Decentralized Information Search

by

Neelakantan Swaminathan

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved June 2013 by the
Graduate Supervisory Committee:

Hari Sundaram, Co-Chair
Hasan Davulcu, Co-Chair
Pavan Turaga

ARIZONA STATE UNIVERSITY

August 2013

ABSTRACT

Our research focuses on finding answers through de-centralized search, for complex, imprecise queries (such as "Which is the best hair salon nearby?") in situations where there is a spatiotemporal constraint (say answer needs to be found within 15 minutes) associated with the query. In general, human networks are good in answering imprecise queries. We try to use the social network of a person to answer his query. Our research aims at designing a framework that exploits the user's social network in order to maximize the answers for a given query. Exploiting an user's social network has several challenges. The major challenge is that the user's immediate social circle may not possess the answer for the given query, and hence the framework designed needs to carry out the query diffusion process across the network. The next challenge involves in finding the right set of seeds to pass the query to in the user's social circle. One other challenge is to incentivize people in the social network to respond to the query and thereby maximize the quality and quantity of replies. Our proposed framework is a mobile application where an individual can either respond to the query or forward it to his friends.

We simulated the query diffusion process in three types of graphs: Small World, Random and Preferential Attachment. Given a type of network and a particular query, we carried out the query diffusion by selecting seeds based on attributes of the seed. The main attributes are Topic relevance, Replying or Forwarding probability and Time to Respond. We found that there is a considerable increase in the number of replies attained, even without saturating the user's network, if we adopt an optimal seed selection process. We found the output of the optimal algorithm to be satisfactory as the number of replies received at the interrogator's end was close to three times the number of neighbours an interrogator has. We addressed the challenge of incentivizing people to respond by associating a particular amount of points for each query asked, and awarding the same to people involved in answering the query. Thus, we aim to design a mobile application based on our proposed framework so that it helps in maximizing the replies for the interrogator's query by diffusing the query across his/her social network.

DEDICATION

I dedicate my thesis to all researchers round the globe, who toil hard to produce quality results.

ACKNOWLEDGMENTS

I am indebted to my thesis advisor, Dr. Hari Sundaram, for his inspiring thoughts, constant guidance and moral support during my MS study. His approach to problem formulation and presentation of the same is a quality, which I try to emulate and has totally transformed my attitude towards research.I would like to thank my committee members, Professor Hasan Davulcu and Professor Pavan Turaga for serving as members of my thesis committee and examining my thesis report. This thesis would not have been possible without the support of my entire family and friends. I wish to thank my family for their endless love and unconditional support.

TABLE OF CONTENTS

Page

CHAPTER

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

Our research focuses on finding answers for queries that have a spatiotemporal constraint associated with it. Most of these queries are qualitative, imprecise and are valid only for a short term. People often address these queries to friends or other people in their social network, and hence a preference similarity and trust factor that comes into play. We explore various ways of utilizing an individual's social network, to find people who might help in finding answers to such queries. Our main objective is to maximize the number of replies attained at the interrogator's (individual who asks a query) end, when he/she asks a question. Hence, we aim to design a framework that helps us in maximizing the replies for the user's query by diffusing the query across his social network. Our framework carries out the query diffusion process using an underlying seed selection algorithm. We did the query diffusion process as a simulation of the framework designed. In this chapter, we will first discuss the graph models that we used for our simulation and the evaluation metrics for the algorithms that we designed. We then discuss our approach to solve the research problem in the later part of the chapter. We conclude the chapter by providing an overview of the whole thesis and the way it is organized.

1.1 Types of Graphs and Metrics for Measurement

In order to simulate the query diffusion process, we need to have graph models that possess similar characteristics as a typical social network. In this section, we discuss the graph models that we use for our simulation and the performance measurement metrics that we analyze when carrying out a diffusion process.

We used three graph models in our research that simulates a real world social network. The three graph models are Watts-Strogatz (Small World) [23], Barabasi (Preferential Attachment) [18] and Random Graphs model [22]. We explain the construction of a typical graph model in Chapter 2. Given a type of graph model, we use the undirected graph $G = (E,V)$ (where "E" is the total number of edges and "V" is the total number of vertices) to carry out the query diffusion process based on the seed selection algorithm. We discuss more on the seed selection algorithms that we designed and its implementation in Chapter 2. For a given graph model, once we carry out the simulation by diffusing the query, we analyze the performance of the underlying seed selection algorithm using three factors. The three factors that we analyze based on the output are Cost with respect to number of activated vertices (that received the query), Cost with respect to number of activated edges (that led to query propagation) and number of replies. We discuss more on the evaluation metrics in Chapter 2.

We tried out different seed selection algorithms that govern the query diffusion process in the graph models discussed in the section. Given a graph model, the performance of an algorithm is high when we can maximize the number of replies with minimal cost using the algorithm. In the next section, we discuss the different algorithms that we tried to improve the performance.

1.2 Proposed Approach

We discussed graph models that we used and an overview of the evaluation metrics in the previous section. In this section, we explain the query diffusion process in a typical social network and the basis on which the seed selection algorithms works. We also discuss the main results of our research.

Figure 1.1 depicts the query diffusion in a social network of a user. Given a social network with an interrogator, Figure 1.1 shows the overview of the query diffusion process into the social network of the interrogator.

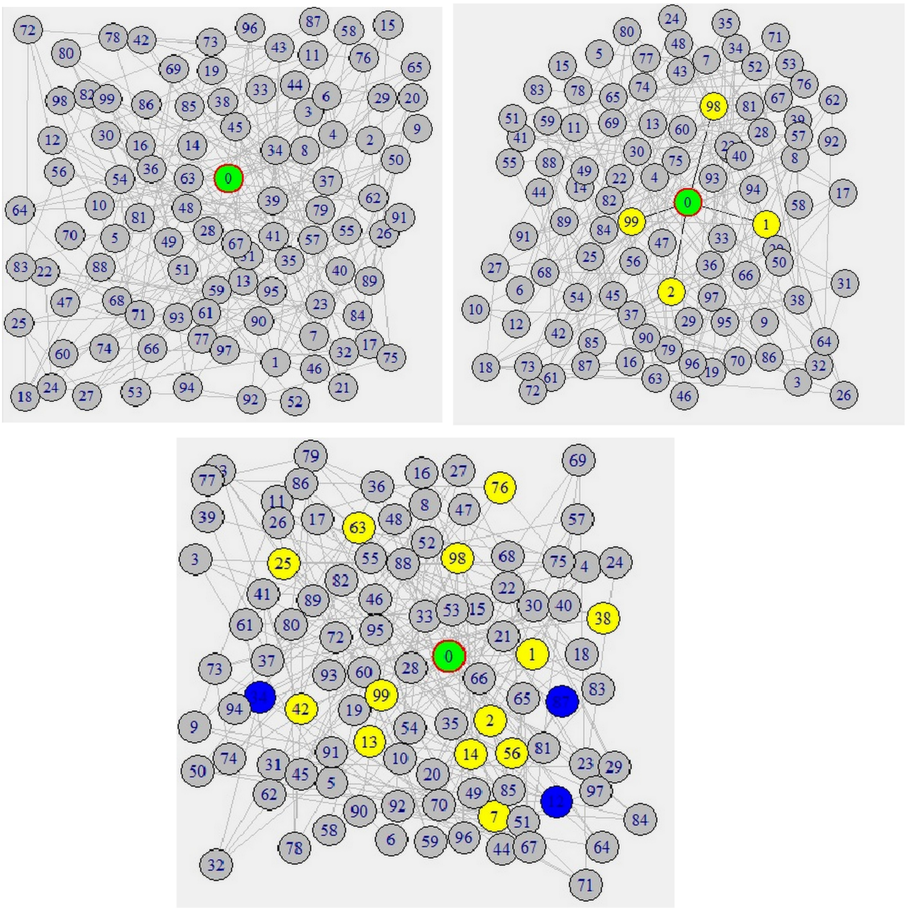


Figure 1.1: The figure shows a typical query diffusion process in a social network of a user. The above figure corresponds to the social network. Consider an interrogator to be a node in the graph as shown in the top left figure (node zero in green color). The interrogator asks the query to few people in his neighbourhood, as shown in the top right figure (nodes in yellow are the one that received the query in the interrogator's neighbourhood). We diffuse the query across the network and in the end there are nodes that answers the query (nodes in blue are the one that reply to the query) as shown in the figure in the bottom.

Once we decide to carry out the query diffusion across the network, we started to fix attributes for each node (that would represent people) in the given network. Figure 1.2 shows the user activity model. It shows the two types of users and the corresponding actions that they could perform.

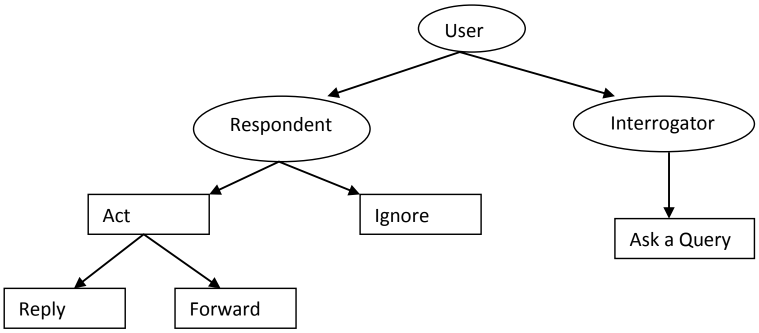


Figure 1.2: The figure shows the User activity model. There are two types of users of the system: Interrogator and Respondent. Interrogator asks the query. Respondent receives that query. The model shows the various actions that respondent can perform. The respondent can choose to act or ignore, when he receives the query. If a respondent chooses to act, he can reply to the interrogator or forward to his neighbours.

Figure1.2 shows the attributes (that has associated values) that helps in differentiating nodes, so that we can select the best subset of nodes, to which we can pass the query. We selected this subset of nodes at each level of the diffusion process, starting from the immediate social circle of the user (who asks the query).

1.3 Main Contributions

We tried out several algorithms for seed selection in order to compute, compare and contrast the different metrics of measurement that accounts for the success of the overall goal. The major algorithms that we analyzed were Seed selection based on sum of all attribute values or each of the individual attributes, Diameter and Cluster Coefficient comparison of a typical network, Giant Component discovery in graph, Nodes picked based on summation of Topic Value and Acting Probability (finalized optimal selection process) and Money Split Algorithm (for incentivizing users to respond). We have explained each of the algorithm mentioned above and their results in Chapter 2 on the following page, Chapter 3 on page 24 and Chapter 4 on page 34. One of the major outcomes of these simulations is that, when we carried out the query diffusion process using the optimal selection algorithm, there was a tremendous increase in the replies received at the user's end. We received as many as 13 replies for a given interrogator's query in a typical graph model with the average number of neighbours of the interrogator being as low as four. The number of replies is almost three times the friends that the user has in his/her immediate social circle. We will discuss more about the findings in the later part of the thesis.

We organize the rest of the thesis into four chapters. Chapter 2 on the next page, presents the work done on attribute setting and initial set of seed selection algorithms. In Chapter 3 on page 24, we discuss the Giant component of a graph and the seed selection algorithm involving appropriate combination of node attributes to maximize replies without saturating the network. Chapter 4 on page 34, provides a discussion on incentivizing users to participate, as well as the mobile application that is under development using the research done. We conclude the thesis by explaining the overall work of our research project and by mentioning the scope for future research work in the final chapter.

Chapter 2

QUERY DIFFUSION AND REPLIES MAXIMIZATION

Our research lies in the intersection of two broad research areas: Information search and Social networks. We focus on designing a framework that explores ways to retrieve the information required by individuals using their social network. We seek answers to qualitative and imprecise questions that individuals ask their friends rather than searching answers online. Hence, we need to maximize the number of responses at the interrogator's end, so that he/she gets to decide a suitable answer from the set of responses received. If we have to utilize the interrogator's social network to find the required information, we need to diffuse the query across their social network. One of the challenges associated with the query diffusion process is preventing the network from being saturated. While trying to maximize responses, we should not end up saturating the interrogator's network by asking too many people. Hence, we aim to diffuse the query in a way that does not saturate the network but reaches specific seeds (respondents) that have a higher probability of knowing the answer and replying. Our proposed framework needs to have an underlying algorithm to govern query diffusion across the interrogator's network. We design algorithms that can pick relevant seeds in the interrogator's social circle and pass the query to them. This chapter focuses on seed selection algorithms that pick seeds efficiently based on certain criteria, to maximize responses at the interrogator's end.

We need to pass the query to a subset of people in the interrogator's social network. We select this subset of nodes (respondents) in the interrogator's network based on certain attributes associated with the node. Hence, we first need to associate certain attributes with each of the nodes, which aid in distinguishing them. The specific challenge that we tackle in this chapter is fixing these specific attributes and designing algorithms that carry out the seed selection process by picking nodes based on these attributes. We assign these attributes based on the respondent's action. The results suggest that if we select only a set of neighbours based on the probability of a particular action and pass the query accordingly at each level, almost half the number of nodes that receive the query will reply. We choose the set of neighbours such that the summation of the probability values of the nodes in the set is greater than one.

In this chapter, we will start with the related work done with respect to information diffusion and then discuss the actual problem in detail, with corresponding mathematical representation. This chapter will cover our proposed method for seed selection to address the challenge of replies maximization, and conclude with specific results.

