

# Final Year Project Report

---

## Reputation Algorithms for the Social Web

John Patrick Bracken (09442961)

---

A thesis submitted in part fulfilment of the degree of

**BA/BSc (hons) in Computer Science**

**Supervisor:** Dr. Michael O'Mahony



UCD School of Computer Science and Informatics  
College of Science  
University College Dublin

March 12, 2014

# Specification

---

**Subject:** Social Network Analysis, Reputation Systems

**Coverage** Social Network Analysis, Reputation Systems

**Project Type:** Design and Implementation

**Software Requirements:** Java, Linux, MySQL (or other)

**Hardware Requirements:** Laptop for development. Access to server will be provided if necessary.

**Preassigned:** No

## General Information

The social web reflects an important paradigm shift in the nature of our online transactions. We increasingly rely on the opinions of others to mediate these transactions and as such the reliability of these users becomes an important indicator of quality. Thus, the concept of user reputation has become increasingly important in the context of today's social web. Recently, there has been considerable research on various approaches to model the reputation of users as they participate in a diverse array of online interactions.

The goal of this project is to design, implement and evaluate reputation algorithms for users of the Stack Exchange network, a popular group of social Q&A sites. There are currently over 80 topical Stack Exchange websites, hosting almost 2 million users. Some 3.8 million questions have been posed, eliciting 7.7 million answers. Users are permitted to post questions that can be answered by other users in the community. Each answer given can be voted up or down by others and the questioner can choose to highlight a single answer as correct, indicating the question has been answered satisfactorily or that answer was the best answer provided. The availability of such data can be leveraged to estimate the reputation of users; for example, if the answers provided by a particular user are frequently deemed to be correct and/or receive many positive votes from the community, then this provides an indication that the user is knowledgeable about particular subject matters.

## Mandatory:

- Download Stack Exchange data (<http://data.stackexchange.com/>) - data from three sites should be obtained.
- Implementation of reputation algorithms from the literature.
- Evaluation: for each dataset, compare the performance of these algorithms to the user reputation model currently used on Stack Exchange.

## Discretionary:

- Predict the correct answers to questions based on user reputation.
- Evaluate the accuracy of predicted answers.

- Perform user-trials and correlate prediction performance with offline metrics.

**Exceptional:**

Any (but not limited to) the following:

- Propose and implement enhancements to improve algorithm performance.
- Analyse the robustness of the reputation algorithms against attack.

**Reading:**

- Stack Exchange: <http://stackexchange.com/>
- Stack Exchange Data Explorer: <http://data.stackexchange.com/>
- A Model of Collaboration-based Reputation for the Social Web ICWSM 2013
- Trust Among Strangers in Internet Transactions: Empirical Analysis of eBays Reputation System - Advances in Applied Microeconomics 2012 (attached)

# Abstract

---

In this project I intend to compare the performance of generic reputation algorithms using the Stack Exchange Question and Answer sites' open-sourced data dumps. These algorithms will include a simple inbound Weighted Sum and Page and Brin's PageRank algorithm. Performance will be analysed by evaluating correlation between these algorithms' scores and the bespoke Stack Exchange reputation model, and by examining the distribution of reputation scores across users.

It is found that the Weighted Sum approach significantly out-performs the PageRank method, and is an excellent generic alternative to Stack Exchange's ad-hoc model.

# Acknowledgements

---

My thanks go out to my project supervisor Dr. Michael P. O'Mahony for his help and guidance throughout the course of this project. His kind support epitomises the nature of a good educator and I am eternally thankful to have had him as a supervisor.

I would also like to thank my family and friends for their support, in particular Sam Dunne, Shane Quigley and my parents for being good sounding boards, and my two dogs Sasha and Trudy for taking me on regular walks.

Finally, I would like to express my gratitude to the Stack Exchange network for making their data so readily available and easy to work with.

# Table of Contents

---

<b>Specification</b>	1
<b>Abstract</b>	3
<b>1 Introduction</b>	7
<b>2 Background Research</b>	9
2.1 Introduction	9
2.2 Trust and reputation	9
2.3 Reputation algorithms	9
2.4 Case studies	12
2.5 Conclusion	15
<b>3 Creating a generic model of user reputation for the social web</b>	16
3.1 The producer-consumer model	16
3.2 The collaboration event	17
3.3 Building the collaboration graph	18
3.4 Computing reputation	19
3.5 Conclusion	20
<b>4 Design &amp; Implementation</b>	21
4.1 Project structure	21
4.2 Development tools	23
4.3 Obtaining the data	23
4.4 Computing reputation scores	25
<b>5 Testing &amp; Evaluation</b>	26
5.1 The datasets	26
5.2 Methodology	26
5.3 Analysis	27
5.4 Coverage	30
<b>6 Conclusions &amp; Future Work</b>	33

6.1	Fulfilment of project goals . . . . .	33
6.2	Conclusions . . . . .	34
6.3	Learning outcome . . . . .	35
6.4	Failings . . . . .	35
6.5	Future work . . . . .	35

# Chapter 1: Introduction

---

With the increasing use of the internet in our day-to-day tasks, it has become more and more important that we be able to verify the trustworthiness of the people we interact with. While the use of technologies such as public key encryption allow us to verify *who* we are talking to with reasonable confidence, we are still left with the problem of determining that person's trustworthiness as an individual—whether that be trust in their knowledge in a particular field, or that they can be relied upon to deliver a good or service to satisfaction.

To that end there has been considerable recent research in the fields of peer-to-peer trust and user reputation systems in social networks (1, 2, 4). There are numerous contexts in which an indication of trust may be desired online, from internet transactions to social Question & Answer sites. For the purposes of this project we will be focusing on the exchange of knowledge and expertise between users on the *Stack Exchange Question and Answer* network. We will define trust as a relationship between users, and reputation will refer to an aggregate score allocated to a user that reflects their trustworthiness.

In this project, I will be implementing a number of generic approaches to reputation (Weighted Sum and PageRank) and comparing their performance on the Stack Exchange datasets to each other and Stack Exchange's own proprietary reputation model. To demonstrate that evaluating the reputation of users is even necessary, it is important that we pay attention to the history of the Web's growth up to its present state, and to look at current trends to predict its future.

When the web first exploded into widespread use in the nineties, it was a static compilation of pages connected by hyper-links. Typically, websites were only published by universities, government organisations and large corporations, with content being controlled by their respective web-masters. It was much easier, then, to evaluate the trustworthiness of online resources; content related to hardware published by IBM was likely to be accurate, but IBM's advice on cake-baking may need to be taken with a pinch of salt.

Over time however, computers and internet technologies rapidly became much more sophisticated, and so-called web 2.0 technologies such as PHP and JavaScript emerged, which allowed end-users to interact dynamically with websites. At the same time, the number of internet users exploded, and the line between the roles of producer and consumer began to blur. Instead, anyone can now post on a social network, publish a music review to potentially millions of people, or share their opinions on YouTube videos, whether or not anyone else cares to read them.

In many cases this is not an issue. With many of these social networks, the users you interact with are already your friends, or people you know, and you will already have some degree of trust or distrust in them. In other cases, the material posted by others is obviously subjective or not entirely objective. But there are increasingly more of these social networks emerging where it is beneficial to know if another person is reputable.

To give an example, a lady named Alice asks a dog lovers' web-forum how much chocolate is safe to feed to her chihuahua named Charlie. She may get a range of different answers. Bob helpfully gives her the correct answer that chocolate is bad for humans and worse for dogs, and to feed Charlie treats made especially for dogs instead. Mallory, out of some bizarre sense of Schadenfreude, intentionally gives her a malicious answer framed as genuine advice. Alice does not know who to trust, but errs on the side of caution and Charlie is spared the



grim fate of diabetes.

Not all negatively impactful users are malicious. Some users may just be misinformed, or ignorant, or have the illusory superiority cognitive bias, in which a person overestimates their own abilities or knowledge (10). In the previous example, you may see answers where the user presents anecdotal evidence that they have been feeding their own dogs chocolate for years with no ill effects, and it is probably fine to feed them to Charlie. While well-intentioned, such responses are unhelpful to the community as a whole. They worsen the signal-to-noise ratio of the social web and being able to determine that these users are disreputable at a glance is beneficial.

While the previous example may seem minor, it serves to illustrate the potential dangers of this social web. Credulous users may take inaccurate information at face value, and in the worst case cause themselves or others harm, due to the maliciousness or ignorance of a stranger.

With over a billion active users and more than ten billion messages every day on Facebook *alone*<sup>1</sup>, for example, there is a considerable amount of potentially inaccurate information being spread across social networks, and it would be desirable to be able to filter some of the wheat from the chaff.

This filtering of unsatisfactory content can take many different forms. Favourable content may ‘rise to the top’ of a user’s news feed in order of best to worst. Unfavourable content might be given some visual de-emphasis to indicate its lower quality, or may even be hidden from the user altogether. Users might be able to customize how strict this filtering is, or create whitelists and blacklists of people they trust and distrust respectively. In addition, users can be encouraged to improve their contributions to a community by gaining additional privileges or prestige for improving their reputation.

