3.6 summarizes the results of our sentiment classification experiments.

		LHR Lexicon			COR+DIST Lexicon			LHR+DIST Lexicon		
	rel **function**	LHR	COR+DIST	LHR+DIST	LHR	COR+DIST	LHR+DIST	LHR	COR+DIST	LHR+DIST
Camera 1	All	0.745			0.656			0.668		
	Human	0.761			0.692			0.719		
	Human*	0.772			0.682			0.710		
	FOA	0.692	0.638	0.682	0.653	0.593	0.632	0.685	0.634	0.662
	FOA*	0.755	0.744	0.760	0.686	0.684	0.686	0.708	0.681	0.695
Camera 2	All	0.703			0.669			0.695		
	Human	0.740			0.700			0.740		
	Human*	0.749			0.706			0.748		
	FOA	0.685	0.650	0.657	0.652	0.596	0.587	0.694	0.643	0.647
	FOA*	0.729	0.743	0.736	0.709	0.695	0.682	0.739	0.735	0.736
Camera 3	All	0.799			0.761			0.672		
	Human	0.797			0.736			0.709		
	Human*	0.845			0.791			0.721		
	FOA	0.785	0.724	0.721	0.695	0.651	0.625	0.689	0.623	0.656
	FOA*	0.828	0.836	0.843	0.761	0.767	0.759	0.724	0.709	0.716
Camera 4	All	0.663			0.667			0.626		
	Human	0.656			0.670			0.660		
	Human*	0.653			0.677			0.667		
	FOA	0.629	0.579	0.606	0.619	0.557	0.589	0.612	0.558	0.585
	FOA*	0.694	0.684	0.673	0.697	0.687	0.687	0.646	0.636	0.636
Average	All	0.727			0.688			0.665		
	Human	0.739			0.699			0.707		
	Human*	0.755			0.714			0.711		
	FOA	0.698	0.648	0.666	0.655	0.599	0.608	0.670	0.614	0.638
	FOA*	0.752	0.752	0.753	0.713	0.708	0.704	0.704	0.690	0.696

Table 3.6: Sentiment Classification Accuracy

Classification results using human tags:

We first compare the sentiment classification accuracies of all opinion words (*All*) and human association (*Human* and *Human**). We can see that the overall accuracies increase if we limit the sentiment classifier to use only opinion words that are tagged by human. These agree with our intuitions that blindly using all opinion words actually produces false results.

An interesting observation is that *Human** performs better than *Human*. The reason is that a sentence usually mentions only 1 or 2 features. When there are no associated opinion words, we either cannot find another feature, or the remaining features are not enough to help inferring its sentiment correctly. Falling back to use all opinion words actually helps in this case. We can consider *Human** as an improved version of the all opinion words method. It tries limit itself to use a subset of opinion words that are most relevant to a feature to improve the sentiment classification accuracy whenever possible.

Classification results using FOA algorithm

Solely using opinion words that are associated by the FOA algorithm for sentiment analysis (FOA in Table 3.6) actually produces poorer results. This is reasonable, given that the accuracies of using human FOA alone (*Human*) are just slightly better than the case where all opinion words are used. However, if we use the second method to deal with the case where no opinion words are associated to a feature (FOA*), the overall accuracy improves and it consistently outperform *All* in our experimental data sets. This suggests that our method is effective in improving the sentiment classification results.

Effects of using different *rel* measurements

We study how different *rel* measurements affect the generated sentiment lexicon as well as the accuracy of sentiment analysis. We pair each of the selected *rel* function with the generated lexicons one by one and compare the sentiment classification results.

We observe that using LHR to perform FOA actually helped to generate the best sentiment lexicon. The reason is that LHR is good at achieving high recalls. It extracts most of the feature-related opinion words in the lexicon building process. Although its precision is not high, the incorrectly associated opinion words are unlikely to be inferred to carry a sentiment because of the threshold limitations. Finally, despite of the fact that LHR generated the best sentiment lexicon, all three *rel* functions performed very similarly during sentiment analysis. This indicates a good sentiment lexicon is important to obtain accurate results. Lose in sentiment analysis accuracies due to poor lexicons cannot be compensated by a good FOA algorithm later on.

3.6 Summary

In this chapter, we study the Feature-Opinion Association (FOA) problem in the sentiment classification process. We propose a novel algorithm to perform the association and suggest a set of possible relevance functions for the algorithm. In contrast to the existing sentiment classification approaches, our work enables the sentiment classifier to pick the opinion terms selectively based on their relevance with the feature in question. We evaluate our work with a publicly available dataset and the result shows that our work is capable of achieving better performance.

Chapter 4

Social Game for Opinion Mining

4.1 Motivation

Human computation is a technique that makes use of human abilities to solve problems. The existence of this technique is because there are always some important problems that are very difficult for computer to solve even with the state-of-the-art technologies.

Image annotation is one of the problems that computer cannot solve easily. Given an image, we would like the computer to be able to tell a few things about it such as the object inside the image, the theme, the place it was taken (in case of photographs), etc. Search engines have been looking for ways to gather this information so that they can give better search results. However, even with the best image process algorithm nowadays, this is still a task that is next to impossible.

However, with the help of human computation, these tasks can be done very efficiently. ESPGame [80] is a social game that asks two anonymous players from the web to annotate images in order to compete for higher scores. The whole idea is simple but the end result is that we are now able to annotate the images on the web without either replying on the unreliable image processing algorithms or paying people to annotate them. Studies have shown that these types of problems, when packaged in the form of social games, could produce results that are of high accuracies.

Opinion mining has been a difficult problem for computers because it involves natural language processing. Various techniques developed have the limitations such as domain dependency, insufficient depth of analysis and so on so forth. The nature of this problem aligns with the prerequisites of human computation, it is difficult. As a result, we would like to design a game that could improve upon existing works.

In this chapter, we first analyze exist social games. We use a mathematical model to extract the common properties of these games. We derive guidelines that can help the design of new social games. Based on the framework, we design two games, namely, OpinionMatch and FeatureGuess for opinion mining. The design of OpinionMatch is straight-forward in a sense that players are asked to label the subjectivity of a passage. FeatureGuess, on the other hand, focuses on solving the Feature-Opinion Association problem in opinion mining. This chapter concludes with a discussion on the advantages and disadvantages of each game.

4.2 Social Game Model

In this section, we propose a formal model for social games. By extracting the common properties of existing games, we analyze their relationships with the characteristic of the human computation problems that they aim at solving, and design a formal framework for social games of solving problems in general. It also shows how existing social games can be presented in our model.

4.2.1 Definitions

Before proceeding further, we start with the definition of data and the definition of problem domain. It considers all general data types. Next, we define the social game problem. After that, we provide a set of definitions for social game framework.

Definition 1. *A data $\mathcal{D}$ is an object with a data type $\mathcal{T}$ and a set of attributes denotes as $\mathcal{A}$:*

$$\mathcal{T} \in \{text, image, video, sound, URL\}$$

$$\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_{\mathcal{X}})$$

where the date type $\mathcal{T}$ is the media type presented by $\mathcal{D}$; and each attribute $\mathcal{A}_{\mathcal{X}}$ has a relationship $Rel(\mathcal{A}_{\mathcal{X}})$ and a set of value $\mathcal{V}(\mathcal{A}_{\mathcal{X}}) = \{\mathcal{V}_1(\mathcal{A}_{\mathcal{X}})$, $\mathcal{V}_2(\mathcal{A}_{\mathcal{X}})$, ..., $\mathcal{V}_Y(\mathcal{A}_{\mathcal{X}})\}$; and each value $\mathcal{V}_Y(\mathcal{A}_{\mathcal{X}})$ is an object with its own data type and contains its set of attributes. $\mathcal{V}_Y(\mathcal{A}_{\mathcal{X}})$ is also called metadata of data $\mathcal{D}$.

Definition 2. *A **social game** is a 4-tuple ($\mathcal{SGPD}$, $\mathcal{GR}$, $\mathcal{GF}$, $\mathcal{ANS}$), where sets:*

1. *$\mathcal{SGPD} = (\mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{C})$ is the social game problem domain.*
   (a) *$\mathcal{E} = \{e_i | i = 1, ..., x\}$ is a set of problems that we want to solve where the problem e_i is to collect metadata of an input data $\mathcal{D}$.*
   (b) *$\mathcal{F} = \{f_i | i = 1, ..., y\}$ is the answer domain. Solutions to any $e_i \in \mathcal{E}$, which f_i is a value of an attribute of $\mathcal{D}$ that we want to collect, can only exist in $\mathcal{F}$.*
   (c) *$\mathcal{G} : \mathcal{E} \times \mathcal{F} \rightarrow \Re \in [0..1]$ is a function that determine whether an answer is correct to a problem.*
   (d) *$\mathcal{C}$ is a set of constraints in the game that*

i. *indicating the attribute(s) we want to collect such that* $\mathcal{A}_{\mathcal{X}} \in \mathcal{A}$*;*

ii. *indicating the set of values that we want to collect within* $\mathcal{V}(\mathcal{A}_{\mathcal{X}})$*.*

2. $\mathcal{GR} = (\mathcal{D}, \mathcal{M}, \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{I}, \mathcal{O}, \mathcal{V}, \mathcal{W})$ *represents rules of a social game.*

   (a) $\mathcal{D}$ *is input data that we want to collect its metadata.*

   (b) $\mathcal{M} = \{m_i | i = 1, ..., x\}$ *is a set of metadata which are the values of attributes of* $\mathcal{D}$ *that we want to collect.*

   (c) $\mathcal{C}$ *is a set of constraints in the game that*

   i. *indicating the attribute(s) we want to collect such that* $\mathcal{A}_{\mathcal{X}} \in \mathcal{A}$*;*

   ii. *indicating the set of values that we want to collect within* $\mathcal{V}(\mathcal{A}_{\mathcal{X}})$*.*

   (d) $\mathcal{R} = \{r_k | k = 1, ..., nR\}$ *is the set of roles that players could have during a game.*

   (e) $\mathcal{P}(r_k) = \{p_j^k | j = 1, ..., n\mathcal{P}(r_k)\}$ *is the set of players that are assigned to the role* r_k *during a game.*

   (f) $\mathcal{I}(p_j^k) = \{i_m^{k,j} | m = 1, ..., n\mathcal{I}(p_j^k)\}$ *is the set of input given to the player* p_j^k *for solving the problem of input* $\mathcal{D}$ *during a game.*

   (g) $\mathcal{O}(p_j^k) = \{o_m^{k,j} | m = 1, ..., n\mathcal{O}(p_j^k)\}$ *is the set of output provided by the player* p_j^k *for solving the problem of input* $\mathcal{D}$ *during a game.*

   (h) $\mathcal{V}()$ *is a procedure that determines whether players have produced outputs that meet specific requirements within a game segment. If so, return a possible answer* $f \in \mathcal{F}$*.*

   (i) $\mathcal{W}(p_j^k)$ *is the reward that the player can receive for solving the problem of input* $\mathcal{D}$ *during a game where*

$\mathcal{W}(p_j^k) \in \{w_i | i = 1, ..., y\}$. *Players will receive a reward when achieving the winning condition of the game.*

3. $\mathcal{GF} = \{pSel, eSel, tMax, pNum, \mathcal{GM}, \mathcal{UI}\}$ *represents the flow of a social game.*

   (a) *pSel() is a procedure that selects players to play a game and assigns roles to them.*

   (b) *eSel() is a procedure that picks a problem from the problem set.*

   (c) $tMax$ *is the maximum duration of a game.*

   (d) $pNum$ *is the number of players of a game. It may be a single-player game, two-player game or multi-player game.*

   (e) $\mathcal{GM} \in \{$*collaborative, competitive, hybrid*$\}$ *is the mechanism of a game.*

   (f) $\mathcal{UI} = \{ui_j | j = 1, 2, ..., x\}$ *is the set of design characteristics of user interface.*

4. $\mathcal{ANS} = (\xi, \tau)$ *represents answer extraction. It defines how answers are generated for each problem based on all the games played.*

   (a) ξ *is a data structure that supports the following operations:*

      i. ***add()*** *takes* $e \in \mathcal{E}$ *as input and updates its internal counters.*

      ii. ***count()*** *returns the internal count for a particular* $f \in \mathcal{F}$

   (b) τ *is a frequency threshold for accepting an answer.*

Definition 3. *An action (AC) is a 2-tuple* $(\mathcal{ACT}, \mathcal{ACO})$ *where sets:*

1. *$\mathcal{ACT}$ is the type of an action.*
2. *$\mathcal{ACO} = \{aco_i | i = 1, ..., x\}$ is the outcome domain of an action. It specifies the possible output values of the action.*

Definition 4. *A role (R) is a 2-tuple $(\mathcal{KW}, \mathcal{ACS})$ where:*

1. *$\mathcal{KW}$ is the knowledge a role can has.*
2. *$\mathcal{ACS} = \{acs_i | i = 1, ..., x\}$ is the set of actions that can be performed by the role R where acs_i is an action.*

4.2.2 Social Game Problem

To define a social game problem, we start with the definition of data. Each data is an object with a data type, which the data is presented in text, image, video, sound or URL format. Each data has a set of attributes, while each attribute has a value and each value is a data. For instance, a picture is an object of image type. A picture has two attributes, they are *label* and *description*. For label attribute, it has a set of values in text format. For description attribute, it has a set of values in text format.