2.1 Related work

Diffusion or propagation of an idea across a network has its usage in various fields. There has been a lot of work done in construction of models that aid in easy diffusion of an idea across a network. Various research projects have explored ways of using human networks in the process of solving imprecise, qualitative issues. This section consolidates the concepts of two such papers that provide an insight on creating models for the purpose of information diffusion and using human networks in recommending appropriate solutions.

David Kempe, Jon Kleinberg and Tardos [8] discuss the various models for spread of influence. The paper proposes various models for the process by which ideas and influences propagate through a social network. The paper discusses the fundamental algorithmic problem posed by Domingos and Richardson, which is, "if we can try to convince a subset of individuals and the goal is to trigger a large cascade of further adoptions, which set of individuals should we target?". The optimization problem of selecting the most influential nodes is NP-Hard. The natural greedy strategy obtains a solution that is probably within 63% of optimal solution. The paper says that, by using the analysis framework, one can suggest a general approach for reasoning about the performance guarantees of algorithms, for these types of influence problems. The paper provides computational experiments, on large collaboration networks, wherein the approximation algorithm significantly out-perform node selection heuristics based on degree and distance centrality. The paper illustrates the fact that a social network plays a fundamental role as a medium for spread of information. Aditya Pal discusses authoritative authors and ways to find them in micro blogging systems. In his paper [13], he proposes a set of attributes for characterizing users and then performs a ranking procedure to find interesting authors. His primary work is on micro blogs. The goal of our research is different but still we follow a similar approach to characterizing users and ranking them. Chen [3] evaluates four recommendation algorithms in a social networking site in his paper. He draws several design implications that helps users to produce better-received recommendations and find more contacts for users.

Adomavicius [1] presents an overview of the field of recommendation system. This paper gave us a lot of insight about general recommendation systems design techniques. Our primary goal is to design a recommendation system that uses the social network of the interrogator to recommend answers for his queries. Vinod Krishnan [9] compares "Movie Lens" algorithmic predictions with recommendations made by active Movie Lens users. The paper first defines a typical collaborative system and says that the early collaborative systems routed recommendations from one human to another. The paper then discusses the automated collaborative filtering system that aggregated opinions of a large set of users to recommend or predict for a target user. It generally uses the similarity of taste or similarity of items while giving out the predictions. The paper discusses the disadvantages of human when compared with that of a typical recommendation system. The main disadvantage is that humans lack the total quantity of data, a system possess. The advantage that they enjoy is that they are extremely good at processing a variety of heterogeneous data. The result of the paper is that Algorithmic collaborative filtering outperformed humans on average, though some individuals outperformed the system substantially and humans on average outperformed the system on certain prediction tasks. The paper concludes by acknowledging the fact that people are highly variable, but interpreting how people predict, especially the efficient ones who are systematically better, is valuable. Another implication is that Collaborative Filtering is stable, and evincing this against humans, only adds to its credibility. Sarwar [15] investigates the use of dimensionality reduction to improve performance of recommendation systems. His paper deals with traditional recommendation systems and their problems. The main limitation of the recommendations made by "Movie Lens" users is that the probability of the recommendations satisfying the interrogator's preference is low since general users may not necessarily have preferences similar to that of the interrogator. Our application utilizes the social network of the interrogator and hence the preference similarity of the respondents and the interrogator is higher. We can thus expect the users of our application to contribute more efficiently than a typical "Movie Lens" user.

2.2 Problem Definition

In this section, we describe the query diffusion process and its challenges. We do the diffusion process as the simulation of the framework we aim to design. This section discusses the general types of graph models (that have the characteristics of a real world social network), attributes of individual nodes, methods of carrying out query diffusion and our ultimate aim of maximizing replies.

2.2.1 Graphs and Attributes

We have given an overview of the graph models that we used, attributes of a node and the metrics for performance measurement in our previous chapter. In this section, we explain those attributes in mathematical terms and explain how we assign them in a typical simulation. Let G=(V,E) be the topology of the user's social network, which is constructed through a model, say for instance Preferential attachment graph model. Preferential Attachment is a type of graph where the probability of receiving new links of a node increases as it becomes more connected. Let the total number of nodes be N, that corresponds to the total number of individuals in a given user's social network.

In our research, we construct individual user profiles based on attribute values. We construct these attributes based on user's actions. Each individual can perform certain actions and each of these actions will have an associated value. Let A be the set of activities that a respondent can perform (as depicted in figure 2.1), such as reply, forward or ignore. Each of the respondents can perform any of these actions and each action has an associated probability.

Let $0 \leq a_{\mathrm{i,j}} \leq 1$, where $a_{\mathrm{i,j}}$ is the probability that user i, will perform activity j.

Each action occurs with a certain probability. If the user decides to act, then he would not ignore and vice versa.

$$\sum_j a_{i,j} = 1 \tag{2.1}$$

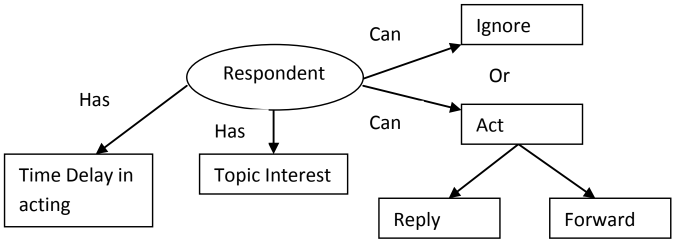


Figure 2.1: The figure shows the User attribute model. There are three types of attribute values associated with each of the user. Every user has a time delay in acting, even after receiving the query. We can find this by his past interactions with the system. Every user has a topic interest value, given a topic. The user can choose to act or ignore, when he receives the query. If a user chooses to act, he can reply to the interrogator or forward to his neighbours.

Let $m_{\mathrm{i,p}}$ be the interest in topic p for a user i. We assign four attributes to each of the node in the graph, based on the activity that they perform. A node can either choose to ignore a message or choose to act. In the real world scenario, we can find the probability value of a person performing an action, based on his/her past replies. For example, lets assume we connect the application to the social network of the interrogator (say twitter or facebook). We can associate the probability value for replying as the ratio of number of replies the respondent has provided to the total number of questions asked by the interrogator. In our simulation process, we assign the probability value based on random number generation. We generate a random number between 0 and 1 and assign it as the node's acting probability. The value obtained by subtracting the random number from one, gives the ignoring probability of that node (as both of these actions are mutually exclusive). Once a node decides to act, it can either reply or forward or do both. We have a probability associated with each of these actions. Let $q_{\mathrm{i,n}}$ be the n^{th} question asked by interrogator i. Each question has a topic vector $m^{q}_{\mathrm{i,p}}$ associated and a time constraint T^{q}_{i}. We assume that, for each respondent i, there is a mean activity delay d_{i} in (0, 1 hour) that indicates the time that it would take to act after receiving the query.

2.3 Main Challenge : Maximizing replies

Given a query, our research focuses on maximizing replies for the query. Each query has an associated time constraint. Hence, we need to maximize replies at the interrogator's end within the time limit associated with the query. Therefore, the query should reach the set of nodes that has a mean activity delay (defined above) less than the time left to answer the query. We define this set of nodes as Reachable node set. Given a graph, let $P_{0,\mathrm{k}}$ be a directed path from node 0 (source) to node k (respondent). A node k is relevant only when

$$R = \{k | \sum_{k \in P_{0,k}} d_k \leq T_{0,l}\} \tag{2.2}$$

where R is the reachable node set and $T_{0,\mathrm{l}}$ is time constraint associated with the l^{th} query for interrogator 0. Hence, one of our sub-goals is to maximize the cardinality of the set R. In this section, we explained the mathematical representation of attributes associated with each node and the main challenges associated with our research. The next section deals with designing algorithms that pick neighbourhood nodes based on their attributes.

2.4 Algorithm design to carry out query diffusion

We designed decentralized greedy algorithms to maximize query spread and in the process, attempt to maximize replies. These greedy algorithms pick neighbourhood nodes at each level of spread based on their attribute values. The attribute values are the probability values associated with the actions they perform and the topic interest value. This section gives a detailed explanation on how the seed selection algorithms work.

We carry out the query diffusion process using the seed selection algorithms on two graph models, namely Watts-Strogatz (Small World) and Barabasi model (Preferential Attachment).

2.4.1 Evaluation Metrics:

We analyze three factors based on the output: Cost with respect to number of activated nodes (received the query), Cost with respect to number of activated edges (edges that led to activated nodes) and the number of replies, based on a uniform probability. In our simulation, the first evaluation metric is the number of nodes that received the query. In real world scenario, number of activated nodes corresponds to number of people we are asking. As discussed before, we should not saturate the network by activating too many nodes in order to maximize the responses. We also need to make sure that we do not end up activating very few and fall short of responses. The second evaluation metric is the number of edges that led to activated nodes. Activated edges denote the path in which the query passes. We need to maintain a good ratio of activated nodes to activated edges. If there are less number of activated nodes and more number of activated edges, it essentially means that our system tries to pass the query to same person but through various paths. The third evaluation metric is the number of replies. In the real world scenario, we can find whether the reply received for a query is correct, based on the interrogator's feedback. In the simulation process we try to judge the answer based on random number generation. Once the query node receives an answer, we generate a random number between 0 and 1 and based on the range that this number falls into, we classify the response as correct/incorrect. We need to have high number of correct replies to ensure that our algorithm works efficiently.

The following is a detailed explanation of the first type of algorithm (nodes selected based their forwarding probability). The other types of algorithms adopt a similar process except that the node selection criteria will be different.

2.4.2 Seed selection algorithm:

In this section we discuss the seed selection algorithm that carries out the query diffusion process by selecting seeds on the basis of forwarding probability is given in Algorithm 1.

Algorithm 1: Algorithm that uses the seed selection technique of picking nodes based on forwarding probability, to carry out query diffusion

```
Input: $G := (V, E)$ - the social network (constructed by either of three graph models
        discussed), Let, for each of the nodes in the graph, have four of its attributes that
        are set, Replying Probability $a_{\mathrm{i},1}$,Forwarding Probability $a_{\mathrm{i},2}$, Ignoring Probability
        $a_{\mathrm{i},3}$ and Time Delay.
Output: Total number of activated nodes, total number of activated edges and total
         number of replies received
Let $0 \leq a_{\mathrm{i,j}} \leq 1$, where $a_{\mathrm{i,j}}$ is the probability that user (node) i, will perform activity j. Let
$a_{\mathrm{i,j}}$ in U(0,1).
for each node $v \in V$ that has received the query, starting from source node do
    if node $v \in V$ is not the source node then
        The node takes part in the diffusion process based on a coin toss.
    end
    if node $v \in V$ is a source node or a node that can participate in diffusion process then
        Let $S^v$ be the set of neighbours of $v$ in the decreasing sorted order of $a_{\mathrm{i}2}$;
        pick the top K nodes in $S^v$ such that
                    $$\sum_{i=1}^{|K|} a_{i,2} \geq 1 \qquad (2.3)$$
        if
                    $$\sum_{i=1}^{|S^v|} a_{i,2} < 1 \qquad (2.4)$$
        then
            Pass the query to all nodes in $S^v$
        end
    end
end
```

2.4.3 Time Complexity

We will discuss the time complexity of the above algorithm in this section. The algorithm has to compute the summation of forwarding probability of the neighbouring nodes at each level of query diffusion. The value of ignoring probability governs the forwarding probability. The ignoring probability determines the action of the node as, on a given chance, a node can choose to either ignore or forward/reply. Hence, it takes a complexity of the order of n in case of n nodes. In the diffusion process, each node finds the forwarding probability of all its neighbours and sorts them in the descending order of their forwarding probability. The former requires $m * n$ complexity, where m is the number of neighbours for each node. It takes $n * (n-1)$ in the worst case, that is O(n^2) complexity and the latter requires $O(nlogn)$ complexity. We also check the time delay of the nodes that reply and it requires additional n operations in the worst case. Hence the total time complexity of the algorithm is of the order of O(n^2).

2.5 Results and Analysis:

We have explained the algorithm and discussed the computational time complexity of the algorithm in the previous section. This section deals with the major outcomes of the algorithms and the corresponding explanation. We will first discuss how we construct a typical small world graph. We explain the construction of small world graphs of two, four, six and eight neighbourhood in figures 2.2, 2.3. A typical small world graph model requires three parameters: Number of nodes, neighbourhood value and rewiring probability. Number of nodes indicates the number of nodes to be present in the graph that we construct. If we consider a circular lattice of our graph, neighbourhood value indicates the number of neighbours that each node will have on either side. In a typical small world graph, rewiring probability indicates the probability value at which we remove an edge from any two neighbours and connect it to one other node in the graph at random.