The rest of this report is structured as follows. In chapter two, a description is given of some reputation algorithms at a high level, and some case studies are examined of reputation systems currently in use online. In chapter three there will be an in-depth description of the producer-consumer model, collaboration events and constructing the collaboration graph, and using this graph to compute reputation. In chapter four there will be a description of the design and implementation of this project. In chapter five, there will be a description and justification of the datasets and methodologies that were used, as well as a presentation of the results and findings. The final chapter presents the conclusions drawn in the course of this project, outlines criticisms of the project, and describes some possible future areas of research in the topic.

---

<sup>1</sup>Taken from ‘A focus on Efficiency’ by Facebook, Ericsson and Qualcomm 2013 (<http://internet.org/efficiencypaper>, September 16, 2013). Last accessed: March 10, 2014

# Chapter 2: Background Research

---

## 2.1 Introduction

The following chapter outlines the research undertaken into online reputation systems. Trust and reputation are explicitly defined, along with a brief explanation of their differences and why drawing a distinction between the two is important. Some research undertaken into user reputation algorithms that are currently in use across the web is presented. Finally, there are a number of case studies on sites that use a user reputation model, such as the Stack Exchange Network, eBay and Klout.

## 2.2 Trust and reputation

For the purposes of this project, *trust* is defined as a relationship between two parties. A trustor is an entity who places a certain amount of faith in the actions or knowledge of another entity (or the trustee). Trust is not a symmetric relationship, so while, for example, a student may trust a teacher's knowledge in their subject matter, the teacher will likely have less (or no) trust in their student's knowledge.

Trust can occur between numerous types of entities, such as between people or web-pages (3). It is inherently a difficult property to quantify in any accurate way, and is why the need to draw a distinction between *trust* and *reputation* arises.

For the purposes of this paper, *reputation* is defined as a numeric measure of a user's *trustworthiness* according to some metric performed on their individual trust scores (1).

There are a number of generic and ad-hoc implementation across the web. Examples of these are Google's PageRank algorithm for measuring a web-page's importance, Stack Exchange's system based on user activity, or eBay's implementation which uses user reviews to calculate reputation. Many of these implementations are bespoke systems, designed especially for their own domain. While these approaches may perform well, the development of a generic reputation algorithm which is applicable in many potential contexts would clearly be beneficial. The development of such a generic reputation algorithm is the focus of this project.

## 2.3 Reputation algorithms

In the following section there is a discussion of a number of ways that reputation can be measured. In these techniques the term *producer* refers to a creator of content and *consumer* to the person who in turn receives that content. The word score refers to a number awarded to the *producers* for that content.

### 2.3.1 TrustRank

TrustRank is a link-analysis algorithm that evolved out of PageRank due to an abuse of the algorithm's weaknesses being exploited to create web-spam. The PageRank algorithm was developed by Page and Brin as they needed an approach to rank search results being presented to users by Google. It works simply by representing the web as a large graph, where web-pages are vertices, and links between them are directed edges between those vertices. PageRank assumes that the more incoming links a page has, the more popular and relevant it must be, and gives each vertex a PageRank score determined by the number of incoming edges it has and the scores of the vertices those edges originated from (9). Figure 2.1 depicts a small network to which the PageRank has been applied. Larger vertex size is indicative of a higher PageRank score.

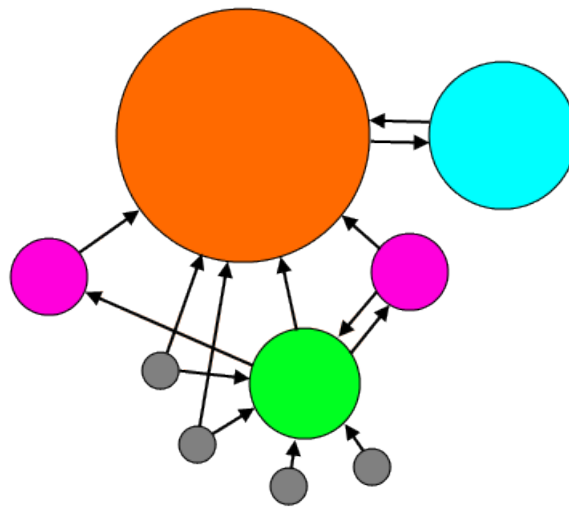


Figure 2.1: A visualisation of PageRank on a very small graph.

The main vulnerability to the PageRank algorithm is that it can not differentiate between a legitimate incoming link and one manufactured to artificially inflate a page's score, which led to huge *link-farms*<sup>1</sup> being created, where people bought and sold links to and from their sites to raise their rankings in search results. To combat this, TrustRank creates a small set of *seed* pages which are manually verified by experts as being legitimate, non-spam web-pages. These seed pages are used to identify incoming links from other pages that are similarly 'good', and this propagates through the graph as it is assumed a 'good' page is unlikely to link to a 'bad' page (6).

Pages can also be seeded negatively in the case that they are found to be 'bad', and distrust can propagate through *outward* links from these pages, as bad pages are likely to link to other similarly bad pages. As figure 2.2 demonstrates, TrustRank has some built in resistance to false-positives and negatives—cases where a 'good' page links to a 'bad' one and vice-versa can be accounted for due to nature of this positive and negative seeding. In this way the effect of manufactured links on page ranking results is significantly reduced.

### 2.3.2 Trust-aware recommenders

Traditionally, recommender systems use one of two approaches (or a hybrid of both) to produce a list of recommendations—content-based filtering, and collaborative filtering(5, 13).

<sup>1</sup><http://www.nytimes.com/2011/02/13/business/13search.html>

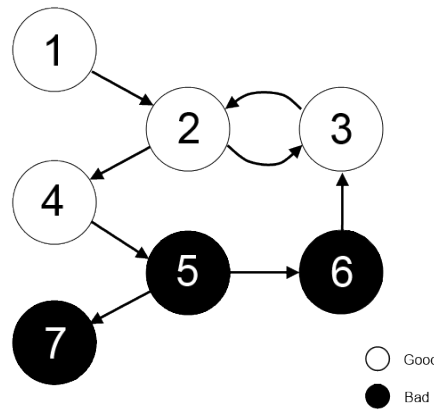


Figure 2.2: A graph of good and bad nodes, taken from (6).

Content-based filtering compares the properties of items to suggest similar ones to the end-user that they might like. This approach is often used to recommend things such as music and films. This can be less helpful in cases where a ‘more like this’ form of recommendation tends to provide stale recommendations with little diversity or variety. Figure 2.3 demonstrates Amazon’s recommendation system recommending batteries to a customer viewing a flash-light.

In collaborative filtering, the recommender system will instead compare the user’s history against that of other users, and make recommendations based on shared viewing, purchase and feedback history (8). These recommendations, when relevant, are useful to both the user and the merchant. The user may be reminded to buy something they had forgotten, and the merchant makes another sale. This information can also be leveraged in other ways, for example a merchant may see that many items are bought together and bundle them at a small discount to encourage further sales.



Figure 2.3: Amazon recommending batteries, among other things, to a customer viewing a flash-light.

Like PageRank, however, these collaborative filtering recommender systems are vulnerable to manipulation. It is often a trivial matter for a person to fabricate a user and artificially create a viewing and feedback history tailored to draw users to their own products. To that end, there has been research into applying reputation to recommender systems (7, 14).

In his research, Massa proposes a reputation model whereby each user explicitly ‘rates’ other users to create tailored trust statements. This system then aggregates all trust into a single trust network that represents all users’ relationships. This trust network can be leveraged to filter out possibly untrustworthy recommendations from reaching the end user (7).

## 2.4 Case studies

This section presents case studies for three bespoke user reputation models. The activity-based Stack Exchange model, eBay's user-feedback centric reputation model and Klout's social network influence score.

### 2.4.1 Stack Exchange – An activity based approach

The Stack Exchange network<sup>2</sup> is a collection of Question and Answer (Q&A) communities, each focused on a specific field of expertise, with a site for everything from programming to bicycling and cooking. The network grew from the original Stack Overflow site, which focused on computer programming questions, and it is currently still the largest site in the network. As of the time of writing this report, the network consists of 114 Q&A sites, almost 4.5 million users and over 8 million questions with 14.6 million answers.



Figure 2.4: A cooking.stackexchange.com profile, showing reputation score.

The Stack Exchange network uses a bespoke model of reputation. A user's reputation is calculated as a sum of points earned by contributing to the sites in various ways (see Figure 2.4 for an example of a Stack Exchange user profile indicating reputation score). Nearly every site activity, from asking and answering questions, to suggesting edits and flagging content for moderation can earn the user reputation points (see Table 2.1 for examples). This system is aimed more towards encouraging site activity than accurately evaluating a user's trust. A user can theoretically gain a considerable amount of reputation without actually demonstrating any domain knowledge.

Site activity	Reputation reward
Your question voted up	+5
Your answer voted up	+10
Your answer 'accepted' as correct	+15
You 'accept' an answer to your own question	+2
Your suggested edit accepted	+2 (up to a total of 1000)

Table 2.1: Stack Exchange reputation rewards.

Additionally, reputation can be lost for a number of reasons, ranging from abuse of the site, to low-quality submissions, although reputation never falls below 1. Moderation is largely performed by the community itself, as increasing reputation scores are rewarded with additional site privileges, ranging from moderation tools to access to chat-rooms.