A social game problem is to collect the values of some specific attributes of an input data. These values can be called the metadata of the input data. There exists a set of constraints on the metadata to collect, such as what the specific attributes are and what values have to be excluded.

4.2.3 Social Game Flow

A social game includes a social game problem domain, rules of game, game flow and the procedure of answer extraction after collecting a set of answers in a number of rounds of game. A game refers to a match played by a set of players inside a social gaming system. The flow of a game is defined as follows.

1. Select players and assign roles to them by $pSel()$.
2. Find a problem from $\mathcal{E}$ to play by $eSel()$.
3. Collect outputs $\mathcal{O}$ from players' actions.
4. If verification $\mathcal{V}()$ is not passed, repeat step 3.
5. If time used time limit $\leq tMax$, repeat step 2.
6. Increase the reward of players by f.

Step 2-3 is called a segment during the game. It corresponds to the period of time when players are working on a particular problem e. While players' actions pass the verification procedure $\mathcal{V}()$, the game proceeds to another segment and players work on the next problem.

4.2.4 Answer Extraction Procedure

Answer extraction procedure in a social game is responsible for generating answers to each problem based on all the games played in the system. The actual procedure is defined in Table 4.1.

```
for each e ∈ E do
    for each game segment GS working on problem e do
        if V() = TRUE then
            ξ.add(f)
        end if
    end for
    for each f ∈ F do
        if ξ.count(f) ≥ τ then
            f is regarded as an answer for e
        end if
    end for
end for
```

Table 4.1: The Answer Extraction Procedure

The procedure counts all the unique answers generated from all game segments for a particular problem e. Answers with frequency lower than threshold τ will be pruned away.

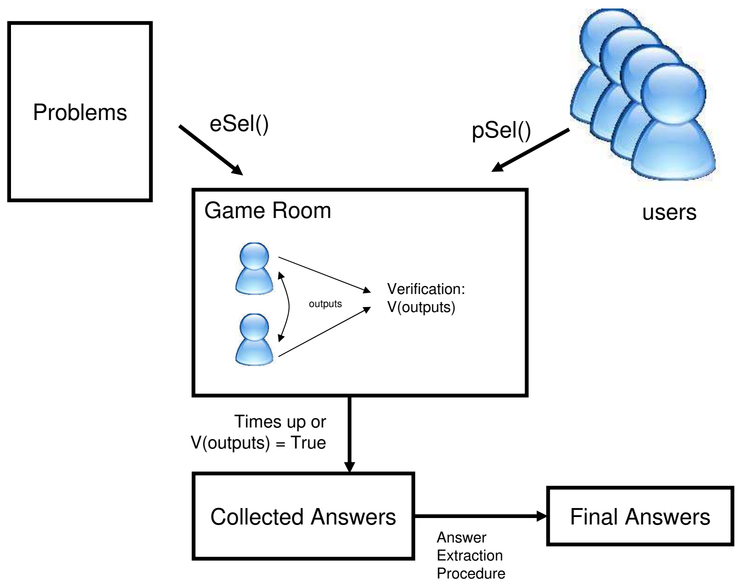

Figure 4.1: Simplified Social Game

Figure 4.1 illustrates the idea of social game. The system keep picking online users to join a game room and select problems for them to play. The game terminates when certain criteria are met. The system makes use of the collected answers from the played games and extract the final answers to problems.

4.3 Social Game Properties

4.3.1 Type of Information

To design a social game, we first declare whether subjective information or objective information aims to be collected:

Subjective Information: The choices of vocabularies for presenting information by different users are varied even on the same subject. The information presented for the same subject is affected by users.

- Assume it is a two-player game and the players aim to provide the common output. For a given problem e, there is a correct answer set $c\mathcal{F} \subset \mathcal{F}$. The correct output given by player p_1 is $\mathcal{O}(p_1) \cap c\mathcal{F}$ and the correct output given by player p_2 is $\mathcal{O}(p_2) \cap c\mathcal{F}$. When the information to be collected is subjective, it has lower probability on players' correct outputs are the same because $(\mathcal{O}(p_1) \cap \mathcal{O}(p_2) \cap c\mathcal{F}) \ll (\ (\mathcal{O}(p_1) \cap c\mathcal{F}) \cup (\mathcal{O}(p_2) \cap c\mathcal{F})\ )$.

Objective Information: The choices of vocabularies for presenting information by different users are the same for the same subject. The information presented for the same subject is not affected by users.

- Assume it is a two-player game and the players aim to provide the common output. For a given problem e, there is a correct answer set $c\mathcal{F} \subset \mathcal{F}$. The correct output given by player p_1 is $\mathcal{O}(p_1) \cap c\mathcal{F}$ and the correct output given by player p_2 is $\mathcal{O}(p_2) \cap c\mathcal{F}$. When the information to be collected is objective, it has higher probability on players' correct outputs are the same because $(\mathcal{O}(p_1) \cap \mathcal{O}(p_2) \cap c\mathcal{F}) \approx (\ (\mathcal{O}(p_1) \cap c\mathcal{F}) \cup (\mathcal{O}(p_2) \cap c\mathcal{F})\ )$.

Table 4.2 shows the categorization of social games. Table 4.3 presents examples of social games based on the categorization.

The current social games are categorized by game structure, verification method, game mechanism, and player requirement. In the following subsections, we describe the characteristic of each category based on our formal model.

Game Structure	Verification Method	Game Mechanism
Output-agreement	Symmetric	Collaborative or Hybrid
Input-agreement	Symmetric	Collaborative or Hybrid
Inversion-problem	Asymmetric	Collaborative or Competitive or Hybrid
Output-optimization	Symmetric or Asymmetric	Collaborative or Competitive or Hybrid

Table 4.2: Categorization of social games

4.3.2 Game Structure

Game structure defines the key elements of a game including the input of players, the output of players, the relationship among the input and output of all players, and the winning condition.

Output-agreement Game. All players are given the same input and must produce outputs based on the common input.

- $\mathcal{I}(p_1^1) = \mathcal{I}(p_2^1)$, the two players of the same role are given the common input in a game.

- In a two-player game, for a given problem e, there is a correct answer set $c\mathcal{F} \subset \mathcal{F}$, player p_1^1 has a set of potential outputs $\mathcal{O}(p_1^1) \subset \mathcal{F}$ and player p_2^1 has a set of potential outputs $\mathcal{O}(p_2^1) \subset \mathcal{F}$. The probability that players' outputs are accepted within a fixed period depends on $|\mathcal{O}(p_1^1) \cap \mathcal{O}(p_2^1)|$, where $\mathcal{O}(p_1^1) \cap \mathcal{O}(p_2^1)$ is the set of potential outputs shared by players. The larger the $\mathcal{O}(p_1^1) \cap \mathcal{O}(p_2^1)$, the higher the chance that an answer will be accepted with a fixed period.

- *An output-agreement game should be used to collect ob-*

Structure	Verification	Game Mechanism	Player Requirement		Examples
			Num of Player	Game Play	
OA	Symmetric	Collaborative	2	Synchronous	ESP, Matchin, Squigl
		Hybrid	Multi-players	Synchronous	Common Consensus
		Hybrid	Multi-players	Asynchronous	Gopher Game
IA	Symmetric	Collaborative	2	Synchronous	TagATune
		Hybrid	N/A	N/A	N/A
IP	Asymmetric	Collaborative	1 or 2	Synchronous	Peekaboom, Verbosity
		Competitive	2	Asynchronous	Dogear, CyPRESS
		Hybrid	1 or Multi-players	Synchronous	Phetch
OO	Symmetric	Collaborative	2	Synchronous	Restaurant Game
		Competitive	N/A	N/A	N/A
		Hybrid	Multi-players	Synchronous	Diplomacy
	Asymmetric	Collaborative	N/A	N/A	N/A
		Competitive	N/A	N/A	N/A
		Hybrid	N/A	N/A	N/A

[OA: Output-agreement; IA: Input-agreement; IP: Inversion-problem; OO: Output-optimization]

Table 4.3: Examples of social games

jective information rather than subjective information, because it has higher probability on players' correct outputs are the same for collecting objective information:$(\mathcal{O}(p_1^1) \cap \mathcal{O}(p_2^1) \cap c\mathcal{F}) \approx (\ (\mathcal{O}(p_1^1) \cap c\mathcal{F}) \cup (\mathcal{O}(p_2^1) \cap c\mathcal{F})\ )$.

- Since the output-agreement game assumes that there are no communications among players, the only information shared by players is the problem itself. Therefore, players who are telling the truth will have a larger $\mathcal{O}(p_1^1) \cap \mathcal{O}(p_2^1)$ and it has a higher chance to get their outputs accepted within a fixed period. In other words, it is very difficult for players to have their outputs accepted if they are not telling the truth.

Input-agreement Game. All players are given inputs that are known by the game (but not by the players) to be the same or different. The players are instructed to produce outputs describing their input, so their partners are able to assess whether their inputs are the same or different. Players see only each other's outputs.

- $\mathcal{I}(p_1^1)$ and $\mathcal{I}(p_2^1)$ are known by the game (but not by the player p_1^1 and p_2^1 of the same role) to be the same or different.

- In a two-player game, for a given problem e, there is a correct answer set $c\mathcal{F} \subset \mathcal{F}$, player p_1^1 has a set of potential outputs $\mathcal{O}(p_1^1) \subset \mathcal{F}$ and player p_2^1 has a set of potential outputs $\mathcal{O}(p_2^1) \subset \mathcal{F}$. The probability that players can correctly determine the input of players are the same or not within a fixed period depends on $|\mathcal{O}(p_1^1) \cap c\mathcal{F}|$ and $|\mathcal{O}(p_2^1) \cap c\mathcal{F}|$, where $\mathcal{O}(p_1^1) \cap c\mathcal{F}$ and $\mathcal{O}(p_2^1) \cap c\mathcal{F}$ are the set of correct outputs given by player p_1^1 and player p_2^1 respectively. The larger the sets $\mathcal{O}(p_1^1) \cap c\mathcal{F}$ and $\mathcal{O}(p_2^1) \cap c\mathcal{F}$, the more detailed information given by players, the higher the chance that players can correctly make determinations.

- *An input-agreement game should be used to collect subjective information rather than objective information*, because it has higher probability on players having detailed information when collecting subjective information compared with objective one.

- Since the two players do not communicate with each other, the only information the first player p_1^1 could has are the given input $\mathcal{I}(p_1^1)$ and the hints $\mathcal{O}(p_2^1)$ given by the second player about $\mathcal{I}(p_2^1)$. On the other hand, the only information the second player p_2^1 could has are the given input $\mathcal{I}(p_2^1)$ and the hints $\mathcal{O}(p_1^1)$ given by the first player about $\mathcal{I}(p_1^1)$. Therefore, players who are telling the truth will have a larger $\mathcal{O}(p_1^1) \cap c\mathcal{F}$ and a larger $\mathcal{O}(p_2^1) \cap c\mathcal{F}$ and it has a higher chance that players can correctly determine their inputs are the same or not within a fixed period. In other words, it is very difficult for players to make accurate determination if not telling the truth.

Inversion-problem Game. The first player has access to the whole problem and gives hints to the second player to make a guess. If the second player is able to guess the secret, we assume that the hints given by the first player are correct.