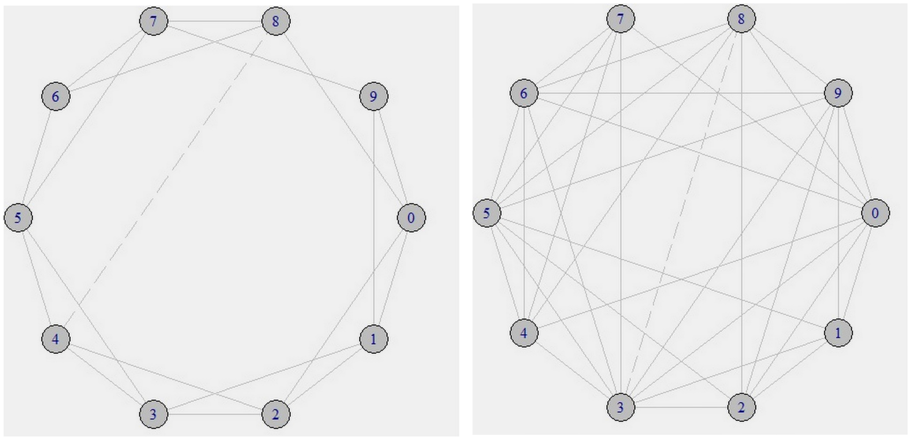


Figure 2.2: The figure on the left shows a two neighbourhood small world type of graph with probability of rewiring as 0.1. A small world graph model of two neighbourhood is constructed by first connecting each node with two of its neighbours on either side with an edge and based on the rewiring probability, we remove one of those edges and connect it to one other vertex in random. In the figure, we construct the small world graph by removing the edge from vertex 8 to vertex 9 and connecting to vertex 4 (denoted as dotted edge). Figure on the right shows a four neighbourhood small world type of graph with probability of rewiring as 0.1. A small world graph model of four neighbourhood is constructed by first connecting each node with four of its neighbours on either side with an edge and based on the rewiring probability, we remove one of those edges and connect it to one other vertex in random. In the figure, we construct the small world graph by removing the edge from vertex 8 to vertex 7 and connecting to vertex 3 (denoted as dotted edge).

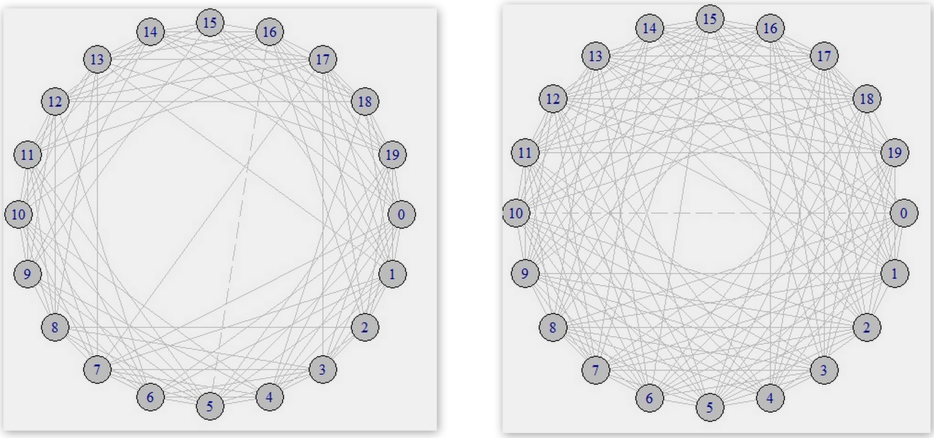
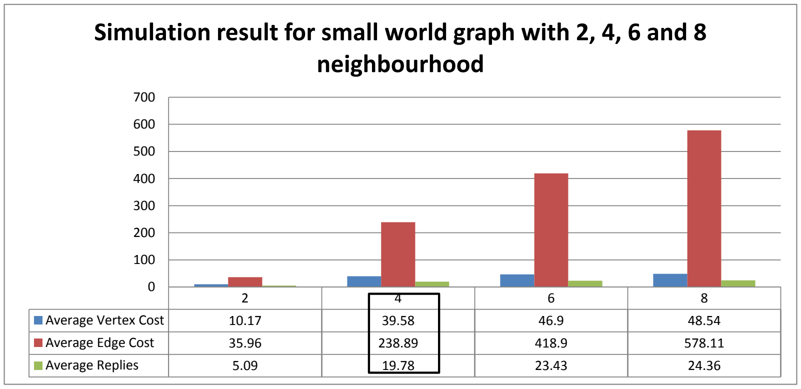


Figure 2.3: The figure on the left shows a six neighbourhood small world type of graph with probability of rewiring as 0.1. A small world graph model of six neighbourhood is constructed by first connecting each node with eight of its neighbours on either side with an edge and based on the rewiring probability we remove one of those edges and connect it to one other vertex in random. In the figure, the edge from vertex 16 to vertex 5 is constructed (denoted as dotted edge), after removing one. Figure on the right shows a four neighbourhood small world type of graph with probability of rewiring as 0.1. A small world graph model of four neighbourhood is constructed by first connecting each node with four of its neighbours on either side with an edge, and based on the rewiring probability we remove one of those edges and connect it to one other vertex in random. In the figure, the edge from vertex 0 to vertex 10 is constructed (denoted as dotted edge), after removing one.

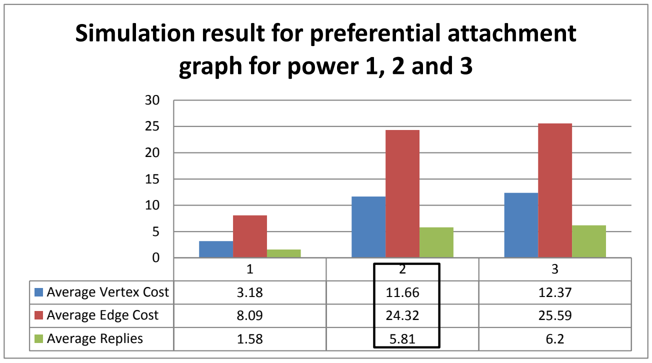
We constructed the graph models with 100 nodes. We simulated the diffusion process by making each node as a query node. For each query node, we ran the simulation 100 times as per the first algorithm and collected the results. Figure 2.4 indicates the output for a "Small World" type of graph for various neighbourhood. We average the probability of rewiring values ranging from 0.1 to 0.9 and compare the relative distribution. The main point in the diffusion process is the selection of neighbourhood nodes based on forwarding probability. There is a coin toss made during each of the selection, owing to the real world scenario, as a node can decide to spread or not. We calculate the replies based on a uniform distribution.



	2	4	6	8
🟦 Average Vertex Cost	10.17	39.58	46.9	48.54
🟥 Average Edge Cost	35.96	238.89	418.9	578.11
🟩 Average Replies	5.09	19.78	23.43	24.36

Figure 2.4: The figure shows the simulation result for the algorithm that carries out the query diffusion process on a small world graph model of 100 nodes. We constructed four types of small world graph based on their neighbourhood (2, 4, 6 and 8): denoted in X-axis. We averaged the three evaluation metric values collected for all values of p (probability of rewiring ranging from 0.1 to 0.9), for a particular neighbourhood of small world graph, as denoted in Y-axis. The highlighted value shows the best case of the simulation, wherein an interrogator with four neighbours would receive as much as twenty replies, when the query was diffused as per the algorithm.

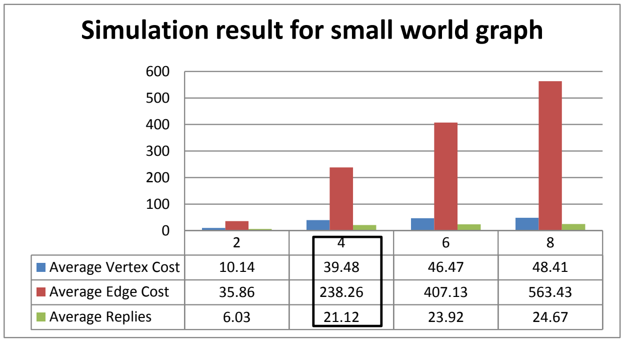
In both the cases (simulation results shown in Figure 2.4 and 2.5 we simulated the diffusion process by making each node as a query node. For each query node, we ran the simulation 100 times as per the algorithm and collected the results. The algorithm picks seeds based on their probability of forwarding the query. Average vertex cost is the average number of activated nodes (number of people who received the query). Average edge cost corresponds to number of activated edges (denote the path in which the query is passed to). Average replies is the average number of correct replies. The graph shown in 2.5 indicates the results obtained for a "Preferential Attachment" type of graph. We constructed three types of preferential attachment graphs based on the power of preferential attachment (1, 2 and 3). A preferential attachment graph of power one implies that it is a linear preferential attachment graph. We simulated the diffusion process by making each node as a query node.



	1	2	3
🟦 Average Vertex Cost	3.18	11.66	12.37
🟥 Average Edge Cost	8.09	24.32	25.59
🟩 Average Replies	1.58	5.81	6.2

Figure 2.5: The figure shows the simulation result for the algorithm that carries out the query diffusion process on a preferential attachment graph model of 100 nodes. We constructed three types of preferential attachment graphs based on the power of preferential attachment (1, 2 and 3): denoted in X-axis. We averaged the three evaluation metric values collected for a particular power of preferential attachment, as denoted in Y-axis. The highlighted value shows the best case of the simulation wherein, when the power of preferential attachment is as high as two, each node on an average receives atleast five replies for his/her query.

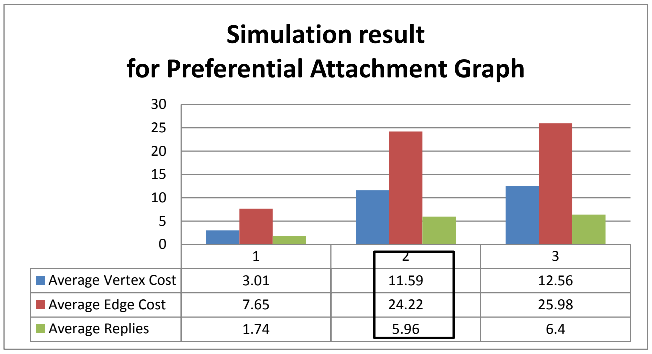
In case the algorithm picks nodes based on replying probability, we obtain similar results. Figures 2.6 and 2.7 indicate the results of the simulation run with algorithm that picks seeds based on their replying probability value in a small world graph model.



	2	4	6	8
🟦 Average Vertex Cost	10.14	39.48	46.47	48.41
🟥 Average Edge Cost	35.86	238.26	407.13	563.43
🟩 Average Replies	6.03	21.12	23.92	24.67

Figure 2.6: The figure shows the simulation result for the algorithm that carries out the query diffusion process on a small world graph model of 100 nodes. We constructed four types of small world graph based on their neighbourhood (2, 4, 6 and 8): denoted in X-axis. The highlighted value shows the best case of the simulation, wherein an interrogator with four neighbours could spread the query to almost 40 respondents on an average, when the query was diffused as per the algorithm.

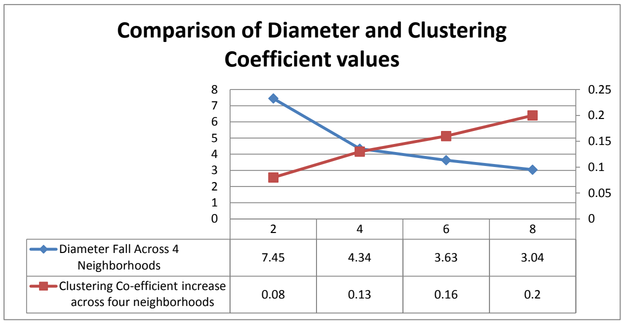
In case the algorithm picks nodes based on replying probability, we obtain similar results. Figure 2.7 indicate the results of the simulation run with algorithm that picks seeds based on their replying probability value in a "Preferential Attachment" type of graph.



	1	2	3
🟦 Average Vertex Cost	3.01	11.59	12.56
🟥 Average Edge Cost	7.65	24.22	25.98
🟩 Average Replies	1.74	5.96	6.4

Figure 2.7: The figure shows the simulation result for the algorithm that carries out the query diffusion process on a preferential attachment graph model of 100 nodes. We constructed three types of preferential attachment graphs based on the power of preferential attachment: denoted in X-axis. We averaged the three evaluation metric values as denoted in Y-axis. The highlighted value shows the best case of the simulation, wherein an interrogator in a graph with power of preferential attachment as two, could spread the query to almost 12 respondents on an average, when the query was diffused as per the algorithm.

We run the algorithm using a greedy approach. We search the optimal node to forward the query to, at each level. We have adopted this approach because of two constraints that the problem has. They are a) in real world scenario, each of the node can know more about their neighbouring nodes only and not the entire graph. Hence, a node cannot determine the optimal node of the entire graph b)the time constraint further restrains the dynamic programming technique, as spread could be maximized only based on higher values of forwarding probability among the neighbours of the given node. The probability of knowing a node with highest forwarding probability in the entire graph, within the limited time provided for the question, is minimal. We observed certain interesting trends from the results and we examine the corresponding reasons. The first interesting result was the ratio between number of activated vertices and the number of replies received in the end. It was approximately 2:1, which implies that roughly for every two nodes that received the question, one node chooses to answer. We found that this was because of the coin toss that we made for each node, when we wanted to determine if the node would reply or not. We implemented the coin toss as we wanted to choose amongst different options using a probability distribution. The second interesting trend in the results was the variation in the values of number of activated edges. The number of activated edges increases by a large quantity, though the number of activated vertices gets saturated (without large increase) in the small world type of graph. The third interesting trend in the results is the saturation of the number of activated vertices and edges, with different rewiring probability values in the same neighbourhood (without large increase). In order to explain the last two trends we computed the comparison of diameter and clustering coefficient values for different types of small world graphs.



	2	4	6	8
Diameter Fall Across 4 Neighborhoods	7.45	4.34	3.63	3.04
Clustering Co-efficient increase across four neighborhoods	0.08	0.13	0.16	0.2

Figure 2.8: The figure shows the diameter and clustering coefficient comparison on a small world graph model of 100 nodes. We constructed four types of small world graph based on their neighbourhood (2, 4, 6 and 8): denoted in X-axis. We averaged the diameter and clustering coefficient values collected for all values of p (probability of rewiring ranging from 0.1 to 0.9), for a particular neighbourhood of small world graph, as denoted in both the Y-axes. The main insight of the figure is that the decrease in the diameter of the graph as the rewiring probability value increases is comparatively lesser than the decrease, when the neighbourhood values increases. Also, the clustering coefficient value increases as the neighbourhood value increases.