---

<sup>2</sup><http://stackexchange.com/>

A key limitation in addition to Stack Exchange's reputation scores reflecting activity rather than reputation, is that it is an ad-hoc system, tailored specifically to Stack Exchange's own needs and can not be easily applied to other domains

## 2.4.2 eBay – A feedback-based approach

eBay Inc. is an online auction-house and market-place based entirely around user-to-user transactions. Originally founded in 1995, it has seen steady growth since, and has become the world's largest online marketplace, with over 112 million active users and \$175 billion USD worth of transactions facilitated by the site in 2012 alone<sup>3</sup>.

With such a large volume of transactions passing through the site, evaluating trust is one of eBay's principle concerns. eBay calculates reputation using user feedback after transactions. For each positive transaction, the user's reputation score is increased by one point. If they receive neutral feedback, there is no change to their reputation, and if a transaction receives negative feedback, the user's reputation is decreased by one point.

In the event that there are multiple transactions between users in a single week, eBay aggregates the reputation points the user would have received. If this number is then positive, the reputation score is increased by one, if it is neutral, it does not change, and if it is negative, they lose one reputation point.

As a user's feedback score grows, they gain a coloured star next to their display name as a quick visual indicator of their reputation. Sellers who consistently gain lots of positive feedback and make frequent sales can also gain a 'top-rated seller' badge that informs buyers that they consistently provide good service<sup>4</sup>. An example of a user's feedback profile is given in Figure 2.5.

Additionally, users can leave reviews with their feedback, where they may describe their interactions with the seller, the quality of the product, and the cost and speed of shipping.

This combination of easy to understand reputation score and collection of user reviews allows buyers to quickly find trustworthy sellers and decide from which of them they wish to conduct their business.

However, this approach has its limitations. As a whole, users receive very few negative feedback scores, and there is considerable evidence of a high degree of reciprocation in feedback scores (12).

## 2.4.3 Klout – A measure of influence

Klout is an online service and mobile application that attempts to measure a user's *influence* across social networks. Launched in 2008, it uses analytics from eight different social media websites to evaluate its users' reputation scores, which is referred to as the *Klout Score*. This reputation score is an integer value bounded between 1 and 100, and becomes increasingly difficult to improve upon as it increases<sup>5</sup>.

---

<sup>3</sup>Taken from 'eBay Inc. Reports Strong Fourth Quarter and Full Year 2012 Results'. Available at: <http://investor.ebayinc.com/releasedetail.cfm?ReleaseID=733959>, last accessed March 10, 2014

<sup>4</sup>Taken from 'How feedback works'. Available at <http://pages.ebay.co.uk/help/feedback/howitworks.html>, last accessed 10th February 2014.

<sup>5</sup><http://klout.com/corp/about> and <http://klout.com/corp/score>, last accessed 10th March 2014

## Feedback profile

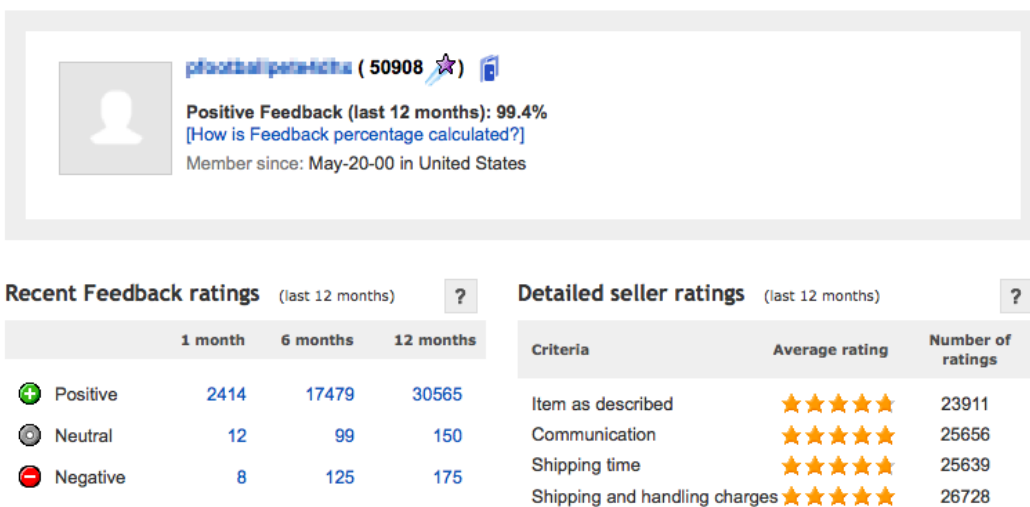


Figure 2.5: An example of eBay feedback for one user.



Figure 2.6: Barack Obama is considerably more influential than I am.

Among the metrics Klout uses to determine reputation are activity from Twitter, Facebook, the user's Wikipedia page (if one exists), Google+ and others. When a user produces content on these sites that gains 'likes', generates discussion or is shared with others, Klout deems that this content generates *influence* to a lesser or greater degree, and so affects their Klout score, an example of which can be seen in Figure 2.6 which compares a typical Klout score with that of Barack Obama.

Although Klout does not specify exactly how it calculates influence, there is evidence that the algorithms used on these *signals* or *collaboration events* are actually quite simple. In June, 2011, a Ph.D candidate in Montana State University, Sean Golliher, managed to calculate Klout's reputation score with an accuracy of 94% simply by calculating a logarithm of the number of a user's followers and retweets<sup>6</sup>.

Besides being an ad-hoc system, the *Klout score* suffers the drawback that a high score is not necessarily indicative of knowledge or reputability in a specific context. A user may develop a high Klout score through a combination of celebrity status and regular use of social media.

<sup>6</sup>Available at: <http://www.seangolliher.com/2011/uncategorized/how-i-reversed-engineered-klout-score-to-an-r2-094/>, last accessed March 12, 2014

## 2.5 Conclusion

This chapter has described some of the methods used to calculate reputation, and how different sites use reputation. It can be seen that many of these reputation models in the wild today are completely proprietary systems that require a lot of specific domain knowledge and are difficult to adapt for other uses. The main goal of this project is to demonstrate that a simplified, generic approach to reputation is an appropriate and viable alternative to engineering an ad-hoc reputation model. This generic approach shall be applicable to any domain in which there are producers and consumers of content.



# Chapter 3: Creating a generic model of user reputation for the social web

This chapter provides an in-depth explanation of how to create a model of user representation for the social web that is generic and can be used in many contexts. The core concepts of this project such as the producer-consumer model of the social web, the collaboration event and collaboration graph, and exactly how to compute reputation are explained in depth.

## 3.1 The producer-consumer model

The producer consumer model is defined as a model of user relationships across an internet site in which there are trust interactions between *producers* and *consumers* of content. Trust interactions may be direct or indirect, and can be explicitly or implicitly defined (15).

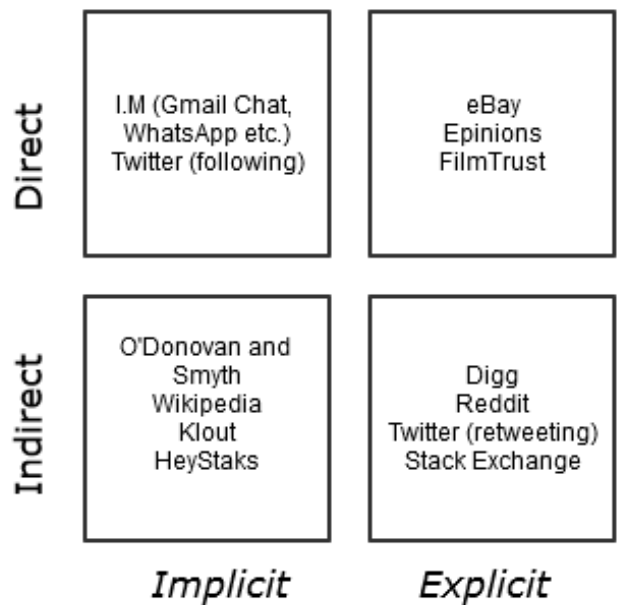


Figure 3.1: Diagram demonstrating examples of direct, indirect, implicit and explicit trust interactions taken from (1).

Direct trust interactions occur directly between two participants, that is to say that one user is rating another user in particular (more specifically a *consumer* is rating a *producer*), whether by following them on twitter or leaving a review on their eBay profile. Conversely, an indirect trust interaction does not occur directly between two users and may instead be a collective or community of *consumers* voting on a single *producer*'s submission as opposed to voting on the *producer* themselves, for example people voting on an article submitted to Digg or Reddit, or multiple people retweeting a tweet. An explicit trust interaction is one in which the measure of trust being given is an explicit quantifiable signal, such as with a Reddit 'up-vote' or an eBay review. Implicit trust interactions, then, are ones in which there

is no explicit, easy to define signal that can be used to indicate the trust ((1), refer to Figure 3.1 for examples).

The generic model of user reputation being investigated in the course of this project will be applicable to any producer-consumer interactions whether direct, indirect, explicit or implicit. Furthermore, this reputation model should be entirely generic, and applicable across a wide range of possible domains by identifying these trust interactions (1).

## 3.2 The collaboration event

Collaboration event is the name given to the flow of trust outlined above. Every collaboration event is a trust interaction between consumer and producer, and in many cases between consumer and multiple producers. For every collaboration event, the producers are awarded with some trust score, which may be positive, negative or neutral. When there is a single producer and consumer, the producer will receive the entirety of the trust score awarded by the consumer. In cases where there are multiple producers, the trust score will be divided among them, with each receiving a trust score corresponding to the perceived value of their contribution as compared to the other producers as seen in Figure 3.2.