- In a two-player game, $\mathcal{I}(p_1^1)$ is given by the game and $\mathcal{I}(p_1^2)$ is set as the output provided by player p_1^1 (i.e., $\mathcal{I}(p_1^2) = \mathcal{O}(p_1^1)$).

- In the inversion-problem game, every hint given by the first player p_1^1 corresponds to a set of possible guesses which are the outputs of the second player $\mathcal{O}(p_1^2)$. Since the players do not communicate, the only information the second player could have are the hints $\mathcal{O}(p_1^1)$ given by the first player. The probability that the second player successfully guesses the secret, the input data $\mathcal{D}$, within a fixed period depends on the size of $\mathcal{O}(p_1^2)$ that is $|\mathcal{O}(p_1^2)|$. The smaller $\mathcal{O}(p_1^2)$, the higher chance the second player can make a correct guess within a fixed period.

- In reality, a small $\mathcal{O}(p_1^2)$ represents that the hints $\mathcal{O}(p_1^1)$ given by the first player is more relevant to the secret, the input data $\mathcal{D}$, that the second player has to guess, which means that it is a better answer to the problem we want to solve.

Output-optimization Game. All players are given the same input and their outputs are the hints of other players' outputs.

- In a two-player game, $\mathcal{I}(p_1^1)$ and $\mathcal{I}(p_1^2)$ are given by the game.
- In the output-optimization game, since players can communicate with each other using their outputs, the output $\mathcal{O}(p_1^1)$ given by the first player affects the output $\mathcal{O}(p_1^2)$ given by the second player, while the output $\mathcal{O}(p_1^2)$ given by the second player affects the output $\mathcal{O}(p_1^1)$ given by the first player. The collected information are the output patterns.
- *An output-optimization game should be used to collect subjective information rather than objective information*, because the output pattern of players reflects outputs of players are strongly affected by others' outputs. It is subjective.

4.3.3 Verification Method

Verification method of a game defines the method to check the output accuracy of players by asking players to do the same tasks or different tasks.

Symmetric Verification Game. The verification of a game is symmetric in a sense that all players are asked to perform the same task and their outputs are checked against each other. *Either an output-agreement game or an input-agreement game is symmetric verification.*

- $\mathcal{R} = \{r_k \mid k = 1\}$, all players in a game could be assigned to the only one role.

Asymmetric Verification Game. The verification of a game is asymmetric in a sense that all players are asked to do different tasks and their outputs are checked against each other. *An inversion-problem game is asymmetric verification.*

- $\mathcal{R} = \{r_k \mid k \geq 2\}$, players in a game could be assigned to one of the roles.

4.3.4 Game Mechanism

Game mechanism defines the relationship of all players in the game in order to achieve the winning condition.

Collaborative Game. To achieve the *winning condition of all players*, a player has to complete his assigned task which is helping other players to complete their tasks. *A game of any game structure can be a collaborative game.*

- For a two-player collaborative game, when both players (i.e., p_1^1 and p_2^1) complete their assigned tasks which is helping each other to complete his tasks, both players (i.e., p_1^1 and p_2^1) achieve the winning condition and receive rewards (i.e., $\mathcal{W}(p_1^1)$ and $\mathcal{W}(p_2^1)$).
- *The accuracy of output is guaranteed by collaboration of all players.*

Competitive Game. To achieve the *winning condition of a player*, a player has to complete his assigned task. His achievement is compared with other players' achievement or his history of game records or information stored in a database. *Neither an output-agreement game nor an input-agreement game can be a competitive game.*

- For a two-player competitive game which determines the precise accuracy of the player's guess based on the information stored in the database, when player p_1^1 can make a guess correctly, player p_1^1 achieve the winning condition and receive a reward, i.e., $\mathcal{W}(p_1^1)$.
- *The accuracy of output is guaranteed by information stored in a database. Players' enjoyment in the game can be increase in competition.*

Hybrid Game. To achieve the *winning condition of some players*, players have to complete their assigned tasks which are helping other players to complete their tasks. After that, the achievements of all players are compared with other players' achievements or their history of game records or information stored in a database. *A game of any game structure can be a hybrid game.*

- For a hybrid game, the player of a role (i.e., p_1^1) tries to help all players of the other role to complete their tasks (i.e., p_1^2, p_2^2, p_3^2, ..., p_M^2). When one of the players of the other role (i.e., p_M^2) complete his assigned tasks, both players (i.e., p_1^1 and p_M^2) achieve the winning condition and receive a reward (i.e., $\mathcal{W}(p_1^1)$ and $\mathcal{W}(p_M^2)$).
- *The accuracy of output is guaranteed by collaboration of the winning two players. Players' enjoyment in the game can be increase in competition.*

4.3.5 Player Requirement

Player requirement defines the rules on accessing the game of all players. They are (1) players accessing the game at different time period are allowed or not, i.e., synchronous or asynchronous; (2) the number of players is required in a game.

Synchronous Game. A game is synchronous in a sense that players of the game who happen to be accessing the game at the same time. Players have to give real-time response to other players' action. *A game of any game structure and any game mechanism can be a synchronous game.*

- All players in a game (i.e., $\forall p \in \mathcal{P}$) are accessing the game during the maximum duration of a game, $tMax$.

Asynchronous Game. A game is asynchronous in a sense that it is not necessary for players of the game to access the game at the same time. Players do not have to give real-time response to other players' action. There is a time delay in between. The information collected from one player is stored in a database and will be used to determine the correctness of other players' output. *A game of any game structure and any game mechanism can be a synchronous game.*

- Not all players in a game (i.e., $\exists p \in \mathcal{P}$) is accessing the game during the maximum duration of a game, $tMax$.

Single-player Game. One player in a game is allowed and the moves of one role can be simulated from the prerecorded game. *Only input-agreement game and inversion-problem game can be a single-player game.*

- $\sum_{k=1}^{X} n\mathcal{P}(r_k) = 1$ where $n\mathcal{P}(r_k)$ is the number of players that are assigned to the role r_k during a game and $X = n\mathcal{R}$ is the number of roles in a game.

Two-player Game. A game allows two players to play together. *A game of any game structure can be a two-player game.*

- $\sum_{k=1}^{X} n\mathcal{P}(r_k) = 2$ where $n\mathcal{P}(r_k)$ is the number of players that are assigned to the role r_k during a game and $X = n\mathcal{R}$ is the number of roles in a game.

Multi-player Game. A game allows multiple players to play together. *Only hybrid game can be a multi-player game.*

- $\sum_{k=1}^{X} n\mathcal{P}(r_k) > 2$ where $n\mathcal{P}(r_k)$ is the number of players that are assigned to the role r_k during a game and $X = n\mathcal{R}$ is the number of roles in a game.

4.4 Design Guideline

Current social games are causal games which are easy for game designers to design and they are designed in ad-hoc based. There does not exist any rules on how to design a social game for solving a specific problem. A set of design guidelines is necessary to help game designers to design a social game for solving a problem in general. They help the designers to layout the properties of the game based on the characteristics of problems. These guidelines based on the properties of our proposed model are shown in Table 4.4.

In the following, we use some current social games as examples to illustrate how to design a social game for a given problem using the design guidelines.

Given our task is to locate objects in the labels of images. The input object of the game is an *image*, and the attribute *label* is our concern. The data of attribute *label* is of data type *text*. Since labels are objective and obvious information, we may design an output-agreement game or an inversion-problem game.

if *data.attr.value* = *objective* **then**
  struct = (**output-agreement or inversion-problem**)
else if (*data.attr.value* = *subjective* **and**
data.attr.value.data-type = *output-pattern*) **then**
  struct = **output-optimization**
else if (*data.attr.value* = *subjective* **and**
data.attr.value.data-type $\neq$ *output-pattern*) **then**
  struct = (**input-agreement or inversion-problem**)
end if
if *struct* = (**output-agreement or input-agreement**) **then**
  if $no\text{-}of\text{-}players > 2$ **then**
    (*mechanism* = *hybrid* **and** *time* = *sync*)
  else if $no\text{-}of\text{-}players = 2$ **then**
    (*mechanism* = *collaborative* **and** *time* = *sync*)
  end if
end if
if *struct* = **inversion-problem then**
  if $no\text{-}of\text{-}players > 2$ **then**
    (*mechanism* = *hybrid* **and** *time* = *sync*)
  else if $no\text{-}of\text{-}players = 2$ **then**
    if *verification of answer based on players' output* **then**
      (*mechanism* = *collaborative* **and** *time* = *sync*)
    else if *verification of answer based on info in DB* **then**
      (*mechanism* = *competitive* **and** *time* = *async*)
    end if
  else if $no\text{-}of\text{-}players = 1$ **then**
    [*mechanism* = (*collaborative* **or** *hybrid*) **and**
moves simulated from the prerecorded game]
  end if
end if
if *struct* = **output-optimization then**
  if $no\text{-}of\text{-}players > 2$ **then**
    (*mechanism* = *hybrid* **and** *time* = *sync*)
  else if $no\text{-}of\text{-}players = 2$ **then**
    mechanism = (*collaborative* **or** *competitive*)
  end if
end if

Table 4.4: The Design Guidelines on Social Games

To design an output-agreement game for locating objects in the labels of images, we have only 2 players in the game and it is the Squigl game. However, if we choose to have more than 2 players in the game, then it is a hybrid game. For example, there are 3 players in the game. Each player is given the same input image and is asked to locate objects in labels of the image. The location of object in an image for a label provided by a player is assumed to be correct when two players drag the same area for the object. The first two players complete the dragging of the area of an object related to a label and the overlapping area is higher than a threshold will gain marks. It encourages the players to drag the object as fast as possible and locate the object correctly in order to have higher probability to match other players' output. At the same time, all the players in the game are competing against each other.

To design an inversion-problem game for solving labeling images problem, we have only 2 players in the game, it is the Peekaboom game. However, if we choose to have more than 2 players in the game, then it is a hybrid game and it may be similar to Phetch. There are 3 players in the game. One of the players is given an input image and the player provides labels to all other players, while other players have to guess which one is the input image from a set of images. The player guessing the correct image in the shortest time and the describer of the image will gain marks.

4.5 Opinion Mining Game Design

4.5.1 OpinionMatch

The simplest way that could surely help machines to perform opinion mining is to outsource the entire sentiment classification problem to human. For instance, we design a social game called **OpinionMatch** and the game directly give players of the social game the review passage and ask them to judge the sentiment expressed. Both players are given the same passage and asked to answer the same question during the game. A match between their answers indicates the agreement on the opinion expressed by the passage and thus regarded as correct. This way, OpinionMatch is an output-agreement game that depends on symmetric verification strategy.

Depending on the details of the opinion mining results that we want, we can ask the player to tell the (1) commented target object or feature and (2) expressed sentiment polarity (+ve / -ve). This way, machines are only responsible for crawling review data and collecting answers generated from the social game.

OpinionMatch in our model

OpinionMatch, in terms of our model, can be defined as:

Data $\mathcal{D}$ is a review passage which is of type *text* crawled from the web. $\mathcal{D}$ contains various attributes such as the actual *review text*, *commented target object (or feature)* and the corresponding *sentiment polarity*.

The actual OpinionMatch **social game** is a 4-tuple ($\mathcal{SGPD}$, $\mathcal{GR}$, $\mathcal{GF}$, $\mathcal{ANS}$), where sets:

1. $\mathcal{SGPD} = (\mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{C})$ is the social game problem domain.
   (a) $\mathcal{E} = \{e_i | i = 1, ..., x\}$ is a set of problems that we want to solve where the problem e_i is to collect the commented target object (or feature) or the sentiment polarity of the input $\mathcal{D}$ (i.e., review passage).
   (b) $\mathcal{F} = \{f_i | i = 1, ..., y\}$ is the answer domain. That is the correct object, feature or sentiment polarity expressed in the input passage.
   (c) $\mathcal{G} : \mathcal{E} \times \mathcal{F} \rightarrow \Re \in [0..1]$ is a function that determine whether an answer is correct to a problem.

(d) $\mathcal{C}$ is a set of constraints in the game that

i. indicating the attribute(s) we want to collect (i.e., commented target object (or feature) or the expressed sentiment polarity)

ii. indicating the set of values that we want to collect within $\mathcal{V}(\mathcal{A}_{\mathcal{X}})$.