We find from Figure 2.8 that the decrease in the diameter of the graph as the rewiring probability value increases is comparatively lesser than the decrease, when the neighbourhood values increases. Hence, the number of activated vertices and edges, among different rewiring probability values in the same neighbourhood, saturate (without large increase). We also find that the clustering coefficient value increases as the neighbourhood value increases. Thus, there is a strong probability that different nodes try to activate the same node, as the node would be present in various paths of diffusion. Hence, the number of activated edges increases by a large quantity, though the number of activated vertices saturate (without large increase).

2.6 Conclusion

In this section, we started discussing the main challenge of our research project. The main challenge of our research is to choose appropriate seeds in a given interrogator's network. In order to choose certain specific seeds among a set of people, we need to have attributes for each of the seed, so that we can select based on those attribute values. We fixed attributes based on the actions that user can perform. We discussed the user actions in the user activity model. We associated probability values for these actions. We constructed algorithms that picks seeds based on these attribute values. We computed the algorithms individually and simulated the corresponding diffusions to compute success, using the three metrics discussed. In the next chapter, we discuss techniques to combine the seed selection methods, discussed in this chapter so that it maximizes the number of replies. We will also explore ways to include the topic vector while performing seed selection, in order to maximize the quality of replies.

Chapter 3

GIANT COMPONENT GENERATION

We discussed different attributes that govern the seed selection process and the different seed selection algorithms based on these attributes in the previous chapter. In this chapter, we will discuss giant component of a graph and the ways of selecting seeds to maximize the spread of queries without compromising the quality of the replies. In order to maximize query spread, we address a sub challenge i.e., maximizing the cardinality of the set of nodes that receive the question. We give importance to the topic relevance factor of the node with respect to the query asked, to improve the quality of replies.

We start this chapter with related work done with respect to expert identification in a community and discuss the problem of a giant component generation in detail, with corresponding mathematical representations. The chapter will then cover the proposed method for seed selection, to address the challenge of maximizing replies and quality improvement, concluding with specific results.

3.1 Related Work

This chapter discusses appropriate ways to maximize spread and ways of improving the quality of replies. In accordance with that, there are two papers discussed in this section, which has considerable insight into the problems addressed in this chapter. Duncan J.Watts [17] presents a model that offers an explanation of social network searchability, in terms of recognizable personal identities, defined along a number of social dimensions. The paper talks about six degrees of separation and then defines searchability as a property of being able to find the target quickly. The study suggests that searchability exists in certain specific classes of networks, which either possess hubs or has an underlying geometry lattice, which acts as a proxy for social space. The paper also discusses the model of a social network based on plausible social structures. The author discusses six contentions regarding the social network, which endows individuals in the network with not only network ties, but also identities. The paper presents a hierarchy and a series of layers wherein group membership is the primary basis for social interactions, and then defines the node's identity as an H-dimensional co-ordinate vector. The authors then construct the social distance measure, which is the minimum of ultra-metric distances over all dimensions. The primary objective of the study was to determine conditions under which average length of a message chain is small.

Agrahri [2] investigates the fundamental questions of social search in his paper. He suggests that social search techniques might improve the effectiveness of web search engines. Harper explores the use of machine learning techniques to automatically classify questions as conversational or informational. In his study[5], he uses human networks to classify question types and achieves 89.7% classification accuracy.

Aditya Pal and Joseph A.Konstan [14] discuss the effective way of expert identification in a social community. The paper first defines a Community Question Answering (CQA) system as the one that provides a platform for internet users to form social communities and exchange knowledge. The CQA enables users to ask and answer questions. The paper suggests that, in such communities there exist a small number of experts amongst large population of users. The study tries to distinguish experts from ordinary users by using a Gaussian classification. The authors hypothesize that experts, who aim to provide answers, tend to prefer answering questions, which do not already have a good answer. This selective answering brings up the problem of "Question Selection Bias". The question selection bias effectively identifies the questions that a user would select for answering. The paper first proposes a model to capture the bias in CQA. The model categorizes questions by giving a value to it, which is the summation of value of its answers. The value of an answer is determined by the number of votes and status of the answer. It then selects users who have provided ten or more answers. The study suggests that there is a tendency of picking question with low existing value, prominent among experts. Ordinary users tend to answer a question after there are a few good answers already. The user selection bias is then calculated. The paper supports the hypothesis by depicting the various results obtained by carrying out experiments with its data set ("Turbo Tax Live Community"). The paper concludes by saying that expert selection in a social community is possible by selecting people, who choose (with high probability) questions with low existing values. Shardanand [16] describes a technique for making personalized recommendations to a user based on similarities between the interest profile of that user and those of other users. We do a similar implementation in our research. We construct profiles and check for topic similarity when we carry out the diffusion process. Ido Guy [4] focuses on recommendations derived from the user's social network. In his paper, he compares recommendations based on user's familiarity and his/her similarity network.

3.2 Problem Definition

We address the challenge of maximizing the spread by making sure that the set of nodes that receives the query forms a giant component of a graph. This section explains giant component and conditions that ensure the formation of a giant component in random graphs. A giant component is a connected component of a given random graph that contains a constant fraction of the entire graph's vertices [21].Our goal is to maximize replies for a given query and hence if the diffusion of the query generates a giant component in the given graph, replies are eventually maximized. The implication is that the query reaches all the nodes in the graph's giant component. We thus try to generate a giant component in the process of diffusion, starting with an initial set of seeds. In a random graph of N nodes and an underlying probability P for edge formation

$$N * P > 1 \tag{3.1}$$

implies that the probability of a giant component existing is non-zero. Equation 3.2 gives the probability of existence of a giant component

$$S = 1 - e^{-C*S} \tag{3.2}$$

where S is the probability of existence of a giant component. C is the average number of neighbours. Since it is certain that a giant component would always exist in a graph, we create a giant component in our graph by re-arranging the equation

$$C = -ln(1-S)/S \tag{3.3}$$

Average number of neighbours C, is given by the formula

$$C = (N-1) * P \tag{3.4}$$

Now, since the main problem is to create a giant component, in our case,

$$C = N * P * P_a * P_f \tag{3.5}$$

P_a is the acting probability Given a node will act, P_f denotes the probability that it would forward. Our aim now is to make sure that the nodes that received the query form a giant component in the graph. The next section will elaborate the approach followed to achieve the giant component formation.

3.3 Proposed Method

In this section, we discuss the algorithm implemented on random graph model (with probability as 0.1). The initial fixture being, in the given graph of 100 nodes, each node is selected as the source node and the information diffusion process is run 100 times, for a given source node. We combine the replying probability and forwarding probability of a node into a single attribute called acting probability. The acting probability of all the nodes is set to 0.20, 0.15 and 0.10 respectively. We analyze the following factors based on the output: cost with respect to the number of activated vertices, cost with respect to the number of activated edges, number of replies based on a uniform probability, the betweenness centrality of each source node and the number of neighbours of each source node.

Once we pick the initial source node, we follow a total neighbourhood seed selection process. If the node decides to forward, we select all the neighbours of a particular node to which we will forward the query. Asking everyone in the given user's neighbourhood and propagating, in case the user decides to forward, is our best case scenario and is the baseline algorithm that defines the value of acting probability at which the giant component is formed (with the nodes in that component being activated). We carry out the node selection/information diffusion algorithm as given in Algorithm 2.

One of the main outcomes of the giant component construction algorithm is the combination

Algorithm 2: Seed selection algorithm that carries out query diffusion by picking nodes to form a giant component of a graph with activated nodes (nodes that have received the query)

```
Input: $G := (V,E)$ - the social network (constructed by Random graph generation
       technique). Let each of the nodes in the graph have three of its attributes that are
       set, Acting Probability $a_{j,1}$, Ignoring Probability $a_{j,2}$ and Topic interest $a_{j,3}$.
Output: Total number of activated nodes, the total number of activated edges and the
        total number of replies received
Let $a_{j,1}$ be set to 0.20, 0.15 and 0.10 on each trial respectively.
for each node $v \in V$ that has received the query, starting from source node do
    if node $v \in V$ is not the source node then
        The node takes part in the diffusion process based on a coin toss.
    end
    if node $v \in V$ is a source node or a node that can participate in diffusion process then
        We toss a coin toss to determine whether the node would forward or reply. if node
        $v \in V$ decides to forward the query then
            Pass the query to all nodes in S
        end
    end
end
```

of replying and forwarding probability into a single attribute, acting probability. However, since broadcasting the message to all of its neighbours does not satisfy our goal, we carry out the second set of diffusion, in which we pick the seeds based on the summation of acting probability and topic relevance. This diffusion increases the amount of replies and the quality of replies. We carried out the node selection/information diffusion algorithm as given in Algorithm 3.

This section covered two algorithms out of which, one carries out the query diffusion process and forms a giant component using a best-case seed selection approach, and the other follows an optimal seed selection approach for query diffusion.

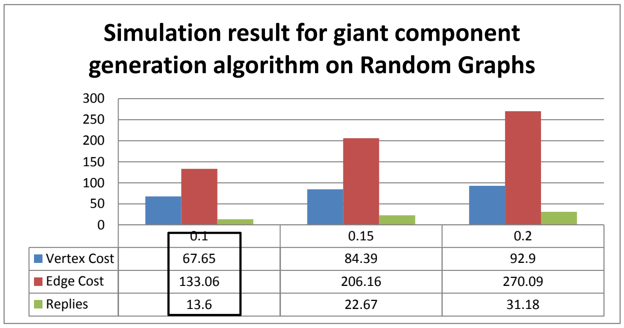
3.4 Results and Analysis

In this section, we will compare and contrast the results of the two algorithms discussed in Section 3.3. We simulate the diffusion process as per the giant component construction algorithm and collect the results. Figure 3.1 indicates these results for a random graph (with possible edge occurrence value p as 0.1). We average the values and compare the relative distribution. The main factor in the diffusion process is the selection of all neighbourhood nodes. In the real world, once a person decides to act, he can either reply or forward or do both. We generate random numbers in our simulation to assign the action taken by the node. If the number generated is less than 0.33, the node decides to reply. In case the generated random number is in the range 0.34-0.66, the node decides to forward, and if the number is between 0.67 and 1, it

Algorithm 3: Seed selection algorithm that carries out the query diffusion process by picking nodes based on fixed acting probability

```
Input: $G := (V, E)$ - the social network (constructed by Random graph generation
        technique). Let, for each of the nodes in the graph, have three of its attributes that
        are set, Acting Probability $a_{j,1}$, Ignoring Probability $a_{j,2}$ and Topic interest $a_{j,3}$.
Output: Total number of activated nodes, the total number of activated edges and the
        total number of replies received
Let $0 \leq a_{i,j} \leq 1$, where $a_{i,j}$ is the probability that user (node) i, will perform activity j. Let
$a_{i,j}$ in U(0,1).
for each node $v \in V$ that has received the query, starting from source node do
    if node $v \in V$ is not the source node then
        The node takes part in the diffusion process based on a coin toss.
    end
    if node $v \in V$ is a source node or a node that can participate in diffusion process then
        We toss a coin to determine whether it would forward or reply. if node $v \in V$
        decides to forward the query then
            Let $S^v$ be the set of neighbours of $v$ in the decreasing sorted order of $a_{j,2}$ + $a_{j,3}$;
            pick the top 'K' nodes in $S^v$ such that
                $$\sum_{l=1}^{|K|} (a_{l,2}) + (a_{l,3}) \geq 2 \qquad (3.6)$$
            if
                $$\sum_{l=1}^{|S|} (a_{l,2}) + (a_{l,3}) < 2 \qquad (3.7)$$
            then
                Pass the query to all nodes in S
            end
        end
    end
end
```

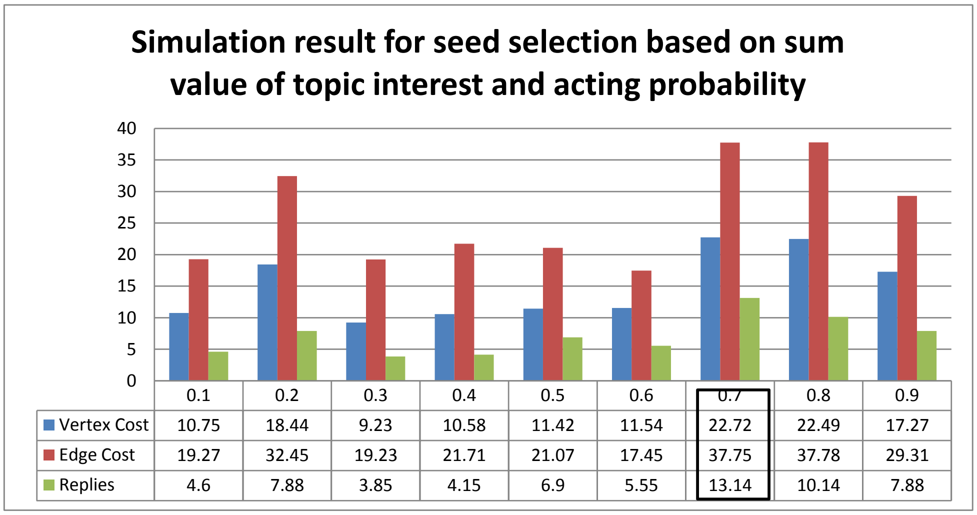
will forward as well as reply. We calculated the replies based on a uniform distribution.