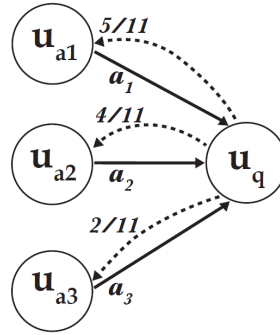


Figure 3.2: A collaboration event with three producers and one consumer taken from McNally et al. 2013.

Identifying collaboration events varies in difficulty. In some instances, such as with an explicit trust interaction (see Figure 3.1) it can be very simple, for example it may just be a unit of positive feedback on an eBay transaction. Other more implicit trust interactions may be more difficult to extract a collaboration event from. It is difficult to determine what some user activities and interactions may mean with regard to trust. A person's contribution to a Wikipedia page being edited may mean that their contribution was of poor quality, or it may simply mean that new information has since arisen that the editor has appended to the Wikipedia page.

To use Figure 3.2 as an example, a consumer on a Stack Exchange site,  $u_q$ , poses a question which receives three answers from the producers  $u_{a1}$ ,  $u_{a2}$  and  $u_{a3}$ . For their answers, these producers receive 5, 4 and 2 stack reputation (i.e. positive votes for their answers) respectively. As a total amount of 11 stack reputation is awarded for this collaboration event,  $u_{a1}$  is rewarded  $\frac{5}{11}$  of the reputation,  $u_{a2}$  receives  $\frac{4}{11}$  and  $u_{a3}$  receives  $\frac{2}{11}$  (1).

### 3.3 Building the collaboration graph

The collaboration events outlined above do not occur in isolation. Active users of these social networks may be involved in many collaboration events as both producer and consumer over time. The accumulation of these collaboration events can be represented by a directed weighted graph, with the users of the network as nodes in the graph, and the collaboration events between users represented by weighted directed edges between the users involved.

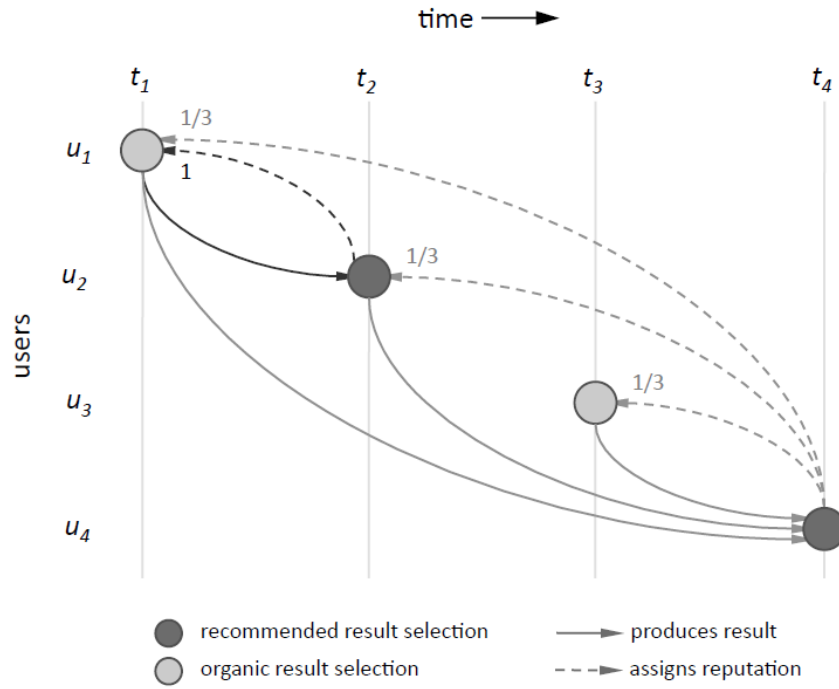


Figure 3.3: Demonstration of a series of collaboration events for social web-search recommendation tool HayStaks as a function of time (11).

To build this graph, all of the collaboration events that make up this network must be examined in order over time. Figure 3.3 demonstrates a small series of collaboration events as a function of time. At time  $t_2$ , the producer  $u_1$  is assigned a unit of reputation by  $u_2$  for their recommendation. At time  $t_4$ , producers  $u_1, u_2$  and  $u_3$  are each assigned a fraction of a reputation unit by  $u_4$ . As the number of collaboration events grows, so too does the graph which also becomes increasingly inter-linked, with the possibility for multiple edges in both directions between the same nodes as shown in Figure 3.4 where node  $u_c$  is both a producer and consumer for nodes  $u_b$  and  $u_d$ .

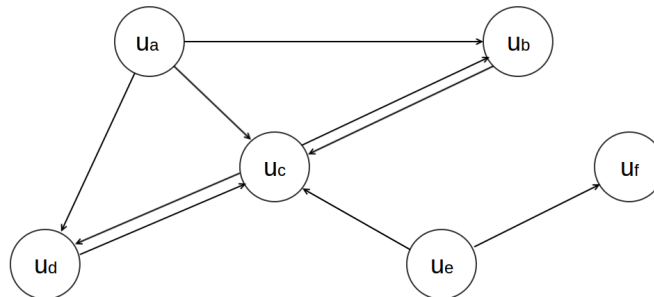


Figure 3.4: Example of a collaboration graph with weights omitted.

## 3.4 Computing reputation

Once a collaboration graph has been constructed, it is then time to compute the reputation scores of the users in the graph. This project examined two algorithms from the literature (1) that can be leveraged to evaluate a reputation score for a user, Google's PageRank (9) and a simple in-degree Weighted Sum technique.

### 3.4.1 Weighted Sum

The Weighted Sum algorithm calculates the reputation score of a node by computing the sum of the weights of all incoming edges for that node. The formula for calculating reputation is as follows:

$$rep(pi) = \sum_{e \in E_{pi}} w_e$$

where  $E_{pi}$  is the set of incoming edges for the producer  $p$ ,  $e$  is the current incoming edge and  $w_e$  is the weight of  $e$ . This Weighted Sum model encourages users to gain reputation by becoming producers of content across many questions and may cause an increase in the number of producers, for example users on the Stack Exchange network may feel more inclined to provide answers for questions. This model may also lead to less repetition of information as a user is less likely to receive as much reputation for simply repeating an answer that has already been given, instead producers are encouraged to produce their own unique content (1).

### 3.4.2 PageRank

PageRank is perhaps one of the most well-known link-analysis algorithms in use today. Originally developed by Sergey Brin and Larry Page at Google (9), it works by making the assumption that important pages in the web (or nodes in a graph) are also the ones most frequently linked to, and that the more important a page or node is, the more weight its own links to other pages should carry (see Figure 2.1 for a visual example). This insight into the nature of the structure of hyper-linked web-pages allowed them to create the PageRank algorithm to sort results for the Google search engine.

The PageRank algorithm is slightly more complex than the Weighted Sum. The PageRank score for each user  $p_i$  is computed as thus:

$$rep(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{|L(p_j)|}$$

where  $d$  is a damping factor,  $N$  is the number of users,  $M(p_i)$  is the set of in-links from consumers to the producer  $p_i$  and  $L(p_j)$  is the set of out-links from  $p_j$  (i.e. the users from which  $p_j$  has consumed content). In the context of the Stack Exchange network, the PageRank score for a given user can be thought of as the likelihood that a randomly selected 'good' answer will have been given by that user (1).

## 3.5 Conclusion

In conclusion, it has been demonstrated that it is possible to create a model of user reputation in a generic fashion when there are consumers and producers of content. With a little thought it is possible define appropriate collaboration events with which a collaboration graph can be built, which can then be leveraged to compute reputation for users using a variety of algorithms..

# Chapter 4: Design & Implementation

---

## 4.1 Project structure

Throughout this project, every attempt has been made to ensure the code written is as concise, clear and efficient as possible. What follows is an attempt to give a brief explanation of the structure of the project and a description of what each class does where appropriate.

The main, root class named 'FYP' begins by finding the user's home directory<sup>1</sup>, and then spawning the 'ParserLauncher' class with the sub-directory of home named 'stack' as a parameter. This ParserLauncher checks the 'stack' directory for any sub-directories.

For each sub-directory found, it is assumed that this directory will contain the input files Users.xml and Posts.xml that need to be parsed, and will launch the appropriate parsers. Once the files have been successfully parsed, the 'ParserLauncher' then creates a 'GraphBuilder' class. Finally, this 'GraphBuilder' class generates the Gephi graphs that will be used later on to compute reputation scores.

A UML class diagram of this project structure can be seen in figure 4.1.