2. $\mathcal{GR} = (\mathcal{D}, \mathcal{M}, \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{I}, \mathcal{O}, \mathcal{V}, \mathcal{W})$ represents rules of a social game.

(a) $\mathcal{D}$ is input data that we want to collect its metadata.

(b) $\mathcal{M}$ = {commented target object (or feature) or expressed sentiment polarity} is a set of metadata which are the values of attributes of $\mathcal{D}$ that we want to collect.

(c) $\mathcal{C}$ is a set of constraints in the game that

i. indicating the attribute(s) we want to collect (i.e., commented target object (or feature) or the expressed sentiment polarity)

ii. indicating the set of values that we want to collect within $\mathcal{V}(\mathcal{A}_{\mathcal{X}})$.

(d) $\mathcal{R} = \{r_0\}$ is the set of roles that players could have during a game. In OpinionMatch, all players share the same role.

(e) $\mathcal{P}(r_k) = \{p_j^k | j = 1, ..., n\mathcal{P}(r_k)\}$ is the set of players that are assigned to the role r_k during a game.

(f) $\mathcal{I}(p_j^k) = \{i_m^{k,j} | m = 1, ..., n\mathcal{I}(p_j^k)\}$ is the set of input given to the player p_j^k for solving the problem of input $\mathcal{D}$ during a game.

(g) $\mathcal{O}(p_j^k) = \{o_m^{k,j} | m = 1, ..., n\mathcal{O}(p_j^k)\}$ is the set of output provided by the player p_j^k for solving the problem of input $\mathcal{D}$ during a game.

(h) $\mathcal{V}()$ is a procedure that determines whether players have produced outputs that meet specific requirements within a game segment (i.e., $\bigcap \mathcal{O}(p_j^k) \neq \phi$). If so, return a possible answer $f \in \mathcal{F}$.

(i) $\mathcal{W}(p_j^k)$ is the reward that the player can receive for solving the problem of input $\mathcal{D}$ during a game where $\mathcal{W}(p_j^k) \in \{w_i | i = 1, ..., y\}$. Players will receive a reward when achieving the winning condition of the game.

Analysis

The obvious advantage of OpinionMatch is that all the difficulties such as natural language processing and sentiment analysis are not longer required for the machine. With proper settings and anti-cheating methodologies [13, 49] applied, we can expect it to perform opinion mining with high accuracy. However, such a "game" barely contains any entertainment values. The game plays are ad-hoc in a sense that players are asked to answer the given questions one by one with no creativeness and competitive feeling at all. Such a game could hardly attract people to play and thus has no practical values.

4.5.2 FeatureGuess

As learn from the example above, social game could not be designed arbitrarily to solve any computation problems. The game, on top of the problem solving capabilities, must also contain entertainment values in order to work. In here, we propose another social game called **FeatureGuess** that could possibly help the opinion mining process. FeatureGuess is a two-player inversion problem game. In each round of the game, player A will be given a feature of a particular object domain (e.g., *picture quality* under the *camera* domain). Player B, who does not know the feature, has to guess it with the hints given by player A. Player A could say anything to player B about the feature without mentioning the feature itself. If player B can guess the feature eventually, everything mentioned by player A are possibly related to the feature.

FeatureGuess in our model

FeatureGuess, in terms of our model, can be defined as:

Data $\mathcal{D}$ is a feature that can be commented under a particular domain. It is of type *text*. $\mathcal{D}$ contains various attributes such as the *adjectives used* (e.g., *excellent* for *picture quality*), *possible values* (e.g., *400* for *ISO values*).

The actual FeatureGuess **social game** is a 4-tuple ($\mathcal{SGPD}$, $\mathcal{GR}$, $\mathcal{GF}$, $\mathcal{ANS}$), where sets:

1. $\mathcal{SGPD} = (\mathcal{E}, \mathcal{F}, \mathcal{G}, \mathcal{C})$ is the social game problem domain.
   (a) $\mathcal{E} = \{e_i | i = 1, ..., x\}$ is a set of problems that we want to solve where the problem e_i is to collect the feature-related attributes.
   (b) $\mathcal{F} = \{f_i | i = 1, ..., y\}$ is the answer domain. That is the correct attributes of the feature.
   (c) $\mathcal{G} : \mathcal{E} \times \mathcal{F} \to \Re \in [0..1]$ is a function that determine whether an answer is correct to a problem.
   (d) $\mathcal{C}$ is a set of constraints in the game that
      i. indicating the attribute(s) we want to collect.
      ii. indicating the set of values that we want to collect within $\mathcal{V}(\mathcal{A}_{\mathcal{X}})$.
2. $\mathcal{GR} = (\mathcal{D}, \mathcal{M}, \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{I}, \mathcal{O}, \mathcal{V}, \mathcal{W})$ represents rules of a social game.
   (a) $\mathcal{D}$ is input data that we want to collect its metadata.
   (b) $\mathcal{M} = \{m_i | i = 1, ..., x\}$ is a set of metadata which are the values of attributes of $\mathcal{D}$ that we want to collect.

(c) $\mathcal{C}$ is a set of constraints in the game that

  i. indicating the attribute(s) we want to collect.

  ii. indicating the set of values that we want to collect within $\mathcal{V}(\mathcal{A}_{\mathcal{X}})$.

(d) $\mathcal{R} = \{r_0, r_1\}$ is the set of roles that players could have during a game. In FeatureGuess, there are two types of player, hints giver and guesser.

(e) $\mathcal{P}(r_k) = \{p_j^k | j = 1, ..., n\mathcal{P}(r_k)\}$ is the set of players that are assigned to the role r_k during a game.

(f) $\mathcal{I}(p_j^k) = \{i_m^{k,j} | m = 1, ..., n\mathcal{I}(p_j^k)\}$ is the set of input given to the player p_j^k for solving the problem of input $\mathcal{D}$ during a game.

(g) $\mathcal{O}(p_j^k) = \{o_m^{k,j} | m = 1, ..., n\mathcal{O}(p_j^k)\}$ is the set of output provided by the player p_j^k for solving the problem of input $\mathcal{D}$ during a game.

(h) $\mathcal{V}()$ is a procedure that determines whether players have produced outputs that meet specific requirements within a game segment (i.e., $\mathcal{I}(p_j^0) \in \mathcal{O}(p_j^1)$). If so, return a possible answer $f \in \mathcal{F}$.

(i) $\mathcal{W}(p_j^k)$ is the reward that the player can receive for solving the problem of input $\mathcal{D}$ during a game where $\mathcal{W}(p_j^k) \in \{w_i | i = 1, ..., y\}$. Players will receive a reward when achieving the winning condition of the game.

Analysis

Unlike OpinionMatch which tries to solve the opinion mining problem directly, FeatureGuess focus on a sub-problem in opinion mining, namely, Feature-Opinion Association (FOA). FeatureGuess helps FOA in a sense that it requires player to list out everything about a feature. This can help FOA algorithms to associate feature and opinion-related items more accurately.

While FeatureGuess itself is entertaining because of its cooperative nature, it could further be used in various places such as in schools and competitions to enhance the practical and entertainment values. In schools, FeatureGuess can act as a tool that help students to learn new concepts and could explore their creativity. For the general public, FeatureGuess can be used in a competition to assess a person's expertise towards a particular field (i.e., camera expert competition). These make FeatureGuess a very useful game that has both entertainment and practical values.

4.6 Summary

In this chapter, we study the properties of existing social games. We develop a general framework that allows us to better understand these games. We categorize existing social games based on their nature and derive a design guideline for developing new games that would solve other problems. Finally we propose two games for opinion mining related purposes and discuss the advantages and disadvantages of them.

Chapter 5

Tag Sentiment Analysis for Social Bookmark Recommendation System

5.1 Motivation

Social bookmarking is a popular Web 2.0 concept in which people store their bookmarks online. Famous social bookmarking site such as Del.icio.us[1] has more than 5 million users and 150 million bookmarked URLs [89]. Users of the site could share their favorite bookmarks with their friends and discover the hottest topics currently on the web by looking at the most popular websites bookmarked by other people [88]. In addition to the traditional URL entries, social bookmarking systems introduce a phenomenon called *tagging* [27, 23], in which users can tag their bookmarks. Tagging provide clear and concise descriptions of websites and is very popular among internet users.

The success of social bookmarking brings us a new interesting topic of research called *bookmark recommendation*. The idea of it is to study the behavior of individual users and suggest bookmarks that they may find interesting. Existing studies on this topic are either social network or semantic-based. Social

[1] http://del.icio.us

network-based models [18] assume that friends or people within the same local community share common interest and trust each others more than the others. Thus, URLs bookmarked by friends are good sources for recommendations. Semantic-based models [74] try to understand the "meaning" of the bookmarked URLs by looking into its content, bookmark structures or tags. Based on this information, similarities among websites are computed. Users are suggested with the websites similar to those they have bookmarked.

Tag	**Bookmark Counts**
funny	453,118
cool	356,372
useful	112,471
awesome	80,012
recommended	16,320

Table 5.1: Del.icio.us Subjectively Tagged Bookmarks (As of 3-Nov-2008)

Although existing approaches are able to generate some good results, they all suffer from some major drawbacks. Social bookmarking, literally, includes two major concepts, 1) social collaboration and 2) bookmarking people's favorite websites. However, current methods in bookmark recommendation do not take both concepts into account. The main reason is that there is not a framework that allows us to incorporate both concepts systematically, and thus limiting the effectiveness of the solutions.

Intuitively, tags should be words that provide short and objective description of websites. For example, some "appropriate" tags for a car manufacturer's website are "car", "automobile" and the name of the company. These are all nouns. However, we have conducted a brief study on Del.icio.us to find out whether there are subjective terms such as "recommended" in the bookmarks' tags. The result is surprising as shown in Table 5.1. There are over 16,000 bookmarks carrying the tag "recommended" which is not a small number considering that the original idea of a tag is just a short objective description. One notable observation is that many of such bookmarks are tagged "subjectively" not because of the title or the content of the site has the word, but rather the users actually recommended the page. This actually raises an interesting question, "Are all bookmarks created equal by a user?" In order words, we want to know whether a user likes some of his/her bookmarks more than the others.

Based on the above observations, we studied the problem of social bookmarking recommendation with tag sentiment analysis. We propose a framework based on collaborative filtering to solve the problem. The main reason is that the nature of our problem is a recommendation problem. Collaborative filtering has been well-studied to solve this type of problem and has solid mathematical foundations [11, 30]. The proposed users' similarities measures in the framework allow us to handle the aforementioned problem and take the tags' subjectivity into considerations for better bookmark recommendations.

5.2 Problem Statement

In this section, we define the basic concepts of social bookmarking systems. Base on these definitions, we futher define the social bookmark recommendation problem.

5.2.1 Social Bookmarking Model

A social bookmarking system is made up of a set S of websites (i.e., URLs), a set T of tags and a set U of users who participated in the system by creating bookmarks B. Bookmark is a ternary relation where user $u \in U$, who is the creator of the bookmark $b \in B$, records website $s \in S$ with a set of tag $t \in T$ describing s in his/her website collection C_u. A user is allowed to create as many bookmark as they like and use any tags that they find suitable to a website. Users can add other users into their social networks N. Fan relation F is a binary relationship between two users $X, Y \in U$, such that if X is a fan of Y then we say that X regards Y as a friend in his/her social network (i.e., $Y \in N_X$).

5.2.2 Social Bookmark Recommendation (SBR) Problem

Social bookmark recommendation is to suggest interesting bookmarks to users given their personal preferences. Giving a system defined as above, the SBR problem is to find out, for each user $u \in U$, a list of top-K set of websites $s \in (S - C_u)$ which is most interesting to u given C_u and N_u. The notion of "interesting website" is the main concern of the SBR problem and is going to be discussed in the following sections.

5.3 Proposed Approach

Existing works on bookmark recommendation do not have a unified framework to combine social network and semantic information into the solution, which hinder the performance of such systems. In this section, we first propose a Social Bookmark Recommendation Framework which gives an overview on how a bookmark recommendation system should looks like. We then discuss each individual component inside this framework and give instances of them based on our observations.

5.3.1 Social Bookmark Recommendation Framework

In bookmark recommendation systems, we assume a user creates bookmarks because he/she is interested in the sites. We also assume that friends of a user at least share some interests. In order to generate recommendations, we must be able to answer these two questions 1) what types of websites (semantically) are being bookmarked by a user? and 2) who (either an explicit friend or someone else in the system) share the same interest with the user?