	0.1	0.15	0.2
🟦 Vertex Cost	67.65	84.39	92.9
🟥 Edge Cost	133.06	206.16	270.09
🟩 Replies	13.6	22.67	31.18

Figure 3.1: The figure shows the simulation result for the algorithm that carries out the query diffusion process on a random graph model with the probability of constructing an edge as 0.1 and of size 100 nodes. We constructed three types of random graphs based on fixing three acting probability values for all nodes (0.1, 0.15 and 0.2): denoted in X-axis. We averaged the three evaluation metric values collected for a particular probability value, as denoted in Y-axis. The highlighted value shows the formation of giant component of nodes that has received the query. As much as 67% of total nodes are present in the giant component.

In both the simulation results (Figure 3.1 and Figure 3.2) we simulated the diffusion process by making each node as a query node. For each query node, we ran the simulation 100 times as per the algorithm and collected the results. Average vertex cost is the average number of activated nodes (number of people who received the query). Average edge cost corresponds to number of activated edges (denote the path in which the query is passed to). Average replies is the average number of correct replies. Though the algorithm runs on a best-case scenario, it full fills the essential requirement of the seed selection process. The algorithm combines both replying and forwarding probability in to a single attribute and ensures that there is a giant component formed even if the value of acting probability is as low as 0.2. In order to maximize spread and replies, the selection of seeds has to have the acting probability of the node as a distinguishing parameter. The other main factor to increase the quality of replies received is to include a topic relevance attribute for each node. It indicates that the node has a greater propensity of interest to the query asked. Hence, the sum of topic relevance and acting probability value is the optimal solution for seed selection. We simulate the diffusion process as the summation of two major attributes identified and collect the results. Figure 3.2 indicates the average cost value for a small world (with four neighbours and probability of rewiring as 0.1) graph.



	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
🟦 Vertex Cost	10.75	18.44	9.23	10.58	11.42	11.54	22.72	22.49	17.27
🟥 Edge Cost	19.27	32.45	19.23	21.71	21.07	17.45	37.75	37.78	29.31
🟩 Replies	4.6	7.88	3.85	4.15	6.9	5.55	13.14	10.14	7.88

Figure 3.2: The figure shows the simulation result for the algorithm that carries out the query diffusion process on a small world graph model of 100 nodes. We constructed small world graph of four neighbourhood with different probability of rewiring value (0.1 - 0.9): denoted in X-axis. We averaged the three evaluation metric values collected for all values of p (probability of rewiring ranging from 0.1 to 0.9), for a particular neighbourhood of small world graph, as denoted in Y-axis. The algorithm picks seeds based on the sum of topic relevance and acting probability value. The highlighted value shows the best case of the simulation. Even after taking the topic interest of each node into consideration to improve the quality of replies, we received as many as 13 replies for a given interrogator’s query with the average number of neighbours of the interrogator being as low as four.

We observe certain interesting trends in our results. One among these is the ratio between the average number of neighbours and the number of replies received. The average number of neighbours in the graph is close to five. The number of replies received at the user end was almost the same. Another important result is the maximum number of replies received. There were as many as 13 replies received for a case. Given a value four for the average number of neighbours, the number of replies is almost three times the number of friends the user has in his immediate social circle. The result re-iterates the fact that the number of replies maximizes when acting probability is included in the process of seed selection.

3.5 Conclusion

We discussed the algorithms for the formation of the giant component of a graph with nodes that have received the queries and the optimal case algorithm for query diffusion in this chapter. We implemented both the algorithms individually and simulated the corresponding diffusion, to compute the algorithms' efficiency. In the next chapter, we will discuss ways to incentivize people to participate in the query diffusion/answering process. The next chapter will also include the architecture and prototype of the mobile application that we design. The mobile application includes the findings of our research work and aids people in finding the right answer.

Chapter 4

INCENTIVIZING USERS TO RESPOND AND MOBILE APPLICATION DEVELOPMENT

In the previous chapter, we discussed giant component generation and the optimal seed selection algorithm for query diffusion. We design a mobile application framework that addresses two major problems. The first problem is distinguishing nodes and to address the issue, we assign a set of attributes to each node that differentiates them. The second problem is designing an algorithm that selects seeds based on the assigned attributes and we tackle the issue by developing an optimal seed selection algorithm. However, lack of contribution is a key challenge and a common problem that affects systems depending on human networks. Incentivization is one approach that encourages users to respond to queries and in turn maximizes the quality and quantity of replies received. In this chapter, we will be discussing the challenges associated with incentivization of users, our approach to the incentivization problem and a user survey to find out the most commonly asked questions, which individuals ask their friends. We will also discuss the general architecture and the prototype of a mobile application developed using the designed framework.

4.1 Related Work

This chapter discusses incentivizing users to respond. In this section, we focus on two papers, one that addresses under-contribution and other that gives an overview of a typical recommendation system for a group. Kimberly Ling [10] proposed various experiments to address the issue of under-contribution. He discusses various experiments that address under-contribution. He says that the main factor that leads individuals to contribute is reminding them of their uniqueness and giving them specific and challenging goals. The paper suggests major principles which when applied would solicit contributions. The first principle is to identify abstract mental states, like reminding the uniqueness of the individuals. The second principle is to design persuasive messages or manipulation that would solicit contributions. The paper also discusses similarity/homogeneity of the group being an important factor, in the sense, people contribute more when they believe they are similar to the group than otherwise. The paper then discusses four experiments for motivating contributions. They are 1) motivating contributions through group homogeneity and individual uniqueness 2) motivating contributions by framing uniqueness and benefits 3) motivating contributions through benefits and 4) motivating contributions through goal setting. McCarthy discusses ways of displaying social media in a workplace context to improve relationships among colleagues. In his paper [12], he highlights the ways in which the system has increased interaction. In our work, we try to incorporate the basic point that this paper addresses- how the creation of a shared window into online media has affected the use of that media.

Many Question-and-Answer websites allow people to post their questions and find answers for them from a community of users. Quora is one such website. Like any other recommendation system that relies on human networks, Quora also has to overcome the problem of under-contribution. Quora has a unique way of incentivizing users. Quora first awards every user 500 points on opening an account. The "Ask me to answer" option allows users to select other users who can answer their query. The above-mentioned process costs 50 points. If the user to whom one directs the question opts not to answer even after a week, then the system refunds 75% of the points to the user who asked the query. Users can offer more credits than the standard number of points for their questions. Respondents can earn credits by answering using the "Ask to answer questions" option or by getting votes for his/her previous answers. The volume of requests that the user receives determines the number of points for a question.

We found this way of incentivizing users to be fair and efficient and tried to adopt a similar process into our mobile application framework, which we discuss later in this section. Harper [6] investigates predictors of answer quality through a study of responses across several online Question and Answering sites. He confirms that the quality of the answer is higher in a fee-based site than in the free sites. One other point that he discusses in his paper is very critical with respect to our design of seed selection. He suggests that incentivizing people to respond with answers led to better outcomes. Joseph F. McCarthy [11] describes in his work titled "Pocket Restaurant Finder": A situated recommender system for groups, a recommendation system that suggests alternatives to a group of people and individuals can use in physical contexts. The recommendation system works in one of the following three ways: 1. analyzing the profile of the user 2. analyzing the profile of other users who are similar 3. analyzing the recommended content. Such a recommendation system also takes into account the preference of the whole group while suggesting. Kazienko [7] focuses on building a social recommender system that supports the creation of new relations between users in a multimedia sharing system. The paper confirms that relationships between users result either from semantic links between objects they operate on or from social connection of these users. In our research, we distinguish users by main attributes: topic similarity and acting probability.

4.2 Problem Definition and Proposed Method

In this section, we discuss the problems associated with incentivization with respect to our framework and an optimal method that would address the challenges associated with incentivization. The best way to improve responses is by making the process of participation a fun activity for the user. We gamify the whole process, in which individuals gain or lose points based on their action, to incorporate the fun aspect. While awarding points, we make sure to split the points amongst the users, without any cycle or loop formation in the answering paths.

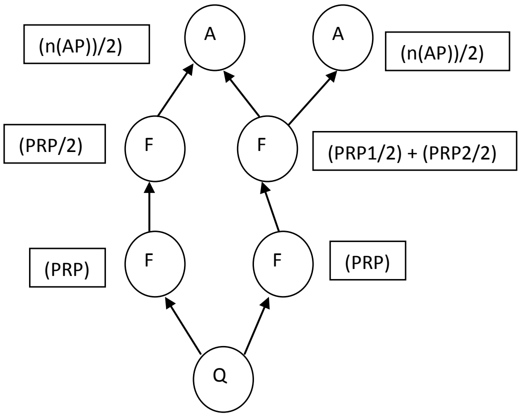


Figure 4.1: The figure shows the way in which we split the points amongst nodes, by the Points Split algorithm. "A" node is the answering node, "F" node is the forwarding node and "Q" node is the query node. Let AP denote the directed path that led to the answer. $n(AP)$ represents the total number of points along each path that led to answer. "PRP" represents the number of points that remain in the given answering path.

The explanation for Figure 4.1 is as follows. The algorithm splits the score equally amongst all the paths that led to the right answer. We consider each path that led to the answer as a tree structure with the node that answered/replied as the root. We award half the points allotted for the path to the root. From the root, we halve the points as the algorithm traverses each of the level in the tree, until the source node. The node that got the query from the source node will not pass the remaining points to the next level (i.e. source node) and hence will have all of the remaining points. We avoid loop formation by ensuring that a node (person) will not ask the question to any of the ancestor nodes in its path.

We allot an initial set of 5000 points to each node in the graph. In our simulation process, we split the points after the query diffuses and the query node receives at least one reply. In the real world scenario, we can determine the correctness of the reply received for a query based on the interrogator's feedback. In the simulation process, we try to judge the answer based on random number generation. Once the query node receives an answer, we generate a random number between 0 and 1 and based on the range that the number falls into, we classify the response as correct/incorrect. Once we find the correct number of replies, and in case it is more than zero, we reduce 50 points from the query node that asked the question.

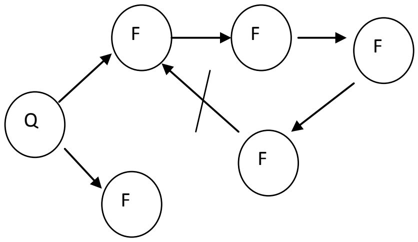


Figure 4.2: The figure shows a scenario in which a loop formation occurs when the query diffusion process takes place. "F" node is the forwarding node and "Q" node is the query node. We need to avoid sending a query through the struck edge to make a loop-free query diffusion.

We account for the loop prevention problem by maintaining a square matrix of the total number of nodes. In the matrix, each node (corresponding to each row in the matrix) will have its entire ancestor's (that has received the question already) column value as one. Before asking a question to a new node, we check the corresponding column value in the forwarding node's row. The algorithm does not forward the questions if the column value is one, to ensure that it does not pass the question again, to its ancestor (that has received the question already). In case the corresponding column value is not one, the algorithm forwards the query to the new node and changes the column value of the forwarding node in the new node's row to one. The algorithm of points splitting amongst other nodes is as follows

4.2.1 Algorithm

In this section we will present the Point Split algorithm that splits the points amongst the nodes that led to the answer. We carry out the Point Split algorithm as given below.

Algorithm 4: Point Split algorithm that carries out query diffusion and splits points amongst the nodes that are present in the path that led to the person, who provided correct answer

```
Input: $G := (V,E)$ - the social network(constructed by Random graph generation
        technique), Let, the diffusion process be carried out based on the optimal seed
        selection algorithm
Output: points Left in each of the node
Let there be an initial amount of points provided to every node $v \in V$, say 5000.
Let the number of correct answers be $n(A)$.
Let AP denote the directed path that led to the answer
Let the total number of points to be awarded for a question be $P$.
Let $P_{0,\mathrm{k}}$ be a directed path from node 0 (user) to node k. A node k is relevant only when
the reply it has provided is correct.
Let the points remaining in an answering path be $P(AP)$ whose initial value is $P/n(A)$.
for Each answering path do
    for each node $v \in P$ starting from answering node do
        if node $v \in V$ has its ancestor as query node then
            Points gained by $v$ = $P(AP)$
        end
        Points gained by $v$ = $P(AP)/2$
        $P(AP) = P(AP)/2$
    end
end
```

4.3 Results and Analysis

In this section, we analyze the output of the simulations carried out using algorithm 4. We used small world graph model of four neighbours and 0.1 probability of rewiring. We carried out the whole diffusion process hundred times for each node. We started the diffusion process by assigning 5000 points to each node. As per the algorithm, we split the points whenever we receive a correct reply. We carried out the diffusion process, by selecting all the neighbours of the given node and passing the query. We then carried out the diffusion process by selecting top three neighbours(on the sum of their acting probability and topic relevance value) of the given node and passing the query. We carried out the whole diffusion process for hundred times for each node. Figure 4.3 and 4.4 shows the points left amongst users, after the diffusion process was simulated.