---

<sup>1</sup>In Windows systems this will fall under 'C:\Users\username', on Unix systems it will be '/home/username'

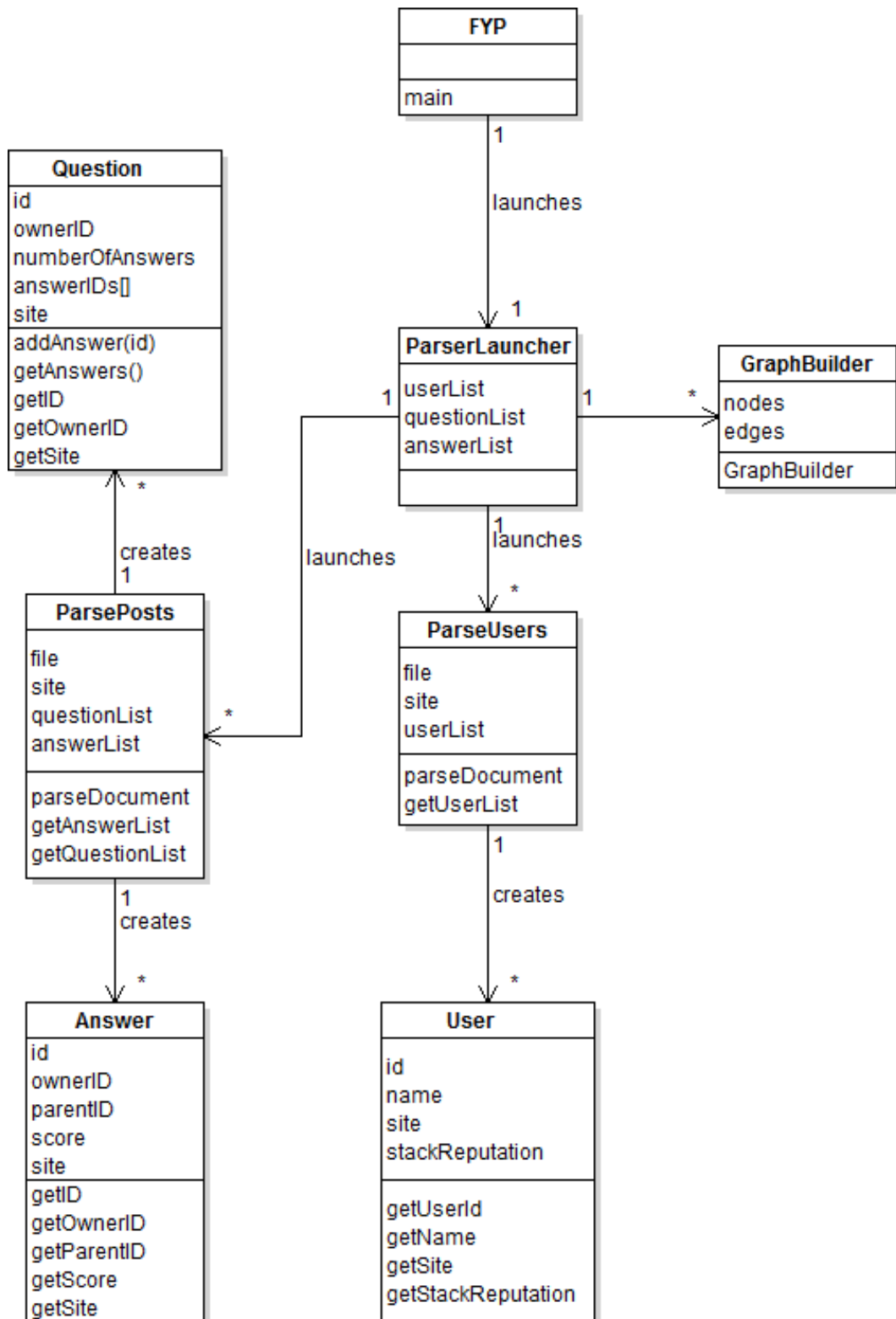


Figure 4.1: Class diagram of the overall project structure.

## 4.2 Development tools

A number of tools were used throughout this project. The majority of the code was written in Java.<sup>2</sup> Apache's Maven was used for dependency management via the m2e plugin bundled with the Eclipse IDE. Initially a PostgreSQL<sup>3</sup> database was used for persistent storage of data, and the Flyway and JDBI<sup>4</sup> Java libraries were used for handling database migrations and executing queries respectively, however the database was abandoned for an in-memory model instead when it became clear the datasets were suitably small.

All code and the contents of this report were kept under version control using git, with a remote on GitHub.com<sup>5</sup> as a backup. In addition, Gephi<sup>6</sup> was used to construct the graph and Matlab was used to perform analysis on the results.

## 4.3 Obtaining the data

The Stack Exchange network publishes quarterly dumps of all user-generated content on the Internet Archive.<sup>7</sup> Data is sanitised to remove private user information, then published in XML format that closely mirrors the tables in a Stack Exchange database, with each XML object corresponding to a row in a conventional relational database. Files can be downloaded directly, or through the BitTorrent protocol (which is encouraged) which also allows a user to seed<sup>8</sup> the data for a while if they wish to contribute back in some way.

Once the data has been downloaded, it must be removed from the archives using the 7zip protocol, and kept somewhere safe. For this project I created a directory named stack in my 'HOME' directory. Each site's data files should be kept in their own sub-directory—my implementation assumes that these sub-directories share the same name as the site they originated from. Of these XML files, the Users and Posts files must be left as is, the others may be removed or left alone—they will be ignored regardless.

The Users and Posts files are then parsed, and the data is stored in one of three HashMaps for users, questions and answers each, with their unique ID numbers used as keys for easy retrieval. The Key-Value access of a HashMap allows convenient and fast lookup of objects by their unique ID number if it is used as a key as retrieving an Object by its key is an  $O(1)$  average-case operation.

### 4.3.1 Processing users

Each User artefact in the Users file contains data corresponding to their unique identifying number, user name, their reputation on this Stack Exchange network and so on. Much of the data provided is extraneous for the purposes of this project and is ignored. When the

---

<sup>2</sup>Oracle's Java 1.7 on Windows machines and OpenJDK 7 on Unix.

<sup>3</sup><http://www.postgresql.org/>

<sup>4</sup><http://flywaydb.org/> and <http://jdbi.org/>

<sup>5</sup><https://github.com/JackBrackeen/FYP>

<sup>6</sup><https://gephi.org/>

<sup>7</sup><http://blog.stackoverflow.com/2014/01/stack-exchange-cc-data-now-hosted-by-the-internet-archive/>

<sup>8</sup>BitTorrent is a peer-to-peer file-sharing protocol. 'Seeding' is its term for remaining connected to the network to continue uploading whatever file(s) you have downloaded for the good of the network.



Users file is parsed, the only data taken directly from it is the user's ID, their user-name and reputation score on that site.

### 4.3.2 Processing posts

As with the Users file, the Posts file contains a large amount of information that is unneeded for this project, such as the number of views a post has and its edit history. For the purposes of this project the only data that is parsed is the post's unique ID, the ID of the author of the post, the post type, the ID of the parent post (if this is an answer) or the number of responses this post has received (if it is a question), and the post's Stack Exchange score.

As we are not told directly whether a post is a question, answer, or something else, we must manually infer this ourselves from the PostTypeId field. Each integer from 1 - 8 corresponds to a post type. Since we are only concerned with questions and answers, any post which does not have a PostTypeId of 1 or 2 is ignored. See Table 4.1 for a full list of post type IDs.

<b>Id</b>	<b>Value</b>
1	Question
2	Answer
3	Wiki
4	TagWikiExcerpt
5	TagWiki
6	ModeratorNomination
7	WikiPlaceholder
8	PrivilegeWiki

Table 4.1: Post type IDs and their meanings.

### 4.3.3 Building the graph

Once the user, question and answer data has been successfully parsed, it is time to create the graph of collaboration events. For this project, the Gephi tool-kit<sup>9</sup> is used to create a graph file which can be interacted with using the Gephi application.

To create this file, first the graph is populated with nodes. This is a simple matter of iterating through the HashMap of users, creating a new node for each user, and appending them to the graph. For the node labels, the user's Stack Exchange reputation score and user-name separated by a comma are used, for easier retrieval later, as Gephi does not provide a facility to insert extra meta-data in nodes. The graph is then populated with the edges that correspond to collaboration events. Every answer is iterated over and used to draw the edges between the answer and its parent question, along with the edge weight, which is the score that the answer received.

Once the nodes and edges have been successfully appended to the graph, creation of the graph file is accomplished using the tool-kit's ExportController class. This class can create either a PDF document with a visualization of the graph or a Gephi graph file with the .gexf extension. Which type of file is created simply depends on the file extension used when writing. The implementation of this project creates a separate graph file for each network examined.

---

<sup>9</sup>The tool-kit is a Java library which allows developers to create and interact with Gephi graphs through its API: <https://gephi.org/toolkit/>

## 4.4 Computing reputation scores

Gephi makes it a simple matter to compute the In-degree Weighted Sum and PageRank scores for the nodes in our graph, among other things. Open the generated graph file in Gephi. In the right-hand pane of the overview tab, there is a statistics view, which can be used to compute statistical data for the nodes and edges of the graph by clicking the corresponding ‘run’ button. Once this is done, a report view is displayed which can be ignored.

Id	Label	Weighted In-Degree	PageRank
1584	29588, Daniel R Hicks	1,740	0.017
76	11774, freiheit	1,017	0.004
1259	17885, zenbike	893	0.012
2355	12804, heltonbiker	777	0.009
-1	1, Community	748	0.004
8	11267, Neil Fein	682	0.003
1487	10902, мауїтew	623	0.005
425	8232, Unsliced	591	0.002

Figure 4.2: Gephi data-lab showing users on the bicycles stack exchange network sorted by Weighted Sum score.