A bookmark recommendation system must take these two problems into consideration. Here we propose a **Social Bookmark Recommendation Framework (SRBF)** as shown in Figure 5.1. The framework gathers all types of available infor-

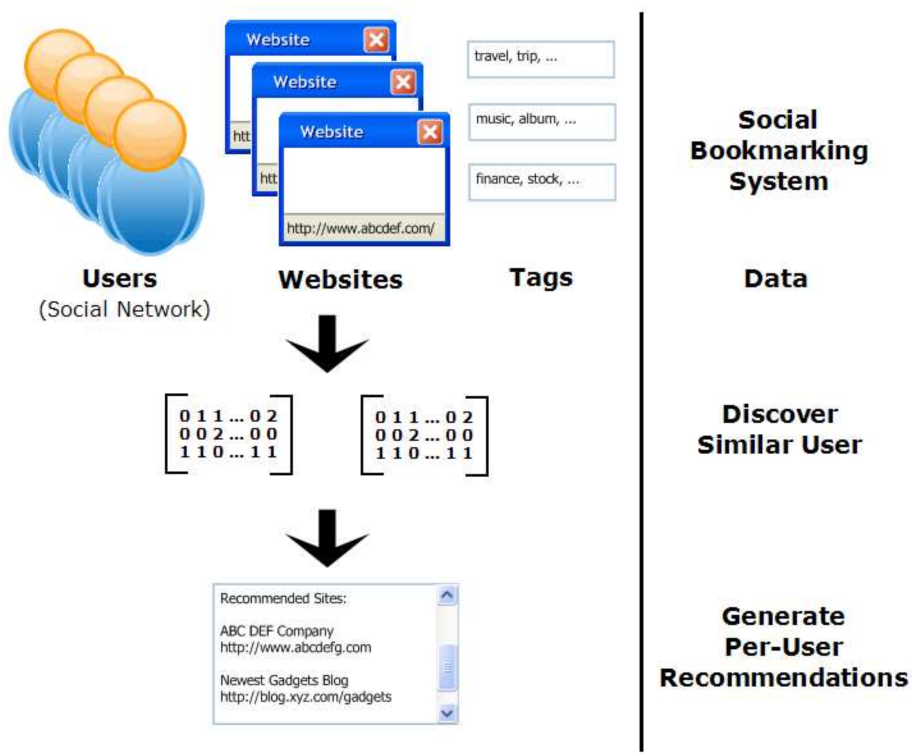

Figure 5.1: Bookmark Recommendation Framework

mation, such as users, friendship relations and bookmarks as inputs. They cover both social network (friendship relations) and semantic (bookmarked websites and tags) aspect of social bookmarking system. Using the above information, the framework discovers similar users in the system and generates per-user recommendations.

In this paper, we propose to use a collaborative filtering approach to handle the similar user discovery and recommendation generation process in SBRF. We introduce the concept of sentimental tag that can be used as a preparation step for collaborative filtering for bookmark recommendation system.

5.3.2 Subjective Tag Detection (STD)

The problem of sentimental tag detection is to determine whether a tag is used by a user to express subjective feeling. When we first look at the problem, it seems that the solution is just a simple lexicon lookup. Since sentiment lexicons are already available, what we need to do is just to check whether a tag is inside such a lexicon. If yes, we say that the tag is subject. However, the reality is always not as simple as it may seem.. Let's take a look at the following bookmark (taken from del.icio.us):

```
Title: ‘‘Recommended Add-ons :: Firefox Add-ons’’
Tags: firefox, addons, add-ons, plugins, recommended
```

Although one of the tag is "recommended", which is term used to express positive opinion, it does not mean that those who use this tag really think that the site is so good. The reason why this tag is chosen is mostly because of the fact that the title also contains this term.

Base on this observation, we propose the **Subjective Tag Detection (STD)** algorithm as shown in Algorithm 3.

The function *correlated* checks the pairwise co-occurrence ratio (PCOR) among tags from the same bookmark. The ratio is defined as follows:

$$PCOR(t_1, t_2) = \frac{|\{b|b \in (E(t_1) \cap E(t_2))\}|}{|E(t_1)|}, \qquad (5.1)$$

Algorithm 3 Subjective Tag Detection

```
INPUT:
title: title of the bookmarked website
tag: tag we want to check
otherTags: other tags of the same bookmark
sentimentLexicon: a list of sentimental words
OUTPUT:
isSubjective: whether the tag is subjective
BEGIN
if title contains the tag then
    return false
else
    if correlated(tag, otherTags) then
        return false
    else
        if tag ∉ sentimentLexicon then
            return false
        end if
    end if
end if
return true
END
```

where $E(t)$ is a set of bookmark containing tag t. $PCOR$ estimates the probability that the two tags are used for the same bookmark. If $PCOR$ is high, there are likely to be words that belong to the same topic. If a particular tag in a bookmark has low $PCOR$ with the rest of the tags and it appears in the sentiment lexicon, then we assume that the tag is sentiment expressing. Let's look at the following tags from the same bookmark:

```
Title: ``ABC Webhosting''
Tags: ABC, webhosting, hosting, recommended
```

If the tag "recommended" in the above bookmark fulfills the following:

$$PCOR(webhosting, recommended) < \varepsilon, and$$

$$PCOR(hosting, recommended) < \varepsilon,$$

where ε is a threshold for the minimum co-occurrence between two words, and the word has an entry in the sentiment lexicon, the subjective tag detection algorithm will say that it is a sentimental tag.

The intuition behind the PCOR measure is that if two tags (e.g. iPod and mp3) can be used to describe websites of similar topics (music player), they should appear together frequently. Opinion expressing tags for bookmarks should to be topic-neutral and therefore these tags should have low PCOR value with other tags. If a tag does not appear in the website's title and have low PCOR values with other tags, there are two possibilities: 1) it is not a common tag or 2) the tag is opinion expressing. The first case could be caused by the used of terms that is too specific or a typing error. The latter case should be detected by our subjective tag detection algorithm with the help of a proper sentiment lexicon.

5.3.3 Similarity Matrices

In traditional collaborative filtering systems [66, 93], a user-item matrix is used for similar neighbor selection as well as rating prediction. Given a system consists of M users and N items, an $M \times N$ user-item matrix could be constructed. Each entry $r_{m,n}$ in the matrix denotes the rating on item n given by user m. The ratings are usually normalized into a number scale with the two extremes representing positive and negative opinions. A positive rating indicates that the user is satisfied with the product and a negative rating carries the opposite meaning. This user-item matrix shows the "tastes" of individual users and also how different people rate on each item, thus, it is the main source for computing both user and item similarities.

While the rating is explicit in product recommendation systems, there are no corresponding concepts in social bookmarking websites. In a social bookmarking system, there is only one action available for a user, bookmark a website. By performing such actions, users implicitly expressed their interests on the sites they have bookmarked. There isn't a rating scale such as 5 stars for site A and 2 stars for site B. There also is not a concept of negative bookmarks. People only bookmark the sites they like, but will not do anything for those they do not like. If a user-item matrix is to be built using only the bookmarks in which item represents website, the rating could only be 1 (the site is bookmarked user) or 0 (the site is not bookmarked by the user). This information may not be enough for effective computation of similarities.

In order to solve this problem, we propose to use three user-item matrices for similarity computation. They are, namely, **User-Website matrix**, **User-Tag matrix** and **Website-Tag matrix**. The following sub-sections explain the details of the 3 matrices.

5.3.4 User-Website matrix:

The User-Website matrix records the website bookmarked by each user in the system. However, given the above mentioned problem, the values inside the matrix will only be 1 and 0. We propose to add a rating value 2 to show that some websites are more interesting with respect to each user. Table 5.2 shows the meaning of different rating value. Subjective tags are discovered

Rating	Meaning
0	User did not bookmarked the site
1	User bookmarked the site
2	User bookmarked the site with subjective tags

Table 5.2: Rating Scale in User-Website Matrix

by our proposed algorithm discussed in section 5.3.2. The reason for proposing this rating scale is that we believe that some users prefer some websites more than others websites even though they all contained in their bookmark collections. Subjective tags are clearer indication on this. Even for tags that are said to be carrying negative sentiment in traditional opinion mining applications, the use of it does not necessarily mean that the user hate the site. Instead, we believe that it is the site that makes the user so excited and use this type of tag.

5.3.5 User-Tag matrix

We know that in addition to bookmarked websites, tags also contain important clues for discovering users' interests. If two users share a lot of tags frequently, we know that they share similar interests even though they might not have bookmarked many websites in common. The idea of User-Tag matrix is to record the frequencies of tags used by each user so that tag usage statistics could be used for similarities computation.

5.3.6 Website-Tag matrix

In addition to user similarities, tags also provide important information for website similarities. The more tags two websites share, the more similar they are. Therefore, Website-Tag matrix records the frequencies of tags assigned to each website.

5.4 Pearson Correlation Coefficient

Pearson Correlation Coefficient (PCC) is a popular similarity definition in collaborative filtering systems [69]. With PCC, the similarity $sim(a, b)$ between user a and b can be computed as follows:

$$sim(a,b) = \frac{\sum\limits_{i \in I(a) \cap I(b)} (r_{a,i} - \overline{r}_a) \cdot (r_{b,i} - \overline{r}_b)}{\sqrt{\sum\limits_{i \in I(a) \cap I(b)} (r_{a,i} - \overline{r}_a)^2} \cdot \sqrt{\sum\limits_{i \in I(a) \cap I(b)} (r_{b,i} - \overline{r}_b)^2}}, \tag{5.2}$$

where $I(a)$ and $I(b)$ denotes the items rated by user a and b respectively. $r_{a,i}$ is the rating on item i given by user a, and $\overline{r}_a$ represents the average rating of user a. This definition uses the items rated by both user a and b to come up with a similarity value between $[0, 1]$. A higher $sim(a, b)$ value means that user a and b are more similar to each other.

Similarly, we can compute the item similarity $sim(i, j)$ between any item i and j as follows:

$$sim(i,j) = \frac{\sum\limits_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r}_i) \cdot (r_{u,j} - \overline{r}_j)}{\sqrt{\sum\limits_{u \in U(i) \cap U(j)} (r_{u,i} - \overline{r}_i)^2} \cdot \sqrt{\sum\limits_{u \in U(i) \cap U(j)} (r_{u,j} - \overline{r}_j)^2}}, \tag{5.3}$$

where $U(i)$ and $U(j)$ denotes the set of users who have rated item i and j respectively. $\overline{r}_i$ represents the average rating received by item i.

However, if two users have rated only a very small number of items in common and those ratings are very similar to each other, PCC will have the problem of overestimating their similarity. Ma [51] proposed a modified version of Herlocker's equation [29] to incorporate significance weighting in the similarity computation. Hence, the new user similarity $Sim'(a,b)$ is defined as:

$$Sim'(a,b) = \frac{\min(|I_a \cap I_b|, \gamma)}{\gamma} \cdot sim(a,b). \tag{5.4}$$

and similarly for item similarity:

$$Sim'(i,j) = \frac{\min(|U_i \cap U_j|, \delta)}{\delta} \cdot sim(i,j). \tag{5.5}$$

The parameters γ and δ in the equations denote the minimum number of commonly rated items (or users) for the significance weighting to be 1. If two users rated too few common items (or two items receive too few common ratings), the similarity will drop to reflect the lack of significance.

5.5 Social Network-based User Similarity

Without consider the underlying social network, the final user similarity $usim(a,b)$ and item similarity $isim(i,j)$ taking both matrices into account in a social bookmarking system can be computed as follows:

$$usim(a,b) = \alpha \cdot sim'_{uw}(a,b) + (1-\alpha) \cdot sim'_{ut}(a,b), \text{ and} \tag{5.6}$$

$$isim(i,j) = \alpha \cdot sim'_{wu}(i,j) + (1-\alpha) \cdot sim'_{wt}(i,j), \tag{5.7}$$

where sim'_{uw} and sim'_{ut} represents user similarities computed by Eq. (5.4) using User-Website and User-Tag matrix respectively. sim'_{wu} is computed by Eq. (5.5) with User-Website matrix. sim'_{wt} is computed by Eq. (5.5) using Website-Tag matrix. α is a value between $[0, 1]$ which controls the weighting of each matrix in the final similarity value.