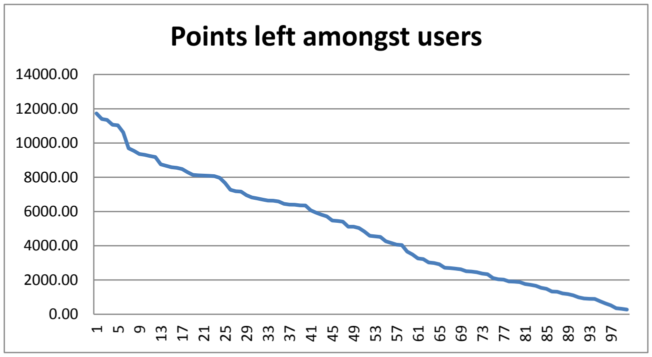


Figure 4.3: The figure shows the points left, amongst all nodes, after we run the points split algorithm. Initial points allotted to all the nodes were 5000. We carry out the diffusion process by selecting all neighbours, at each level of diffusion. The points left amongst the hundred users has been sorted in descending order and plotted.

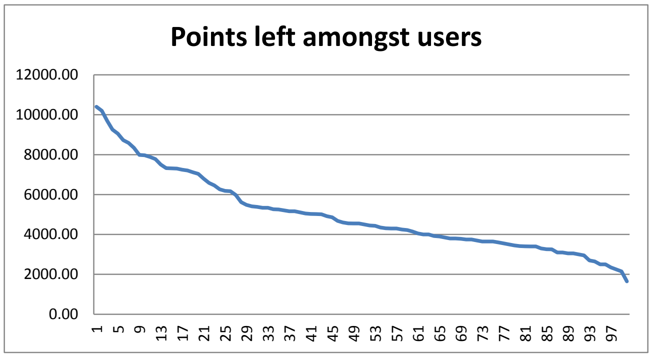
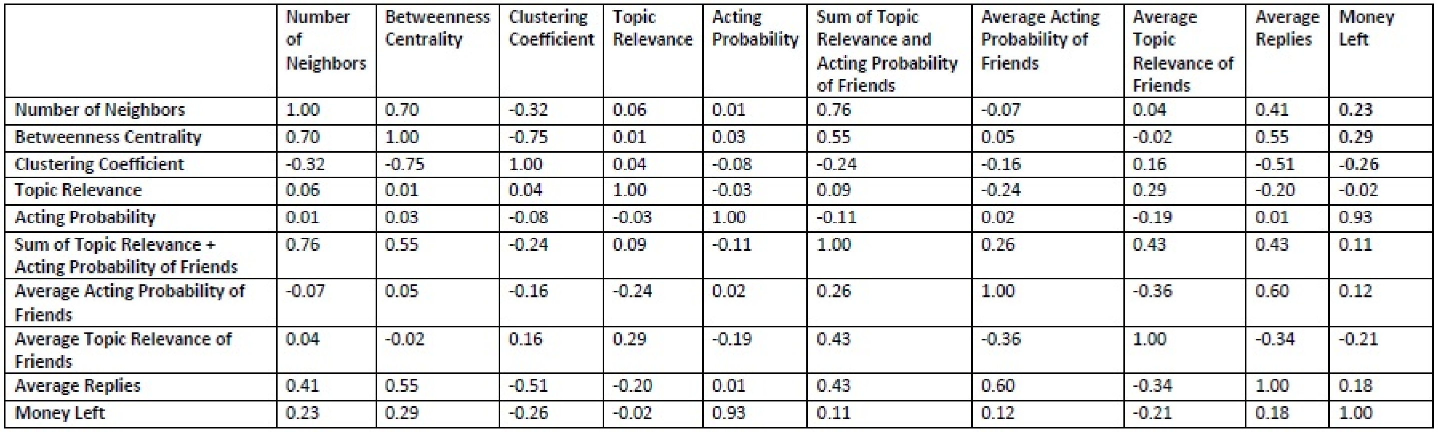


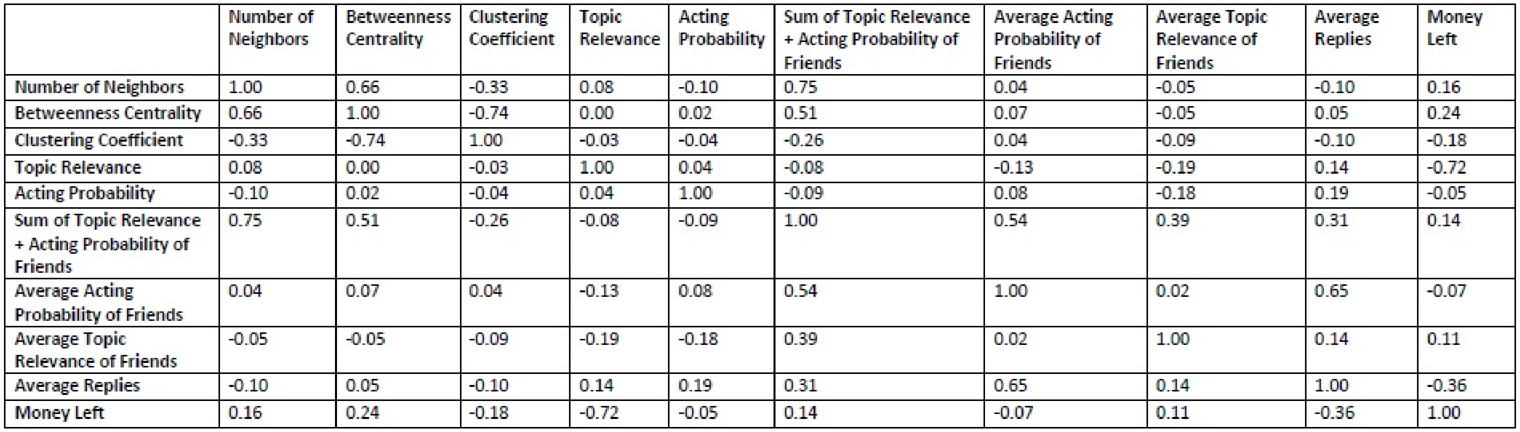
Figure 4.4: The figure shows the points left, amongst all nodes, after we run the points split algorithm. Initial points allotted to all the nodes is 5000. We carry out the diffusion process by selecting top three neighbours as per the seed selection process, at each level of diffusion. The points left amongst the hundred users has been sorted in descending order and plotted.

We then analyzed the relationship between the Money/Points left (after following incentive mechanism) and network structure. We measured the Betweenness Centrality, Clustering Coefficient and Number of Neighbours for each node. Once we got the result i.e. the Money/Points left at the end of diffusion, we calculated the correlation of each of the factor with the Money/Points left.



	Number of Neighbors	Betweenness Centrality	Clustering Coefficient	Topic Relevance	Acting Probability	Sum of Topic Relevance and Acting Probability of Friends	Average Acting Probability of Friends	Average Topic Relevance of Friends	Average Replies	Money Left
Number of Neighbors	1.00	0.70	-0.32	0.06	0.01	0.76	-0.07	0.04	0.41	0.23
Betweenness Centrality	0.70	1.00	-0.75	0.01	0.03	0.55	0.05	-0.02	0.55	0.29
Clustering Coefficient	-0.32	-0.75	1.00	0.04	-0.08	-0.24	-0.16	0.16	-0.51	-0.26
Topic Relevance	0.06	0.01	0.04	1.00	-0.03	0.09	-0.24	0.29	-0.20	-0.02
Acting Probability	0.01	0.03	-0.08	-0.03	1.00	-0.11	0.02	-0.19	0.01	0.93
Sum of Topic Relevance + Acting Probability of Friends	0.76	0.55	-0.24	0.09	-0.11	1.00	0.26	0.43	0.43	0.11
Average Acting Probability of Friends	-0.07	0.05	-0.16	-0.24	0.02	0.26	1.00	-0.36	0.60	0.12
Average Topic Relevance of Friends	0.04	-0.02	0.16	0.29	-0.19	0.43	-0.36	1.00	-0.34	-0.21
Average Replies	0.41	0.55	-0.51	-0.20	0.01	0.43	0.60	-0.34	1.00	0.18
Money Left	0.23	0.29	-0.26	-0.02	0.93	0.11	0.12	-0.21	0.18	1.00

Figure 4.5: Correlation values amongst pair of factors, when we carry out query diffusion process by passing query to top three neighbourhood nodes (arranged in the descending order of the sum of topic value and acting probability)at each level when a given node decides to act.



	Number of Neighbors	Betweenness Centrality	Clustering Coefficient	Topic Relevance	Acting Probability	Sum of Topic Relevance + Acting Probability of Friends	Average Acting Probability of Friends	Average Topic Relevance of Friends	Average Replies	Money Left
Number of Neighbors	1.00	0.66	-0.33	0.08	-0.10	0.75	0.04	-0.05	-0.10	0.16
Betweenness Centrality	0.66	1.00	-0.74	0.00	0.02	0.51	0.07	-0.05	0.05	0.24
Clustering Coefficient	-0.33	-0.74	1.00	-0.03	-0.04	-0.26	0.04	-0.09	-0.10	-0.18
Topic Relevance	0.08	0.00	-0.03	1.00	0.04	-0.08	-0.13	-0.19	0.14	-0.72
Acting Probability	-0.10	0.02	-0.04	0.04	1.00	-0.09	0.08	-0.18	0.19	-0.05
Sum of Topic Relevance + Acting Probability of Friends	0.75	0.51	-0.26	-0.08	-0.09	1.00	0.54	0.39	0.31	0.14
Average Acting Probability of Friends	0.04	0.07	0.04	-0.13	0.08	0.54	1.00	0.02	0.65	-0.07
Average Topic Relevance of Friends	-0.05	-0.05	-0.09	-0.19	-0.18	0.39	0.02	1.00	0.14	0.11
Average Replies	-0.10	0.05	-0.10	0.14	0.19	0.31	0.65	0.14	1.00	-0.36
Money Left	0.16	0.24	-0.18	-0.72	-0.05	0.14	-0.07	0.11	-0.36	1.00

Figure 4.6: Correlation values amongst all pair of the following factors: Number of Neighbours, Betweenness Centrality, Clustering Coefficient, Topic Relevance, Acting Probability, Sum of friends Attributes, Friends' Average Acting Probability, Friends' Average Topic Relevance, Average Replies and Money/Points left, when we carry out query diffusion process by passing query to all neighbourhood nodes, at each level.

In order to analyze the impact of network structure, we analyzed three main factors that best describes the network structure of a node: Betweenness Centrality, Clustering Coefficient and Number of Neighbours. In both the cases, we found Betweenness Centrality and Number of Neighbours to have a positive correlation with the Money/Points left, whereas the Clustering Coefficient has a negative correlation.

Compared Quantities	Top three neighbours	All neighbours
Money/Points left and Betweenness Centrality	0.24	0.29
Money/Points left and Number of Neighbours	0.15	0.23
Money/Points left and Clustering Coefficient	-0.17	-0.25

Table 4.1: Comparison of correlation values between Money/Points left and Betweenness Centrality, correlation values between Money/Points left and Number of Neighbours and correlation values between Money/Points left and Clustering Coefficient, when we carry out query diffusion process by passing query, to top three neighbourhood nodes (arranged in the descending order of the sum of topic value and acting probability), and all neighbourhood nodes, at each level when a given node decides to act

4.3.1 Betweenness Centrality:

Betweenness Centrality is a measure of a node's centrality in a network. It is equal to the number of shortest paths from all vertices to all others that pass through that node [19]. Betweenness Centrality is a more useful measure (than just connectivity) of both the load and importance of a node. Table 4.1 gives correlation values between Money/Points left and the Betweenness Centrality. As per the value indicated in table 4.1, Betweenness Centrality and Money/Points left have a positive correlation, which essentially means that a node gains more points if it has a higher Betweenness Centrality value. The gain is because a higher Betweenness Centrality value indicates a higher probability of being present in more number of paths (from all vertices to all other vertices). This increases the probability of the node being present in a path that leads to the answer and hence has a higher probability of gaining points.

4.3.2 Number of Neighbours:

As per table 4.1, Number of Neighbours and Money/Points left have a positive correlation, which essentially means that a node gains more points if it has more neighbours. This is because the algorithm presents this node with more number of queries to answer than a node with fewer neighbours.

4.3.3 Clustering Coefficient:

Clustering Coefficient is a measure of the degree to which nodes in a graph tend to cluster together. The local Clustering Coefficient of a vertex (node) in a graph quantifies how close its neighbours are to being in a clique [20]. As per the value indicated in table 4.1, Clustering Coefficient and Money/Points left have a negative correlation, which essentially means that a node loses more points if it has a higher Clustering Coefficient value. When the Clustering Coefficient value of a node is high, the node's neighbours have a higher probability of being in a clique. In our algorithm, before we pass the query to a node, we verify that the node we pass the query to is not one amongst its ancestors that has received the query already. If the Clustering Coefficient of a node is more, it essentially means that the probability that it has received the query from a different path/node increases. This is because its neighbours have a higher chance of being in a clique and hence the node that sent the query might have sent to its neighbour also. Therefore, the probability of the node being in a path that leads to answer decreases.

4.3.4 Linear Regression Model :

We built a linear regression model for the two cases, viz., passing the query to all neighbours and to top three neighbours at each level. We used the proc Reg procedure available in the statistical analysis package SAS to build the regression model. We built the model with money/points left being the dependent or the X-variable and all the other attributes being the independent or Y variables. We see that the model built with all the attributes is statistically significant, having a p value of < 0.0001 and R2 values of 0.66 and 0.94 for the models that message top three and all neighbours respectively. p value indicates the probability that an event occurs by chance alone and not otherwise. Most research studies will have a p-value cut off of 0.05, below which an event is statistically significant. R-squared values indicate that the linear regression model we developed can explain 66% and 94% of the variance in the respective models. Figure 4.7 and 4.8 shows the QQ plot or the quantile residual plots for both the cases, which indicate that our data fits the model very well, with only a few outliers near the upper tail of the graphs.