Once the Weighted Sum and PageRank values have been computed, they are inserted into the ‘Data Laboratory’ view. From this view it is possible to interact with the data in a number of ways, from searching for specific nodes, adding, deleting and duplicating nodes or edges, sorting by property (see Figure 4.2), and changing the properties of nodes. It is also possible to export this data to a comma-separated-value file for later analysis.

# Chapter 5: Testing & Evaluation

---

This chapter is a presentation of the results of this project, with a description of testing methodology, and an explanation of the findings.

## 5.1 The datasets

For the purposes of this project, three different datasets were selected from the Stack Exchange network. These datasets were the bicycles, cooking (Seasoned Advice or SA) and electronic engineering (EE) networks, and were primarily chosen to attempt to create a somewhat complete spread of the types of questions that may be asked on a social Q&A site. More specifically, the questions asked on SA may be of a more qualitative type, where the answers can vary wildly according to the outlook of the person answering, where as the questions on the EE network are likely to have more concrete ‘correct’ and ‘incorrect’ answers, although of course a large number of questions may still be entirely qualitative. The bicycles site was chosen for questions that are largely a mix between qualitative and ‘right or wrong’ questions.

	Dataset		
	Bicycles	SA	EE
# Users	8,009	15,802	25,335
# Questions asked	4,023	9,037	24,404
# Answers given	12,196	24,679	49,006
# Answers on votes	62,974	130,446	210,694

Table 5.1: Statistics for the three Stack Exchange sites used in the course of this project—the bicycles, SA and EE networks.

Table 5.1 gives some exact figures to give an idea of the scale of the sites chosen. The Bicycles site was the smallest of the three and EE was the largest, while SA fell quite close to the median between them. The bicycles site had the most average number of answers per question at 3.03 answers, while the EE site had the least with 2.01 answers per question on average. Conversely the EE site had the most answers per user at 1.93, while the bicycles site had the least at 1.52 answers per user. In both instances the SA site fell somewhat in the middle, however SA had the most average votes per answer at 5.29, while EE and bicycles had 4.30 and 5.16 respectively.

## 5.2 Methodology

Once reputation figures have been calculated as outline in section 4.4 and the results have been exported to comma-separated-value format, it is then time perform analysis of the data. To determine the accuracy of results, histograms of distribution of the different reputation

scores were generated with Matlab, and in the case of the SA and EE datasets were compared to the findings in the work by McNally et al. It is important to note that the range of values for each set of reputation scores was normalised between 0 and 1 as follows, where  $v$  is the vector of values,  $n$  is the vector of normalised values and  $\max(v)$  is the largest value in  $v$ :

$$n = \frac{v}{\max(v)}$$

Once the accuracy of the reputation scores has been determined to be in line with the results in the literature as outlined above, a correlation analysis is then performed on the scores. Correlation was calculated between each combination of scores, i.e., between SE and WS, SE and PR and WS and PR scores. For each dataset, this correlation was calculated on a series of ‘buckets’ of data, i.e. the top 10, 50, 100, 200, 500, 1000 and 1500 users by WS score.

## 5.3 Analysis

This section presents the results of the experiments carried out for this project and a description and analysis of those results. Comparisons are drawn between the performance of the generic models for reputation with Stack Exchange’s own ad-hoc system.

### 5.3.1 Evaluating performance

The distribution of reputation scores for the three reputation models—SE, WS and PR—were plotted on a line-plot using Matlab, and are shown below. Figure 5.1 displays the distribution of scores across the bicycles network, Figure 5.2 displays the reputation distribution across the SA network and Figure 5.3 is the distribution of reputation across the EE network. The x-axis on each plot indicates number of users, while the y-axis is the normalised reputation score of those users. Line thickness was adjusted for maximum visibility of the plots and is not indicative of anything.

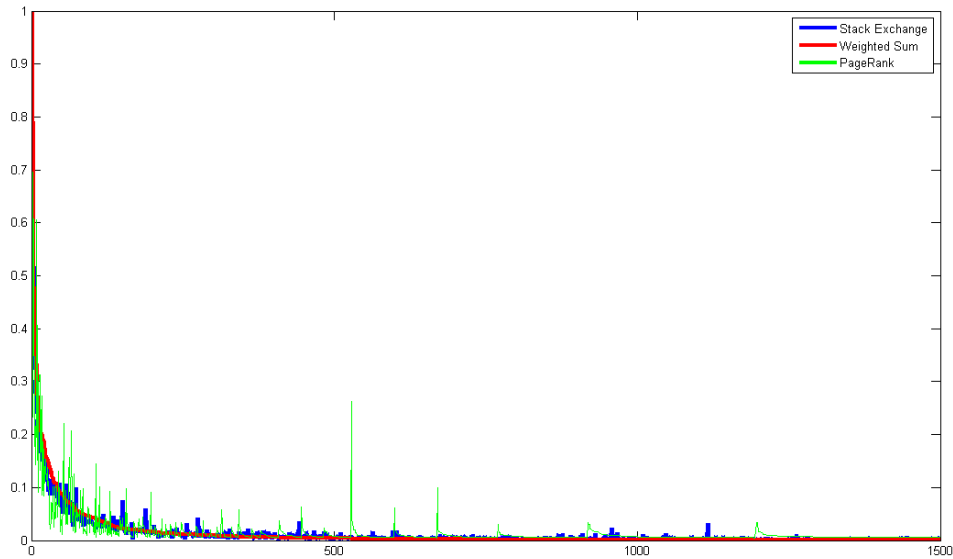


Figure 5.1: Distribution of reputation scores across the Bicycles SE network.

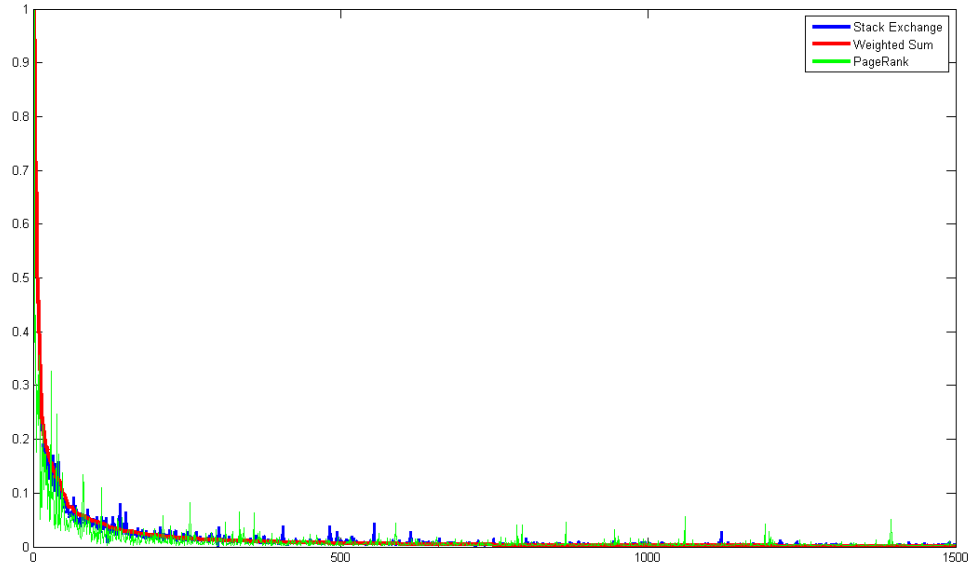


Figure 5.2: Distribution of reputation scores across the Seasoned Advice SE network.

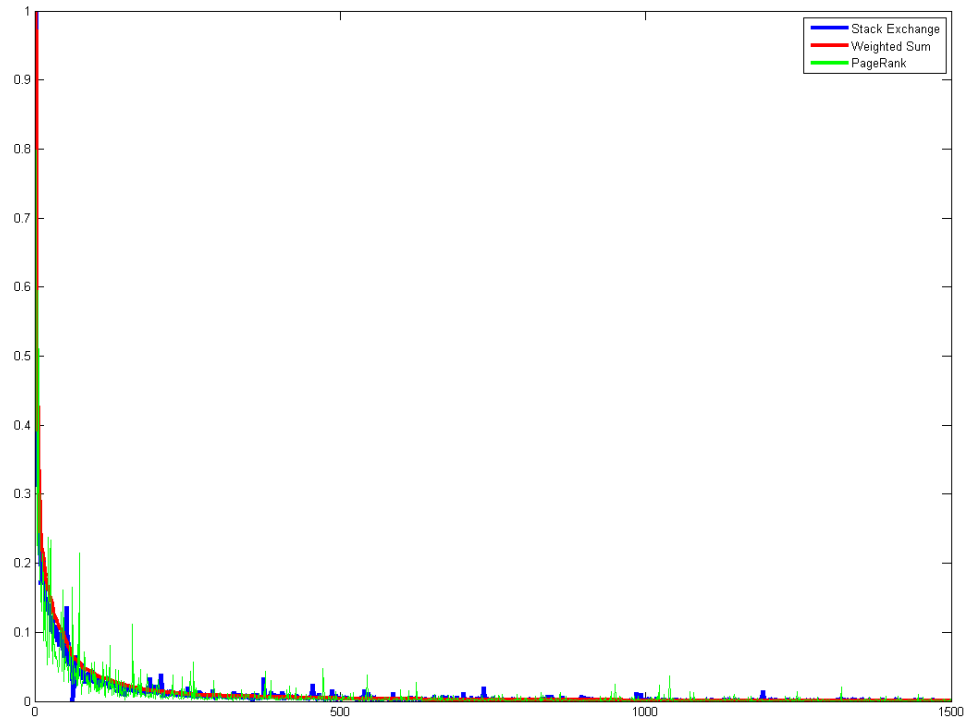


Figure 5.3: Distribution of reputation scores across the Electronics SE network.