However, as mentioned before, social network provides important clues for discovering users' interests. That means we can do more on user similarity if we have the social network information. If user a is a fan of user b, then user a may be interested in some of the user b's favorite sites. This relation could be transitive but the effects should diminish for users who are far away in the social network. Taking this into account, our proposed user similarity function is defined as follows:

$$usim'(a, b) = \beta \cdot usim(a, b) + (1 - \beta) \cdot FanFactor(a, b), \quad (5.8)$$

where $usim(a, b)$ is the similarity function without social network information as described in Eq. (5.6). $FanFactor$ is a function that computes how "friendly" user b is with respect to user "a". The definition is provided in Eq. (5.9). β is a parameter between $[0, 1]$ that controls the importance $FanFactor$ in user similarity computation.

$$FanFactor(a, b) = \begin{cases} \frac{1}{dist(a,b)}, & \text{if a path exists between } a, b \\ 0, & \text{otherwise} \end{cases} \quad (5.9)$$

In the equation, $dist(a, b)$ counts the number of edge needed to be traveled from user a to user b in the social bookmarking network. We use the reciprocal of the peer distance to represent the fact that friends who are far away from the user tell us less about the user's interests. In the case where no paths exists between two users, $FanFactor$ would be 0, meaning that the two users are independent of each others in the network.

5.6 User-oriented Website Ranking

The problem of bookmark recommendation can be viewed as predicting whether a user will like a website not inside his/her bookmark collection (i.e., predicts the missing ratings in the User-Website matrix). Using the missing data prediction method [51], we can generate website ratings for each user based on the computed user and item similarities.

Given a user u, we can find out a set of similar users $S(u)$ as follows:

$$S(u) = \{u_a | usim'(u_a, u) > \eta, u_a \neq u\}. \tag{5.10}$$

Similarly for a website i, we can find out a set similar websites $S(i)$ as follows:

$$S(i) = \{i_k | isim(i_k, i) > \theta, i_k \neq i\}. \tag{5.11}$$

$\eta, \theta \in [0, 1]$ are used to avoid selecting users with very low similarity values. Then the prediction $P(r_{u,i})$ (i.e., the rating of website i by u) can be computed using one of the following cases, depending on whether set $S(u)$ and $S(i)$ are empty or not:

Case 1, $S(u) \neq \emptyset \wedge S(i) \neq \emptyset$:

$$\begin{aligned} P(r_{u,i}) =& \lambda \times (\overline{u} + \frac{\sum_{u_a \in S(u)} usim'(u_a, u) \cdot (r_{u_a,i} - \overline{u}_a)}{\sum_{u_a \in S(u)} usim'(u_a, u)}) + \\ & (1-\lambda) \times (\overline{i} + \frac{\sum_{i_k \in S(i)} isim(i_k, i) \cdot (r_{u,i_k} - \overline{i}_k)}{\sum_{i_k \in S(i)} isim(i_k, i)}). \end{aligned} \tag{5.12}$$

Case 2, $S(u) \neq \emptyset \wedge S(i) = \emptyset$:

$$P(r_{u,i}) = \overline{u} + \frac{\sum\limits_{u_a \in S(u)} usim'(u_a, u) \cdot (r_{u_a,i} - \overline{u}_a)}{\sum\limits_{u_a \in S(u)} usim'(u_a, u)}. \tag{5.13}$$

Case 3, $S(u) = \emptyset \wedge S(i) \neq \emptyset$:

$$P(r_{u,i}) = \overline{i} + \frac{\sum\limits_{i_k \in S(i)} isim(i_k, i) \cdot (r_{u,i_k} - \overline{i}_k)}{\sum\limits_{i_k \in S(i)} isim(i_k, i)}. \tag{5.14}$$

Case 4, $S(u) = \emptyset \wedge S(i) = \emptyset$:

$$P(r_{u,i}) = \lambda \times \overline{r}_u + (1 - \lambda) \times \overline{r}_i. \tag{5.15}$$

Since we are predicting the missing rating in the User-Website matrix, all the rating-related variables (i.e., $\overline{u}$, $r_{u_a,i}$, $\overline{i}$, r_{u,i_k} etc.) inside the above equations are coming from the same matrix.

The above cases try to make use of all information (i.e., similar users or websites) whenever possible. In case there are no similar users $S(u)$, the prediction is done solely based on similar websites $S(i)$, and vice versa. For the case where no similar users nor websites exist, the prediction is done by using the average rating of the user and the website.

Using these ratings, we can rank the websites not bookmarked by users and the recommendations are simply the top-K ranked websites for each user.

5.7 Evaluation

5.7.1 Bookmark Data

We manually crawled[2] the bookmarks on Del.icio.us. We have collected a data set containing more than 700,000 records from over 500 users in the following format:

$$< User\_ID, URL\_Hash, Title, Tags >$$

$User\_ID$ is a unique identifier of the user who created this bookmark record. URL_Hash is the URL of the bookmarked website in hashed format. $Title$ is the title of the website in the bookmark record. $Tags$ contain a list of terms used by the user to describe the website.

5.7.2 Social Network

We also collected the relationship among the bookmarked users from del.icio.us. Figure 5.2 is a sample page showing the network of a dummy user (ABC). The social network of a delicious user is divided into "Network" and "Fans". The "Network" list are users who ABC is a fan of them. The "Fans" list are uses who are fans of ABC. In this paper, we only collect user from the "Network" list and is stored in the following format:

$$< User\_ID, Network_List > .$$

5.7.3 Subjective Tag List

A sentiment lexicon is required in our proposed bookmark recommendation process. Although there exist some automated lexicon generation algorithms [32, 37, 21], the lexicon generated from these methods tends to be too noisy for the task of tag sentiment analysis. As a result, we have manually constructed

[2]Crawled on 28-Oct-2008

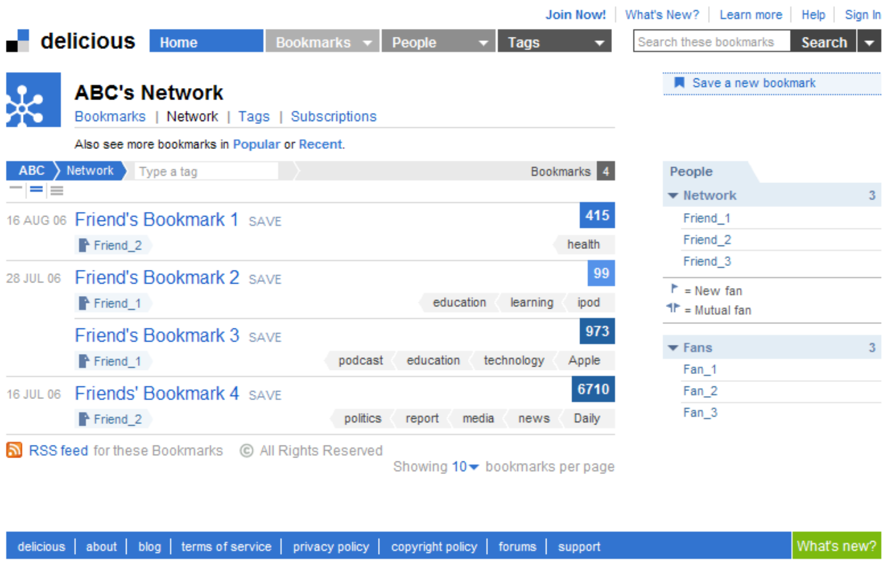

Figure 5.2: Sample User Network Page

a subjective tag list for evaluation purposes. The construction process is similar to the automated approaches in the sense that a list of seed words is used. However, instead of expanding the list automatically, we manually followed the synonym network of the dictionary and pick only the terms that are suitable to be considered as a subjective tag. The resulting list is list containing 238 English terms. We do not distinguish the polarity (i.e., positive or negative) of a term in this list.

Table 5.3 summaries the statistics of our data. The network size of a user u is the number of other user in the system who user u is a fan of him/her. Bookmarks with subjective tag counts bookmarks with at least one tag in the subjective tag list.

5.7.4 Subjective Tag Detection

In order to analyze the performance of our proposed subjective tag detection algorithm, we have extracted all the bookmark records that contain at least one tag in our subjective tag list.

	Count
User	518
Network Size (Per User)	8.52
Bookmark	703,089
Bookmark (Per User)	1,357
Bookmark with Subjective Tag	13,790
Average Tag (Per Bookmark)	3.42
Unique Tag w/ Freq. ≥ 50	3,877
Unique URL	432,420
URL w/ Freq. ≥ 2	58,422
URL w/ Freq. ≥ 10	4,699
Freq. of Most Occurred URL	224

Table 5.3: Data Statistics

Since the size of the extracted data is too large for us to label manually, we have sampled 3 datasets each containing 200 bookmark records. We go through each record and label whether each tag is used to express subjective feeling of the user. Table 5.4 shows the statistics of the datasets.

Dataset	1	2	3
Number of records	200		
Tags	1142	1108	1139
Subjective Tags (Labeled)	205	201	206

Table 5.4: Dataset Statistics

To the best of our knowledge, there is yet a method for tag subjectivity detection. Therefore, we use the simplest and straightforward approach as the baseline for comparison. The approach is to regard every tag appeared in the sentiment lexicon as subjective. We compare the detected subjective tags of both methods with our labeled data and use the traditional precision, recall and recall to measure the performances. We use $\varepsilon = 0.1$ as the $PCOR$ threshold in our algorithm. Table 5.5

		Dataset 1	Dataset 2	Dataset 3	Average
STD	Precision	0.94	0.92	0.95	0.94
	Recall	0.87	0.84	0.89	0.87
	F-Score	0.90	0.88	0.92	**0.90**
Baseline	Precision	0.87	0.83	0.90	0.87
	Recall	0.89	0.85	0.91	0.88
	F-Score	0.88	0.84	0.90	**0.87**

Table 5.5: Subjective Tag Detection Performance

summaries the results.

From the results, we can see that our algorithm improves the detection precision by a significant margin (7%) with only a small drop of recall (1%) over the baseline. The overall F-Score's improvement is 3%. This suggests that our algorithm is accurate as well as efficient in extracting subjective tags from social bookmarks.

5.7.5 Bookmark Recommendation Quality

The most important goal of bookmark recommendation system is to provide useful website suggestions to user. However, there is not a well established method for evaluating this kind of systems. Previous works use case studies or some ad-hoc methods to analyze the system. The lack of a standard evaluation method makes comparisons among bookmark recommendation systems difficult.

Traditional collaborative filtering based recommender systems can be evaluated by metrics such as Mean Absolute Error (MAE) or Root Mean Square Error (RMSE) [77] but bookmark recommendation systems cannot. This is because we do not have the concept of explicit rating in bookmark recommendation system. Given a bookmark recommendation, a user will either be interested or not interested. We say that the system is good if most of its recommendations are interested by the target users. However, we cannot ask all the users to help evaluating our system by providing information as to whether our recommendations are good or not. This is time consuming, inefficient and not always feasible.

Given the above difficulties, we propose to use the users' bookmarks themselves to evaluate the system's recommendation quality. The idea is to simply split the bookmark data into two partitions, namely, training dataset and testing dataset. The splitting is done is such a way that every user's bookmarks appear in both partitions. The training dataset provide us information to compute similarities and generate recommendations. The testing dataset allows us to evaluate our recommendation quality.

However, we all know that it is impossible for any systems to generate recommendations that perfectly meet the bookmarks in the testing dataset. Therefore, our way to measure quality is to check, for every user, the number of bookmarks in the testing dataset that have received a predicted rating larger than a threshold. We define the **Bookmark Recommendation Assessment Index (BRAI)** as follows:

$$BRAI = \frac{\sum_{u \in U} |\{b | b \in T(u) \wedge P(u,b) > \omega\}|}{\sum_{u \in U} |T(u)|}. \tag{5.16}$$

In the equation, $T(u)$ is the set of bookmarks in testing dataset of user u. $BRAI$ measures the portion of users who have received at least one correct recommendation. Under this definition, higher the $BRAI$ means better the recommendation quality.