We built the model with money/points left being the dependent or the X-variable and all the

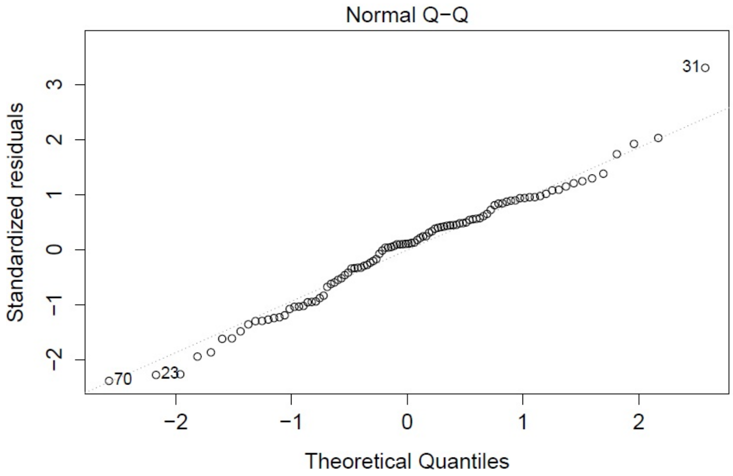


Figure 4.7: The figure shows the linear regression model for, passing the query to top three neighbours at each level.

other attributes being the independent or Y variables, as shown in Figure 4.7 and 4.8. We see that the model built with all the attributes is statistically significant, having a p value of < 0.0001 and R2 values of 0.66 and 0.94 for the models that message top three and all neighbours respectively. p value indicates the probability that an event occurs by chance alone and not otherwise. R-squared values indicate that the linear regression model we developed can explain 66% and 94% of the variance in the respective models.

We aimed to design an incentivization mechanism for the sole purpose of overcoming the prob-

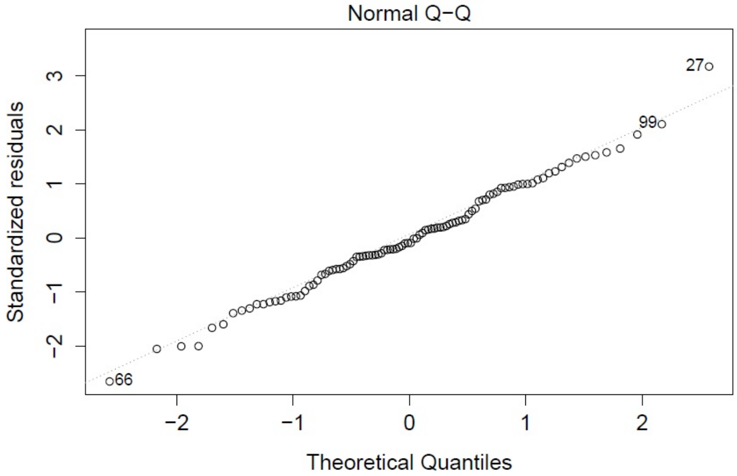


Figure 4.8: The figure shows the linear regression model for, passing the query to all neighbours at each level.

lem of under-contribution. Once we designed a setup for incentivization (by the usage of points) and an algorithm that distributes it, our focus was on making the whole of the incentivization process fair and efficient. After we simulated the query diffusion and implemented the incentivization process, we found through the results that we succeeded in our aim. After analyzing the results, we found few parameters that make the incentivization process very interesting and opens up a whole lot of experiments and analysis that we need to do to incorporate the parameters and modify the routing algorithm. We aim to design a system that learns continuously and incorporate the parameters in the future work to improve the quality of seed selection.

4.4 Mobile Application:

In the last section, we discussed incentivizing users using Point Split algorithm, once query diffusion is finished. The Point Split algorithm is the underlying logic for incentivizing users of the framework we aim to design, which is a mobile application framework. This section and the following sections provide a detailed description of the mobile framework.

4.4.1 Categorization of question

In this section, we will present the different categories/topics of question that we expect the user to ask his friends. We found a set of thirty questions by conducting a survey among 35 graduate students of Arizona state university to find out what questions they generally ask their friends rather than finding online answers. The list of questions is as follows

Question	Average Points
Grocery stores/restaurant/shops/cycle shops/saloon	49.5
Apartment related queries (safety, rent, cleanliness etc.)	49.2
Freebies inside campus (Free food, free coupons/t-shirts etc.)	49
Tourist destination tips	49
Car (rides/to be used for a day)	47.4
Money related (bank accounts, investments, insurance, claims)	45.2
Classes' information (coaching classes)	45
Academic related questions (courses, information about TAs)	45
Roommate search	45
Relocating help	42.5
Simple recipes	42.5
Club details (Music, dance, sports)	40.8
Festival and celebration related details (Indian)	40.1
Equivalent items [e.g.: brand of rice or liquor]	37.8
Sports related (like league, timing, venue, procedure to join)	36.5
Places to stay (cheap motels nearby or in a particular place)	34.5
Attire related (traditional, cheap)	32.6
Travel ticket Booking (agent, offers, friends to accompany)	31.5
Job search reference	30
Gift suggestion	30
Doctor Consultancy (Medical queries including tablets enquiry)	30
Repair shops (from bikes to mobiles)	28.4
Software related issues (Bestsoftware for a particular task)	27.5
Reviews (Resume/articles/blogs)	25
Gadgets related (e.g.: Xbox)	20
Safety related (General roads to be avoided)	20
Electronic gadgets (to buy)	17.5
Movie ratings	15
Website suggestion	15

Table 4.2: List of questions from survey: We formed a set of thirty questions (sorted in descending order in the Table) by conducting a survey amongst 35 graduate students of Arizona state university to find out what questions they generally ask their friends rather than finding online answers. We also asked them the points that they would award if they got an answer for the question. We have averaged the points for each question and sorted the questions in descending order of their average points.

After finishing the survey, we group the set of most asked questions into five categories as given in Table 4.3.

Question Category	Example
Where (Finding a Place)	Freebies, Restaurant, salon, furniture
How (Finding a Procedure)	Recipes, insurance, bank investments
Can someone (Finding a Person)	Car ride, lend gadgets, review
Which (Finding the best)	Club, league, best coaching class, best course
Other (Other type of queries)	movie rating, subject expert

Table 4.3: Categorization of Queries with examples: After we found the set of questions from the survey we did with the student of Arizona state university, we categorized the questions in to five. The five categories are Place related, Procedure related, Person related, Qualitative and other general queries. We have explained the five types of queries with example in the table.

We do the query formation as a three-step process: Form a question, add attributes like cost/type (e.g. cheap/Indian cuisine) and add the corresponding Time attribute

4.4.2 Seed selection

In this section, we will describe the actions that the user can perform in the application and the underlying seed selection logic of the application. We allow the user to perform only three actions, ask a question, reply to a question received or forward a question. The system will select the seeds based on topic relevance for the first time, as the probability of replying would be the same for all the seeds. We compute the topic relevance by asking for user's interests, when they login for the first time. From then on, the system will construct the profiles for each user. The system database would store and then learn from the profiles of users to find short cuts.

4.4.3 Points Distribution

In this section, we will discuss the distribution of points once the diffusion process is over and the interrogator has received replies. In our mobile application, after starting with a set of points, we reduce the points each time an interrogator asks a specific question. We also reduce few points once a respondent forwards a question. The user's points increases only when he/she answers a particular query or is present in a path that led to the answer. The interrogator gives a feedback after receiving an answer, and we distribute the points based on a tree structure, with the answer provider as the root. There is also an option given to the interrogator for finding a short cut, in addition to finding his/her friends in the social network, in which case he has to pay additional points.

4.4.4 Architecture and Prototype

In this section, we will discuss the system architecture of the mobile application. Our mobile application essentially has two major components. The first component allows the user to create the query and picks seeds from his network to pass the query. We have discussed the procedure for seed selection in our previous sections. We use a database for storing user profiles that would aid in seed selection. The second component is the Receiver. It constantly listens i.e. keeps track of the query and finds the nodes that are involved in answering the query. It retrieves the answer back to the interrogator. We have provided the block structure of the architecture in Figure 4.11.

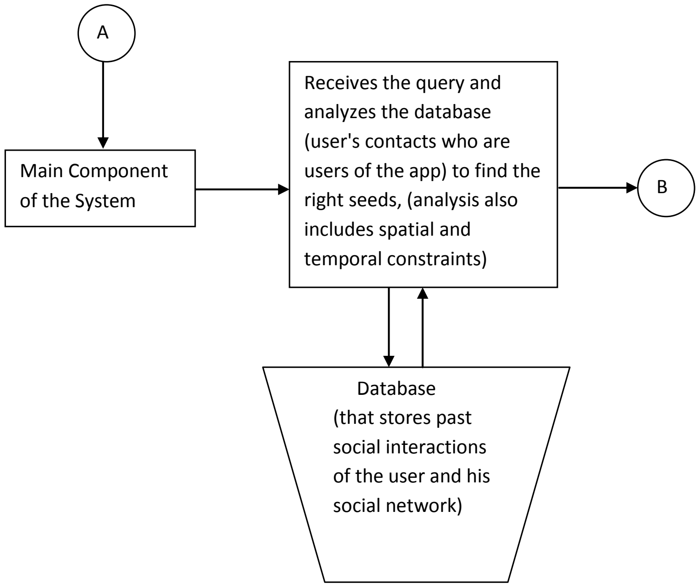

A
Main Component of the System
Receives the query and analyzes the database (user's contacts who are users of the app) to find the right seeds, (analysis also includes spatial and temporal constraints)
B
Database (that stores past social interactions of the user and his social network)

Figure 4.9: The figure shows the main component of the system. Main component of the system aids in forming the query and choosing seeds in the interrogator's social network by analyzing the database. It uses the optimal seed selection algorithm for query diffusion process.

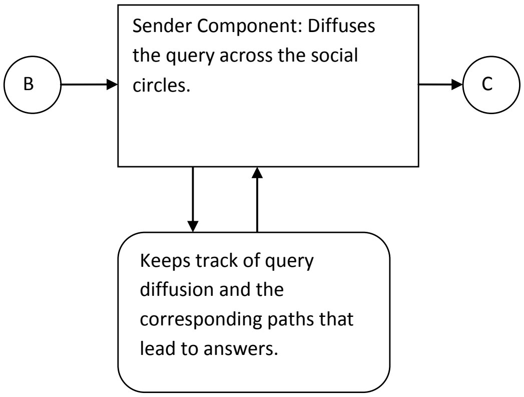

Sender Component: Diffuses the query across the social circles.
B
C
Keeps track of query diffusion and the corresponding paths that lead to answers.

Figure 4.10: The figure shows the Sender component of the system. Sender component diffuses the query across the social circle of the interrogator and keeps track of the path of diffusion. Sender component aids in splitting points across the nodes that answer the query.

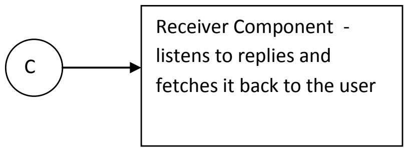
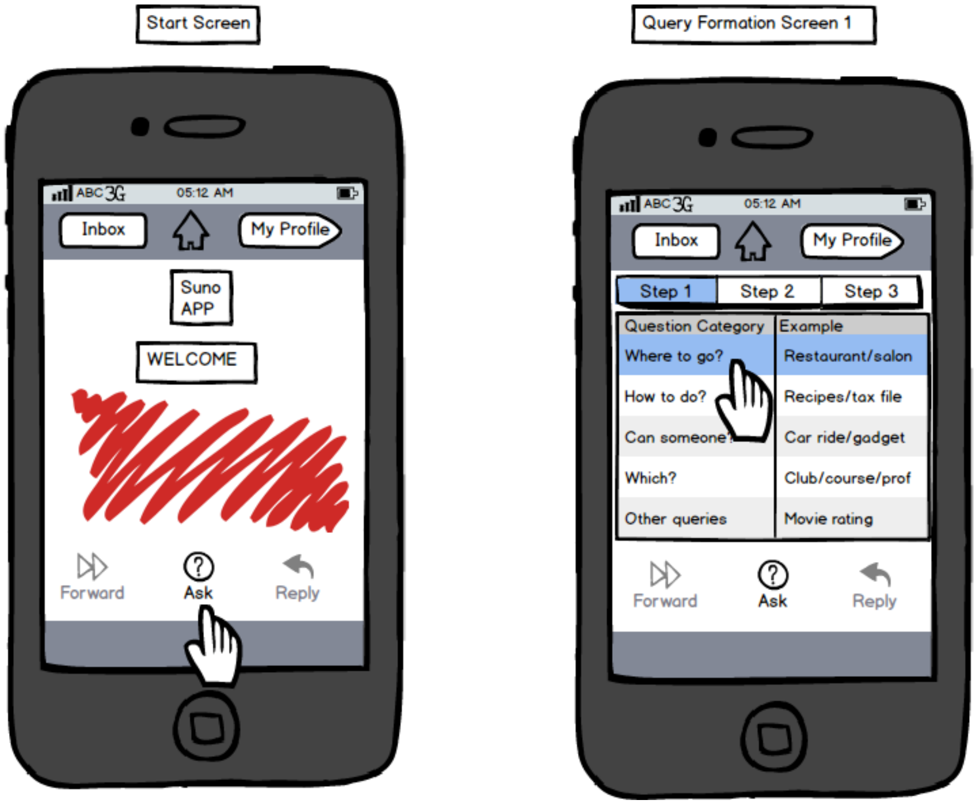