As can be seen in Figures 5.1, 5.2 and 5.3, the normalised reputation scores follows a long-tailed distribution in line with the experiments performed by McNally et al. (it is important to note that the axis plotting number of users in the histograms in the literature is set to a logarithmic scale, which accounts for the slight difference with the data plotted here). That is, a very small number of users have extremely high scores for each reputation model, while the vast majority have middling or lower scores.

In conclusion the generic algorithms for reputation being investigated have a very similar spread of distribution of scores as the ad-hoc Stack Exchange model, which is indicative of a level of accuracy in these scores.

### 5.3.2 Correlation analysis

Correlation analysis was performed only on sets of users for whom WS and PR scores could be calculated. For each ‘bucket’ of users being evaluated, both Pearson and Spearman correlation was calculated between each combination of reputation algorithm. For the purposes of this analysis, the Spearman correlation is considered the more accurate of the two. The Pearson correlation algorithm makes the assumption that the two datasets are linked by some linear relationship, whereas the Spearman rank accounts for the fact that this may not be true. Pearson was calculated as it cost little extra effort and out of interest in the results Pearson would return.

Both the Spearman and Pearson correlation coefficients produce a decimal number between  $-1$ , which indicates a total negative correlation and  $+1$ , which indicates a total positive correlation (a score of 0 indicates absolutely no correlation between the datasets whatsoever). Ideally both the WS and PR algorithms should correlate very highly with the SE model (scores of 0.90 and above, for example). This would indicate that they have a very similar performance to the SE model and are a suitable alternative. In reality it is expected that the WS algorithm correlates very highly with the SE model and the PR algorithm will correlate moderately well with SE.

Site	Users	Pearson			Spearman		
Correlations	between	SE/WS	SE/PR	WS/PR	SE/WS	SE/PR	WS/PR
Bicycles	Top 10	0.9523	0.8225	0.7222	0.9515	0.5636	0.4182
	Top 50	0.9791	0.8846	0.8501	0.9520	0.8023	0.8082
	Top 100	0.9815	0.8808	0.8504	0.9555	0.7029	0.6864
	Top 200	0.9822	0.8915	0.8668	0.9364	0.7240	0.6980
	Top 500	0.9833	0.9027	0.8819	0.8798	0.6606	0.6467
	Top 1000	0.9839	0.8950	0.8759	0.8394	0.5220	0.5155
	Top 1500	0.9839	0.8965	0.8782	0.8293	0.4969	0.5083
Cooking	Top 10	0.9656	0.6547	0.5103	0.9394	0.3455	0.4545
	Top 50	0.9901	0.7855	0.7536	0.9501	0.6267	0.6700
	Top 100	0.9920	0.8292	0.8058	0.9571	0.7452	0.7469
	Top 200	0.9921	0.8551	0.8375	0.9468	0.8017	0.8094
	Top 500	0.9921	0.8703	0.8564	0.9036	0.7176	0.7259
	Top 1000	0.9921	0.8763	0.8640	0.8792	0.6651	0.6775
	Top 1500	0.9921	0.8777	0.8660	0.8325	0.6304	0.6476
Electronics	Top 10	0.9881	0.7840	0.8204	0.9515	0.7212	0.8545
	Top 50	0.9922	0.8716	0.8948	0.9669	0.6926	0.7247
	Top 100	0.9902	0.8781	0.9008	0.9545	0.7570	0.8011
	Top 200	0.9900	0.8925	0.9115	0.9245	0.7802	0.8206
	Top 500	0.9893	0.9038	0.9212	0.9009	0.7634	0.7918
	Top 1000	0.9893	0.9085	0.9251	0.8812	0.7446	0.7654
	Top 1500	0.9893	0.9099	0.9262	0.8613	0.7035	0.7269

Table 5.2: Pearson and Spearman correlations between the SE, WS and PR reputation models for the three datasets.

As shown in Table 5.2, the WS algorithm correlates very highly with the SE model across all three datasets. There is a tendency for this good performance to trail off slightly, as seen when the larger buckets are examined. The lowest Spearman correlation between the SE and WS algorithms occurs in the largest bucket in the bicycles site (a correlation of 0.8293 is still very high, however), but is in line with the performance of the algorithm across the other datasets.

When SE is correlated with the PR scores, it is immediately apparent that there is a much

lower correlation between them than between SE and WS, with the highest Spearman correlation (0.8023) falling slightly short of WS's worst case, and its worst case being a very low Spearman score of 0.3455 in the SA dataset. Interestingly, this worst-case performance occurs in the 'Top 10' bucket, and when the Spearman correlation between PR and SE is examined in detail, the 'Top 10' bucket also has a very low score in the bicycles dataset. Upon further examination it seems that PR begins to perform better when a larger number of users are examined, before slightly worsening again.

### 5.3.3 Conclusion

In conclusion it appears that, as expected, the WS algorithm correlates very well SE's ad-hoc reputation model, but suffers a very minor drop in correlation as more low activity users are considered. The PR algorithm correlates very poorly with the other models when only looking at the very highest ranked users, but sees a rapid increase in correlation when lower reputation users are considered, most likely due to PR's propensity to assign low default reputation scores to users with activity.

## 5.4 Coverage

It is important that the reputation algorithms can be applied to as many users in a network as possible, and as such, the coverage of each algorithm on the datasets is outlined below (exact numbers can be seen in Table 5.3). As the reputations evaluated in this investigation can only be applied to users who are producers of content (i.e. they have answered at least one question), they are bound to have a much lower coverage than SE's model of reputation, which takes all user activity into account, and gives all users a default reputation of 1 upon registering.

Site	%WS coverage	%PR coverage
Bicycles	19.60%	24.6%
Cooking	20.61%	30.1%
Electronics	12.59%	17.5%

Table 5.3: Coverage figures of users with computable WS and PR scores.

As can be seen in Figures 5.4, 5.5 and 5.6 which follow, the WS and PR reputation algorithms have significantly less user coverage than the SE model, which always has 100% coverage. The PR algorithm always has better coverage than the WS algorithm and this is due to the fact that PR will often assign a low default value as explained previously. McNally et al. came to the conclusion that these extra users covered by the PR algorithm were likely consumers of content at some point and a similar conclusion was drawn for this project.

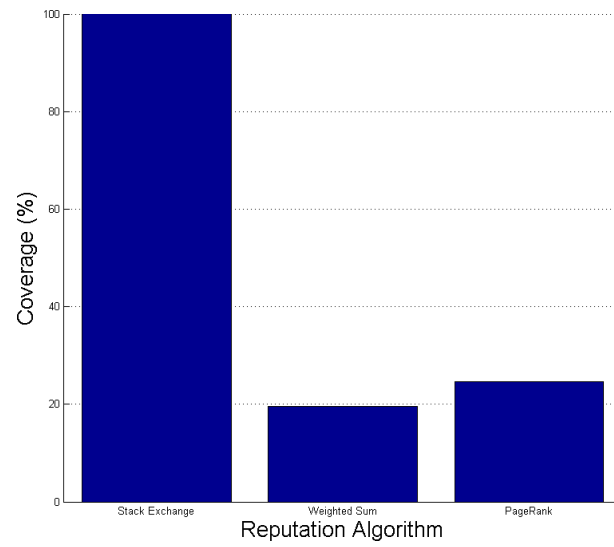


Figure 5.4: Bar chart visualising coverage provided by reputation algorithms on the bicycles network.

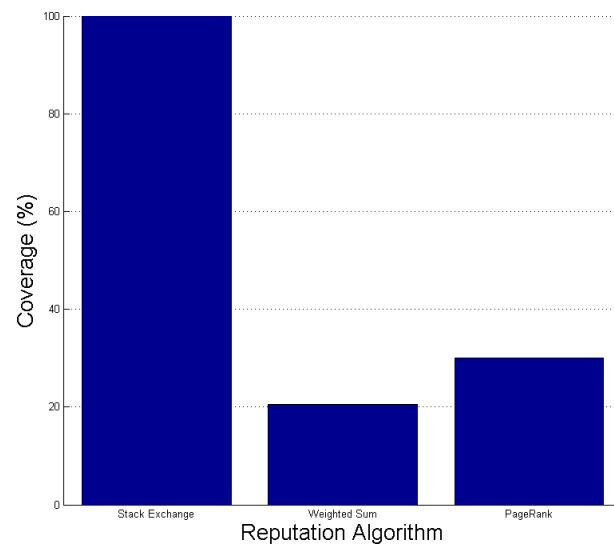


Figure 5.5: Bar chart visualising coverage provided by reputation algorithms on the SA network.