5.7.6 System Evaluation

In this section, we use the BRAI metric to evaluate our system. However, due to memory and computation power constraints, it is infeasible for us to take the whole dataset, which contains over 700,000 bookmarks, for evaluation. Therefore, we use only a subset of all available data to evaluate our system. We prune away websites which appears less than 10 times and tags that appear less than 50 times from the dataset. This reduces the problem to a solvable size.

The partitioning of the bookmark data set is done as follows. For each user who has more than 100 bookmarks, we take 50 of them to the testing dataset. Users having less than 100 bookmarks are not included in the testing dataset but are left in the training dataset. This is because there are either too few evidences to learn their preferences or there are not enough testing data for us to have a reasonable evaluation. They are left in the training dataset to help predicting our users' ratings.

We want to answer a few questions through a set of experiments: 1) what is the effect of using both User-Website and User-Tag matrix instead of using just one of them? 2) Does social network information works well with semantic information? 3) Can tag sentiment analysis really improve bookmark recommendation? In other to answer these questions, we setup four sets of experiments each differ from each other by the use of our proposed methods. The parameter or thresholds for our experiments are empirically set as: $\lambda = 0.7$, $\alpha = \beta = 0.5$, $\gamma = \sigma = 10$, $\eta = \theta = 0.1$, $\omega = 0.7$. The results are summarized in table 5.6.

Experiment Set	**BRAI**
1 (baseline)	0
2 (website + tag)	0.14
3 (social network)	0.14
4 (include STD)	**0.25**

Table 5.6: Bookmark Recommendation Quality

Experiment set 1 is the baseline of our test which uses only User-Website matrix (without subjective tag detection) for user and website similarity computation. Experiment set 2 includes also the User-Tag matrix to demonstrate the performance of our recommendation system when using semantic information only. Experiment set 3 adds $FanFactor$ on top of set 2 to show how well social network information plays in the system. The last set is the complete system that includes our whole framework as well as subjective tag detection.

Result Interpretation

From the result we can see that the performance of the bookmarking system achieved the best when everything including STD is used for similarity computation. We suspect that the similar performance achieved by both experiment set 2 and 3 are due to the limited size of the dataset. Not much peer interests can be used for better recommendations. The reason why the baseline get 0 score is because the data in the User-Website matrix is so sparse that similarity computed based on this matrix is not good enough to grouping similar users together, making recommendation difficult.

5.8 Summary

In this chapter, we address the bookmark recommendation problem with the use of tagging information in social bookmarking systems. In contrast to existing systems, our proposed framework takes into account the sentiment expressed in bookmark tags. This allows collaborative filtering algorithms, which depend heavily on existing user ratings, to better capture the taste of users. We evaluate our proposed sentiment tag detection algorithm and the entire bookmark recommendation algorithm with a real-world data set and the result shows that our proposed work is capable of achieving good results.

Chapter 6

Conclusion and Future Work

In this first part of this thesis, we have presented a Feature-Opinion Association (FOA) algorithm to improve the sentiment analysis results. The algorithm make use of a simple observation that some opinion words are more related to a particular feature than the others. It maximizes the sum of the relevance scores between features and opinion words. We further proposed 6 relevance measures that make use of the structural information of sentences as well as the statistical information collected from a commercial review web site. The proposed FOA algorithm can be used in both the lexicon generation and sentiment classification process. Our method is tested using a publicly accessible data set and the results show that it is effective in improving the sentiment analysis accuracy over the traditional methods where classification is done using all opinion words.

In the future, we plan to extend our study of FOA to opinions that are expressed in other forms (e.g. verbs and nouns). This is important because opinions are sometimes expressed in various ways. For example, the sentence

"I hate the user-interface."

clearly expressed a negative opinion on the user-interface. The current study of sentiment analysis, considering only opinions expressed in adjectives and adverbs, will not be able to handle the above situation. This is also suggested by the imperfect sentiment classification results achieved by human FOA.

Another important work is to study the effect of association truncation in sentence-level sentiment analysis. Current FOA assumes features and their related opinions to be existed in the same sentence. However, this assumption may not always holds. It can be shown in the following example:

"This camera has 32 MB internal memory. It is too small for practical use."

The word "small", which is opinion expressing, does not appeared at the same sentence as the feature it describe. In this case, we need to resolve what "It" is talking about. However, due to the difficulty of natural language processing, this is not a straight-forward task. So an extensive study should be carried out to solve the problem.

We believe that by handling the above problems, we will be allowed to exploit more information from the review text and thus further improve the sentiment analysis results.

In the next part of our thesis, we studied the properties of social games and derived a framework for designing social games. We further designed two social games that could possibly improve the opinion mining process. The first game, OpinionMatch, is a game that directly outsource the entire opinion mining problem by human computation which turn out to have little entertainment value. The second game, FeatureGuess, improve upon the first game which focus only on a subproblem in opinion mining, namely, Feature-Opinion Association (FOA). Although the scope of problem being solved by FeatureGuess is smaller than OpinionMatch, the game is designed to attract people to play. We believe that the entertainment value of a social game heavily affect the practical value of it.

Finally, we have presented a framework for bookmark recommendation systems. The framework combines both semantic and social network information to generate recommendations to users. We observe that a significant amount tags in an existing system are subjective in nature and proposed a novel tag subjectivity detection algorithm to handle it. The subjectivity information is incorporated into the recommendation process in order to have a better understand of user's preferences. We present a standard metric called BRAI for evaluating bookmark recommendation quality. Evaluation results show that our proposed work is effective in generating useful recommendations.

In the future, we plan to extend the study of sentiment analysis in bookmark recommendation system so that we can order users' bookmark' according to their level of interest. We would also like to study other models of user and website similarities that can improve recommendation quality.

Another important topic of research is on the reduction of computational complexity of bookmark recommendation algorithms. Since the current method is limited to handle small datasets because of the space and time constraints, if new methods are developed with lower computational complexity, the system could become even more useful.

Appendix A

List of Symbols and Notations

Symbol	Meaning
Feature Opinion Association	
c_{+ve}	positive conjunction frequency
c_{-ve}	negative conjunction frequency
COF	co-occurrence frequency
COR	co-occurence ratio
F	set of features
$matched$	set of words matched to a particular feature
rel	relevance score
th	relevance threshold
W	set of opinion words
Social Game for Opinion Mining	
$\mathcal{A}$	attribute
AC	an action
$\mathcal{ACO}$	outcome domain of an action
$\mathcal{ACS}$	set of actions of a role
$\mathcal{ACT}$	type of an action
$\mathcal{ANS}$	answer extraction procedure

Symbol	Meaning
$\mathcal{C}$	constraints of a game
$\mathcal{D}$	data
$\mathcal{E}$	set of problems to solve
$\rceil$	data structure for answer extraction
$eSel$	problem selection procedure
$\mathcal{F}$	answer domain
$\mathcal{G}$	answer correctness function
$\mathcal{GM}$	mechanism of a game
$\mathcal{I}$	player's input
$\mathcal{KW}$	knowledge a role can has
$\mathcal{M}$	metadata
$\mathcal{O}$	player's output
p_j^k	the j-th player assigned with role k
$pNum$	number of player in a game
$pSel$	player selection procedure
$\mathcal{R}$	player roles
$\mathcal{T}$	data type
$\mathcal{UI}$	design characteristics of user interface
$\mathcal{V}$	value
$\mathcal{V}$	verification procedure
$\mathcal{W}$	reward for solving a problem
τ	answer acceptance frequency threshold
$tMax$	maximum duration of a game
Tag Sentiment Analysis for Social Bookmark Recommendation System	
a, b	users

Symbol	Meaning
α	weighting of user-website matrix
β	weighting of CF-based user similarity
$I(a)$	items rated by user a
λ	weighting of user-based similarity
$\overline{r}_a$	average rating of user a
$\overline{r}_i$	average rating of item i
$r_{a,i}$	rating on item i given by user a
$U(i)$	users who have rated item i

Table A.1: List of Symbols and Notations

□ **End of chapter.**

Appendix B

List of Publications

- **K. T. Chan** and I. King, "Lets Tango – Finding the Right Couple for Feature-Opinion Association in Sentiment Analysis," in Proc. Advances in Knowledge Discovery and Data Mining 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009
- I. King, J. Li, and **K. T. Chan**, "A Brief Survey of Computational Approaches in Social Computing," in Proc. Proceedings of the 2009 International Joint Conference on Neural Networks (IJCNN2009), 2009.
- W. Wei, **K. T. Chan**, I. King, and J. H. Lee, "RATE: a Review of Reviewers in a Manuscript Review Process," in Proc. Proceedings to the IEEE/WIC/ACM WI-2008 (WI2008), Sydney, Australia, 2008.
- **K. T. Chan** and I. King, "Tag Sentiment Analysis for Social Bookmark Recommendation System" (In preparation)
- **K. T. Chan** I. King and Man-Ching Yuen, "Mathematical Modeling of Social Games" (In submission)

□ **End of chapter.**

Bibliography

[1] Del.icio.us. http://del.icio.us/.

[2] Google Image Labeler. http://images.google.com/imagelabeler/.

[3] GWAP. http://www.gwap.com/gwap/.

[4] Mindpixel project. http://www.mindpixel.org.

[5] The open mind project. http://www.openmind.org.

[6] M. Ames and M. Naaman. Why we tag: motivations for annotation in mobile and online media. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 971–980, New York, NY, USA, 2007. ACM.

[7] J. Bigham, R. Kaminsky, R. Ladner, O. Danielsson, and G. Hempton. WebInSight:: making web images accessible. In *Proceedings of the 8th international ACM SIGACCESS Conference on Computers and Accessibility*, pages 181–188. ACM New York, NY, USA, 2006.

[8] K. Bischoff, C. S. Firan, W. Nejdl, and R. Paiu. Can all tags be used for search? In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 193–202, New York, NY, USA, 2008. ACM.

[9] S. Blair-Goldensohn, K. Hannan, R. McDonald, T. Neylon, G. Reis, and J. Reynar. Building a sentiment summarizer

for local service reviews. In *WWW2008 Workshop on NLP Challenges in the Information Explosion Era*, 2008.

[10] S. Branavan, H. Chen, J. Eisenstein, and R. Barzilay. Learning document-level semantic properties from free-text annotations. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2008.

[11] J. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *Learning*, 9:309–347.

[12] S. Casey, B. Kirman, and D. Rowland. The gopher game: a social, mobile, locative game with user generated content and peer review. In *International Conference on Advances in Computer Entertainment Technology*, pages 9–16, 2007.

[13] L. J. Chen, B.-C. Wang, K. T. Chen, I. King, and J. H.-M. Lee. An analytical study of puzzle selection strategies for the esp game. In *Proceedings to the IEEE/WIC/ACM WI-2008 (WI2008)*, Sydney, Australia, December, 2008.

[14] L. A. Dabbish. Jumpstarting relationships with online games: evidence from a laboratory investigation. In B. Begole and D. W. McDonald, editors, *CSCW*, pages 353–356. ACM, 2008.

[15] X. Ding and B. Liu. The utility of linguistic rules in opinion mining. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 811–812, New York, NY, USA, 2007. ACM.

[16] X. Ding, B. Liu, and P. S. Yu. A holistic lexicon-based approach to opinion mining. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 231–240, New York, NY, USA, 2008. ACM.

[17] dpreview. Digital camera reviews and news: Digital photography review: Forums, glossary, faq:, 2007. [Online; accessed 8-October-2007].

[18] C. Dugan, M. Muller, D. R. Millen, W. Geyer, B. Brownholtz, and M. Moore. The dogear game: a social bookmark recommender system. In *GROUP '07: Proceedings of the 2007 international ACM conference on Supporting group work*, pages 387–390, New York, NY, USA, 2007. ACM.

[19] C. Dugan, M. J. Muller, D. R. Millen, W. Geyer, B. Brownholtz, and M. Moore. The dogear game: a social bookmark recommender system. In T. Gross and K. Inkpen, editors, *GROUP*, pages 387–390. ACM, 2007.

[20] K. Eguchi and V. Lavrenko. Sentiment retrieval using generative models. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 345–354, 2006.

[21] A. Esuli and F. Sebastiani. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, pages 417–422, 2006.

[22] O. Feiguina and G. Lapalme. Query-based summarization of customer reviews. *Lecture Notes in Computer Science*, 4509:452, 2007.