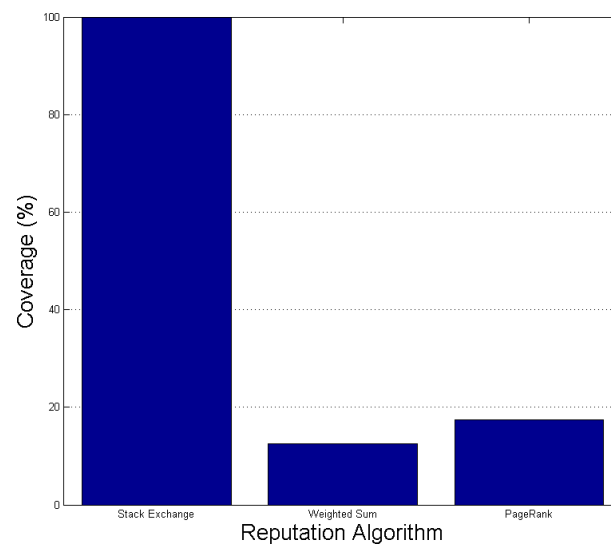


Figure 5.6: Bar chart visualising coverage provided by reputation algorithms on the EE network.

# Chapter 6: Conclusions & Future Work

---

The scope of this project covered both the development of a way to extract the Stack Exchange data, and an academic investigation and analysis of the generic user reputation models proposed in the literature. This chapter discusses exactly which goals were met and to what success, draws conclusions about the subject area, evaluates and makes criticism of the overall implementation of the project and outlines possible future areas of research into the field.

## 6.1 Fulfilment of project goals

### 6.1.1 Mandatory goals

The stated mandatory goals of this project were to:

- Download Stack Exchange data (<http://data.stackexchange.com/>) - data from three sites should be obtained.
- Implementation of reputation algorithms from the literature.
- Evaluation: for each dataset, compare the performance of these algorithms to the user reputation model currently used on Stack Exchange.

All mandatory goals of the project were met to satisfaction.

### 6.1.2 Discretionary goals

The stated discretionary goals of this project were to:

- Predict the correct answers to questions based on user reputation.
- Evaluate the accuracy of predicted answers.
- Perform user-trials and correlate prediction performance with offline metrics.

While none of these goals were successfully met, considerable thought was put into a method for answer prediction and evaluation of prediction accuracy. The idea being that a collaboration graph is constructed using the first 80% of the collaboration events (questions) in the network. This graph is then used as training data, in which users' WS and PR reputation scores are calculated. For the remaining 20% of the collaboration events, each answer is given a ranking for each of their respective owner's reputation scores, i.e. the user with the highest score for a particular reputation algorithm is given the ranking of 1 for that particular algorithm as demonstrated in Table 6.1.

ID	Score	True rank	SE	WS	PR
A <sub>1</sub>	+20	1	2	1	3
A <sub>2</sub>	+10	2	1	3	1
A <sub>3</sub>	+5	3	4	4	4
A <sub>4</sub>	−2	4	3	2	2

Table 6.1: Example attempt at predicting a correct answer to a question by assigning rankings to each answer according to the author’s reputation scores for each of the reputation algorithms and comparing them to the actual ranking of answers according to their scores.

These rankings can then be correlated against the actual answer ranking (that is, the ranking of answers by points awarded). Once the final remaining 20% of the collaboration events have been tested in such a way, the mean of the Spearman ranks for each algorithm can be compared to find which is most likely to predict a correct answer. This approach can also be modified to consider answers that are ‘Accepted’ by the consumer as correct to be ranked ahead of the highest scoring answer in cases where it is not already.

### 6.1.3 Exceptional goals

The stated exceptional goals of this project were to:

- Propose and implement enhancements to improve algorithm performance.
- Analyse the robustness of the reputation algorithms against attack.

None of these were attempted due to time constraints.

## 6.2 Conclusions

Through examination of the results of this project it was found that the generic Weighted Sum approach to user reputation is superior to Rank, which falls in line with expected results according to the research undertaken. WS was far superior to PR for the datasets used, but there is evidence that PR may have superior performance to WS given a large enough dataset with a very large percentage of low-activity users.

In addition, there is considerable evidence that the WS reputation model performs nearly identically to the SE model in terms of distribution of reputation scores, and after performing correlation analysis, it has been demonstrated that the WS model for user reputation is a viable alternative to the SE model of reputation with regards to a pure reputation model. This WS model also has the benefit that it is an entirely generic approach and can be used for any domain in which there is a *producer-consumer* model from which *collaboration events* can be extracted.

## 6.3 Learning outcome

Much was learned throughout the course of this project. In particular, the process of realising an academic project from conception to completion and of the difficulties in developing a piece of non-trivial software in terms of managing both time and complexity of said software. In addition, learning to use and integrate different tools and libraries into the project was enlightening, even in cases where these tools were unused in the end.

## 6.4 Failings

Many aspects of this project could have been improved upon, in particular the implementation of the project and the choice of tools used. If a more complete understanding of exactly what the project entailed had been reached before beginning implementation, a lot of dead ends and wasted time could have been avoided. If this project were to be attempted again, an alternative to Gephi may be found—while Gephi is a fantastic piece of software, it is still in public beta and as such suffers from some instability. Additionally many of Gephi’s features were unneeded for the scope of this project.

## 6.5 Future work

This section will give very brief descriptions of possible future paths this work can be taken down.

1. A continued investigation into generic algorithms to compute user reputation can be undertaken, with attention paid to other link-analysis algorithms such as the HITS algorithm (Hubs & Authorities).
2. An investigation into these algorithms’ resistance to abuse or tampering may be undertaken. For example, it has already been demonstrated PageRank is vulnerable to exploitation with the use of artificial in-links. Perform experiments to determine if the Weighted Sum and PageRank algorithms can be exploited in the context of a collaboration graph.
3. Attempt to partition expert users in a collaboration graph by field of expertise. Stack Exchange offers ‘Tag’ meta-data for posts in its Posts.xml file. Leverage this information to find the most reputable users in the contexts of different fields of expertise. For example, the most reputable cake-baker on Seasoned Advice may not be as reputable when it comes to fry-cooking.
4. Evaluate whether using the most reputable users according to these generic reputation algorithms is a viable way to seed the TrustRank algorithm with ‘good’ users.
5. Combine the previous two examples to provide an ‘even spread’ of seeds for the TrustRank algorithm.

- (1) McNally, K., O'Mahony, M.P., and Smyth, B. 2013 – "A Model of Collaboration-based Reputation for the Social Web." In Seventh International AAAI Conference of Weblogs and Social Media, 2013
- (2) Cheng, R., and Vassileva, J. 2005 – "Reward Mechanism for Sustainable Online Learning Community." Proceedings of the 2005 conference on Artificial Intelligence in Education. IOS Press.
- (3) Page, L., Brin S., Motwani, R., and Winograd, T. 1999 – "The PageRank citation ranking: Bringing order to the Web."
- (4) Mui, L., Mohtashemi, M., Halberstadt, A. 2002 – "A Computational Model of Trust and Reputation." In System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference, IEEE, 2002 p2431-2439.
- (5) Resnick, P., Zeckhauser, R. 2002 – "Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System." Advances in Applied Microeconomics 11, 2002, p127-157
- (6) Gyongyi, Z., Garcia-Molina, H., Pedersen, J. 2004 – "Combating Web Spam with TrustRank." Proceedings of the Thirtieth International Conference on Very Large Data Bases (2004), Volume 30, p576-587
- (7) Massa, P., Avesani, P. 2007 – "Trust-Aware Recommender Systems." In Proceedings of the 2007 ACM conference on Recommender systems. ACM. p17-24
- (8) Breese, J.S., Heckerman, D., Kadie, C. 1988 – "Empirical Analysis of Predictive Algorithms for Collaborative Filtering." In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., May 1998, p43-52.
- (9) Brin, S., Page, L. 1998 – "The anatomy of a large-scale hypertextual Web search engine." Computer networks and ISDN systems 30.1, 1998, p107-117.
- (10) Hoorens, V. 1993 – "Self-enhancement and Superiority Biases in Social Comparison." The European Review of Social Psychology (1993), Volume 4, Issue 1 p113-139
- (11) McNally, K., O'Mahony, M., Coyle, M., Briggs, P., Smyth, B. 2010 – "A Case Study of Collaboration and Reputation in Social Web Search." In ACM Transactions on Intelligent Systems and Technology, Volume 9, No. 4, Article 39, March 2010, p39-69
- (12) Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J. 1994 – "Grouplens: An Open Architecture for Collaborative Filtering of Netnews." In Proceedings of ACM CSCW94 Conference on Computer-Supported Cooperative Work, Sharing Information and Creating Meaning, 1994, p175-186
- (13) Balabanovic, M., Shoham, Y. 1997 – "Fab: Contentbased, Collaborative Recommendation." Communications of the ACM , Vol. 40, No. 3, ACM Press, March, 1997, p6672
- (14) O'Mahony, M., Hurley, N., Kushmerick, N., Silvestre, G. 2004 – "Collaborative Recommendation: A Robustness Analysis." ACM Transactions on Internet Technology (TOIT), Special Issue on Machine Learning for the Internet, Vol. 4, No. 4, ACM Press, November, 2004, p344377
- (15) O'Donovan, J., Smyth, B. 2005 – "Trust in Recommender Systems". In Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI '05). ACM, New York, NY, USA, 2005, p167-174