[23] G. W. Furnas, C. Fake, L. von Ahn, J. Schachter, S. Golder, K. Fox, M. Davis, C. Marlow, and M. Naaman. Why do tagging systems work? In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 36–39, New York, NY, USA, 2006. ACM.

[24] C. Gentry, Z. Ramzan, and S. G. Stubblebine. Secure distributed human computation. In J. Riedl, M. J. Kearns,

and M. K. Reiter, editors, *ACM Conference on Electronic Commerce*, pages 155–164. ACM, 2005.

[25] A. Ghose and P. Ipeirotis. Estimating the Socio-Economic Impact of Product Reviews: Mining Text and Reviewer Characteristics. *Information Systems Research*, 2008.

[26] A. Ghose, P. Ipeirotis, and A. Sundararajan. Opinion mining using econometrics: A case study on reputation systems. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 416, 2007.

[27] S. Golder and B. Huberman. The Structure of Collaborative Tagging Systems. *Arxiv preprint cs.DL/0508082*, 2005.

[28] S. A. Golder and B. A. Huberman. The structure of collaborative tagging systems. *CoRR*, abs/cs/0508082, 2005.

[29] J. Herlocker, J. Konstan, and J. Riedl. An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms. *Information Retrieval*, 5(4):287–310, 2002.

[30] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, New York, NY, USA, 1999. ACM.

[31] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, New York, NY, USA, 2004. ACM Press.

[32] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, New York, NY, USA, 2004. ACM Press.

[33] M. Hu and B. Liu. Mining opinion features in customer reviews. In *Proceedings of AAAI-04, the 19th National Conference on Artificial Intellgience*, San Jose, US, 2004.

[34] N. Jindal and B. Liu. Opinion spam and analysis. In *Proceedings of the international conference on Web search and web data mining*, pages 219–230. ACM New York, NY, USA, 2008.

[35] N. Kaji and M. Kitsuregawa. Building lexicon for sentiment analysis from massive collection of HTML documents. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1075–1083, 2007.

[36] R. Kanawati and M. Malek. A multi-agent system for collaborative bookmarking. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 1137–1138, New York, NY, USA, 2002. ACM.

[37] H. Kanayama and T. Nasukawa. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *EMNLP '06*, 2006.

[38] S. Kim and E. Hovy. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, pages 1–8, 2006.

[39] S. Kim and E. Hovy. Crystal: Analyzing predictive opinions on the web. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1056–1064, 2007.

[40] S.-M. Kim and E. Hovy. Determining the sentiment of opinions. In *Proceedings COLING-04, the Conference on Computational Linguistics*, Geneva, CH, 2004.

[41] A. Krzywinski, W. Chen, and A. Helgesen. Agent architecture in social games the implementation of subsumption architecture in diplomacy. In C. Darken and M. Mateas, editors, *AIIDE*. The AAAI Press, 2008.

[42] L. Ku, Y. Lo, H. Chen, and T. Taipei. Using Polarity Scores of Words for Sentence-level Opinion Extraction. In *Proceedings of NTCIR-6 Workshop Meeting*, 2007.

[43] S. Laumer, A. von Stetten, A. Eckhardt, and T. Weitzel. Online gaming to apply for jobs - the impact of self- and e-assessment on staff recruitment. In *HICSS*, pages 1–10. IEEE Computer Society, 2009.

[44] E. L. M. Law and L. von Ahn. Input-agreement: A new mechanism for data collection using human computation games. In *ACM CHI*, 2009.

[45] E. L. M. Law, L. von Ahn, R. B. Dannenberg, and M. Crawford. TagATune: A Game for Music and Sound Annotation. In *International Conference on Music Information Retrieval*, 2007.

[46] D. Lee, O. Jeong, and S. Lee. Opinion mining of customer feedback data on the web. In *Proceedings of the 2nd international conference on Ubiquitous information management*

and communication, pages 230–235. ACM New York, NY, USA, 2008.

[47] D. B. Lenat. Cyc: a large-scale investment in knowledge infrastructure. *Commun. ACM*, 38(11):33–38, 1995.

[48] H. Lieberman, D. Smith, and A. Teeters. Common Consensus: a web-based game for collecting commonsense goals. In *ACM Workshop on Common Sense for Intelligent Interfaces*, 2007.

[49] C.-W. Lin, K. T. Chen, L. J. Chen, I. King, and J. H.-M. Lee. An analytical approach to optimizing the utility of esp games. In *Proceedings to the IEEE/WIC/ACM WI-2008 (WI2008)*, Sydney, Australia, December, 2008.

[50] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 342–351, New York, NY, USA, 2005. ACM Press.

[51] H. Ma, I. King, and M. R. Lyu. Effective missing data prediction for collaborative filtering. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 39–46, New York, NY, USA, 2007. ACM.

[52] M. Mandel and D. Ellis. A web-based game for collecting music metadata. *Journal of New Music Research*, 37(2):151–165, 2008.

[53] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of english: the penn treebank. *Comput. Linguist.*, 19(2):313–330, 1993.

[54] C. Marlow, M. Naaman, D. Boyd, and M. Davis. HT 06, tagging paper, taxonomy, Flickr, academic article, to read. In *Conference on Hypertext and Hypermedia: Proceedings of the seventeenth conference on Hypertext and hypermedia*. Association for Computing Machinery, Inc, One Astor Plaza, 1515 Broadway, New York, NY, 10036-5701, USA,, 2006.

[55] R. Mihalcea, C. Banea, and J. Wiebe. Learning multilingual subjective language via cross-lingual projections. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 976, 2007.

[56] D. R. Millen, J. Feinberg, and B. Kerr. Dogear: Social bookmarking in the enterprise. In R. E. Grinter, T. Rodden, P. M. Aoki, E. Cutrell, R. Jeffries, and G. M. Olson, editors, *CHI*, pages 111–120. ACM, 2006.

[57] M. J. Muller. Comparing tagging vocabularies among four enterprise tag-based services. In *GROUP '07: Proceedings of the 2007 international ACM conference on Supporting group work*, pages 341–350, New York, NY, USA, 2007. ACM.

[58] NLTK. Natural language processing toolkit.

[59] J. Orkin and D. Roy. The restaurant game: Learning social behavior and language from thousands of players online. *Journal of Game Development*, 3(1):39–60, December 2007.

[60] B. Pang and L. Lee. *Opinion mining and sentiment analysis*. Now Publishers, 2008.

[61] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86,

Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[62] E. Peck, M. Riolo, and C. Cusack. Wildfire wally: a volunteer computing game. In *Future Play '07: Proceedings of the 2007 conference on Future Play*, pages 241–242, New York, NY, USA, 2007. ACM.

[63] D. Pemberton, T. Rodden, and R. Procter. GroupMark: A WWW Recommender System Combining Collaborative and Information Filtering. In *Proceedings of the 6th ERCIM Workshop*, 2000.

[64] S. Raaijmakers, K. Truong, and T. Wilson. Multimodal Subjectivity Analysis of Multiparty Conversation. In *Proceedings of EMNLP*, 2008.

[65] E. Rader and R. Wash. Influences on tag choices in del.icio.us. In *CSCW '08: Proceedings of the ACM 2008 conference on Computer supported cooperative work*, pages 239–248, New York, NY, USA, 2008. ACM.

[66] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, New York, NY, USA, 1994. ACM.

[67] J. Rucker and M. J. Polanco. Siteseer: personalized navigation for the web. *Commun. ACM*, 40(3):73–76, 1997.

[68] M. Shaikh, H. Prendinger, and I. Mitsuru. Assessing sentiment of text by semantic dependency and contextual valence analysis. *Lecture Notes in Computer Science*, 4738:191, 2007.

[69] U. Shardanand and P. Maes. Social information filtering: algorithms for automating "word of mouth". In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[70] R. Shaw and P. Schmitz. Community annotation and remix: a research platform and pilot deployment. In *Proceedings of the 1st ACM international workshop on Human-centered multimedia*, pages 89–98. ACM New York, NY, USA, 2006.

[71] A. Simon. Social heroes: games as apis for social interaction. In S. Tsekeridou, A. D. Cheok, K. Giannakis, and J. Karigiannis, editors, *DIMEA*, volume 349 of *ACM International Conference Proceeding Series*, pages 40–45. ACM, 2008.

[72] V. Sindhwani and P. Melville. Document-word co-regularization for semi-supervised sentiment analysis. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 1025–1030. IEEE Computer Society Washington, DC, USA, 2008.

[73] K. Siorpaes and M. Hepp. Games with a Purpose for the Semantic Web. *IEEE Intelligent Systems*, 23(3):50–60, May/June 2008.

[74] L. Stoilova, T. Holloway, B. Markines, A. G. Maguitman, and F. Menczer. Givealink: mining a semantic network of bookmarks for web search and recommendation. In *LinkKDD '05: Proceedings of the 3rd international workshop on Link discovery*, pages 66–73, New York, NY, USA, 2005. ACM.

[75] D. Stork and L. C.P. Open mind animals: Ensuring the quality of data openly contributed over the world wide web.

In *Learning from Imbalanced Data Sets: Papers from the AAAI Workshop*, pages 4–9, 2000.

[76] Q. Su, D. Pavlov, J. Chow, and W. Baker. Internet-scale collection of human-reviewed data. In *Proceedings of the 16th international conference on World Wide Web*, pages 231–240. ACM New York, NY, USA, 2007.

[77] D. Sun, T. Zhou, R. Liu, C. Jia, J. Liu, and B. Wang. Information filtering based on transferring similarity. *Arxiv preprint arXiv:0807.4495*, 2008.

[78] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[79] L. Von Ahn, M. Blum, N. Hopper, and J. Langford. CAPTCHA: Using hard AI problems for security. *Lecture notes in computer science*, pages 294–311, 2003.

[80] L. von Ahn and L. Dabbish. Labeling Images with a Computer Game. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 319–326, 2004.

[81] L. von Ahn and L. Dabbish. Designing games with a purpose. *Communications of the ACM*, 51(8):58–67, August 2008.

[82] L. von Ahn, S. Ginosar, M. Kedia, and M. Blum. Improving Image Search with PHETCH. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1209–1212, 2007.

[83] L. von Ahn, M. Kedia, and M. Blum. Verbosity: A Game for Collecting Common-Sense Facts. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 75–78, 2006.

[84] L. von Ahn, R. Liu, and M. Blum. Peekaboom: A Game for Locating Objects in Images. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 55–64, 2006.

[85] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science*, 321(5895):1465, 2008.

[86] F. Wang. Social Computing: Fundamentals and applications. In *IEEE International Conference on Intelligence and Security Informatics, 2008. ISI 2008*, 2008.

[87] F. Wang, K. Carley, D. Zeng, and W. Mao. Social computing: From social informatics to social intelligence. *IEEE Intelligent Systems*, 22(2):79–83, 2007.

[88] R. Wash and E. Rader. Public bookmarks and private benefits: An analysis of incentives in social computing. *Proceedings of the American Society for Information Science and Technology*, 44(1):1–13, 2007.

[89] Wikipedia. Delicious (website) — wikipedia, the free encyclopedia, 2008. [Online; accessed 30-October-2008].

[90] Wikipedia. Stumbleupon — wikipedia, the free encyclopedia, 2008. [Online; accessed 30-October-2008].

[91] D. H. Wilson, A. C. Long, and C. Atkeson. A context-aware recognition survey for data collection using ubiquitous sen-

sors in the home. In *ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 1865–1868, 2005.

[92] Y. Xia, R. Xu, K. Wong, and F. Zheng. The Unified Collocation Framework for Opinion Mining. In *Machine Learning and Cybernetics, 2007 International Conference on*, volume 2, 2007.

[93] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, and Z. Chen. Scalable collaborative filtering using cluster-based smoothing. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 114–121, New York, NY, USA, 2005. ACM.

[94] Y. Yang, B. B. Zhu, R. Guo, L. Yang, S. Li, and N. Yu. A comprehensive human computation framework: with application to image labeling. In A. El-Saddik, S. Vuong, C. Griwodz, A. D. Bimbo, K. S. Candan, and A. Jaimes, editors, *ACM Multimedia*, pages 479–488. ACM, 2008.

[95] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, page 427, Washington, DC, USA, 2003. IEEE Computer Society.

[96] W. Zhang, C. Yu, and W. Meng. Opinion retrieval from blogs. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 831–840. ACM New York, NY, USA, 2007.