C
Receiver Component - listens to replies and fetches it back to the user

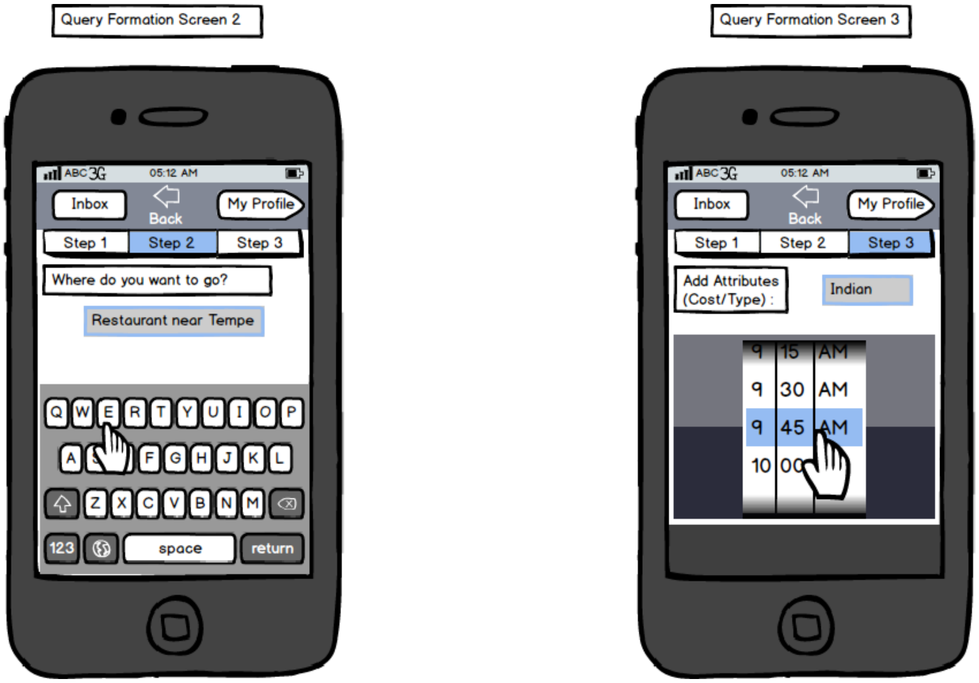
Figure 4.11: The figure shows the Receiver component of the system. The receiver component keeps track of the replies and sends it back to the interrogator's inbox.

We created the prototype of the mobile application. The prototype is the model of how the mobile application works. We analyzed the different actions that the user can perform in the previous section and the underlying algorithm for seed selection in the user's network. We discussed the system architecture in the last part. We have designed the prototype that covers the working of the mobile application. The prototype is as follows



Question Category	Example
Where to go?	Restaurant/salon
How to do?	Recipes/tax file
Can someone?	Car ride/gadget
Which?	Club/course/prof
Other queries	Movie rating

Figure 4.12: The figure on the left shows the home screen of the mobile application and shows that the user has selected Ask(which means that he has a query to find an answer). The one on the right shows the main component of the application, querying friends. The figure on the right lists the categories of questions from which the user can choose.



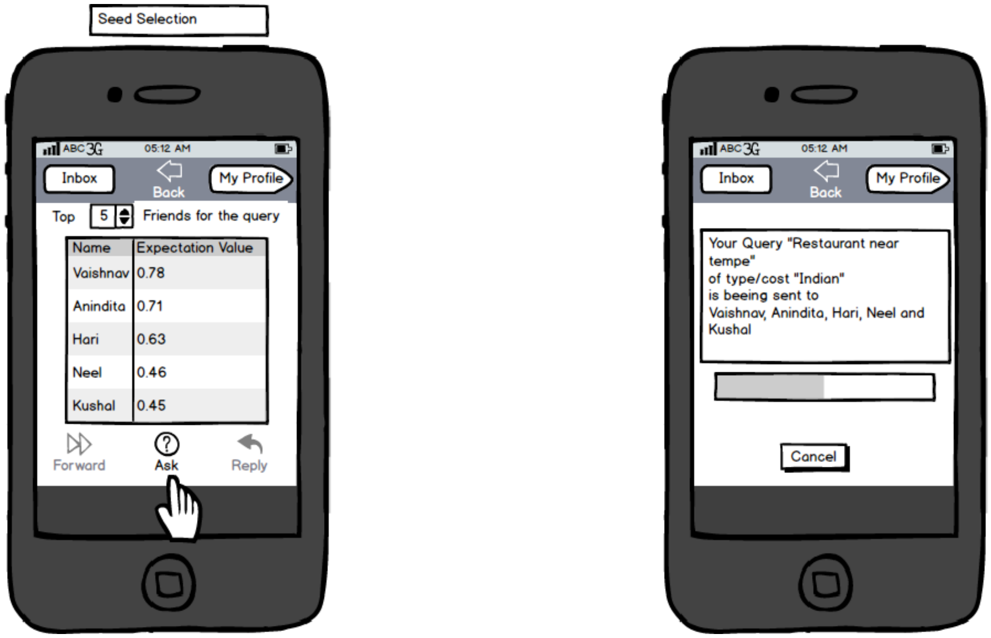
Where do you want to go?

Restaurant near Tempe

Add Attributes (Cost/Type) :

Indian

Figure 4.13: Both the figures show the query formation process by an interrogator. We do the query formation as a three-step process: Form a question (Figure on the left), add attributes like cost/type (e.g. cheap/Indian cuisine) and add the corresponding Time attribute (Figure on the right side).

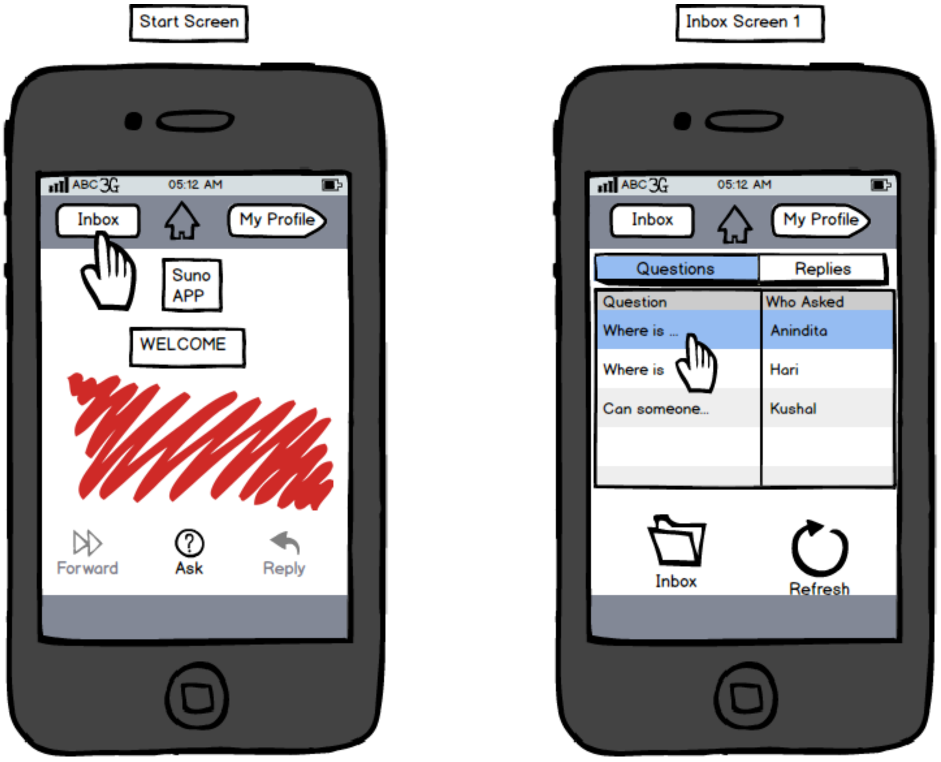


Top 5 Friends for the query

Name	Expectation Value
Vaishnav	0.78
Anindita	0.71
Hari	0.63
Neel	0.46
Kushal	0.45

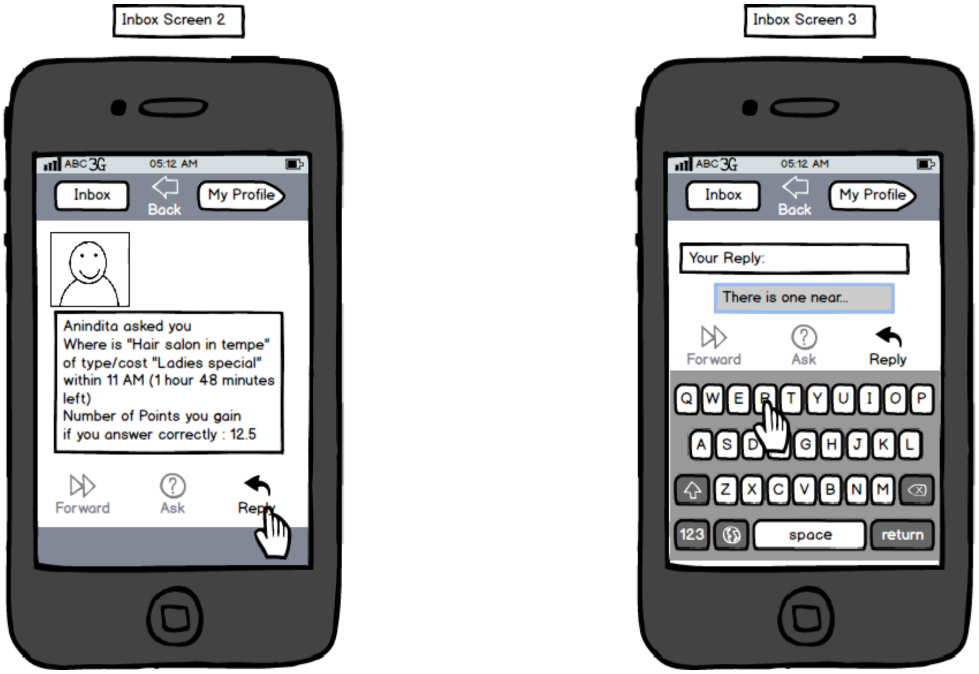
Your Query "Restaurant near
tempe"
of type/cost "Indian"
is beeing sent to
Vaishnav, Anindita, Hari, Neel and
Kushal

Figure 4.14: The figure shows the resulting seeds (top five) selected from the user's mobile network by the application for the interrogator's query. The figure on the right shows that the algorithm forwards the query to the selected seeds.



Question	Who Asked
Where is ...	Anindita
Where is	Hari
Can someone...	Kushal

Figure 4.15: The figure on the left shows the home screen of the mobile application and the user is selecting Inbox. Figure on the right side shows the inbox of the user. As depicted in the second figure, the algorithm categorizes the user's inbox into two: Queries forwarded to the user and replies received for the user's queries. We see that the inbox has the questions tab and the first query received by the user selected.

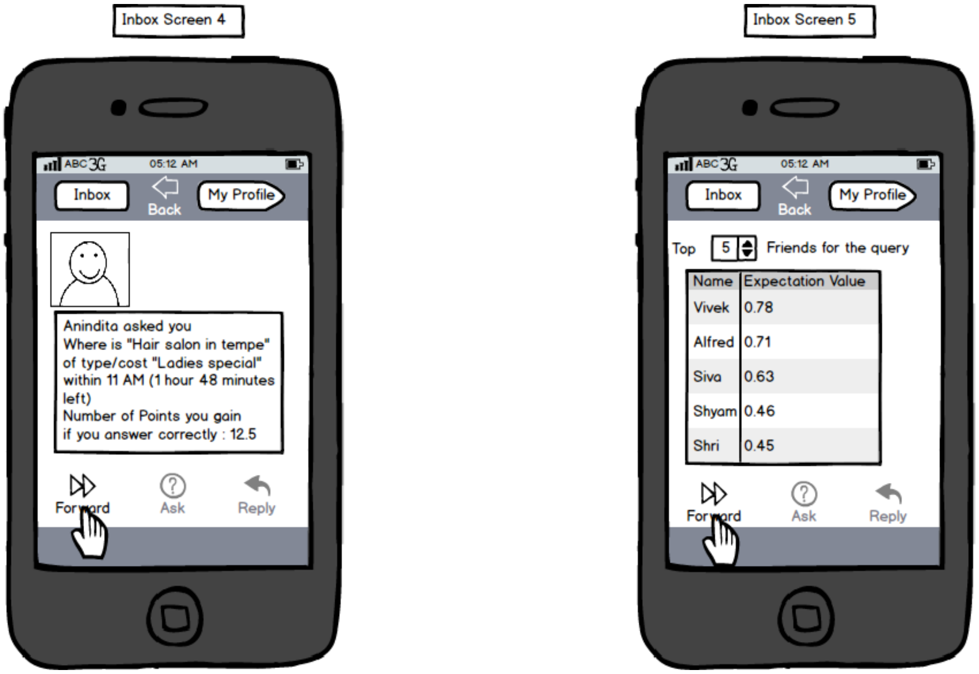


Anindita asked you
Where is "Hair salon in tempe"
of type/cost "Ladies special"
within 11 AM (1 hour 48 minutes
left)
Number of Points you gain
if you answer correctly : 12.5

Your Reply:

There is one near...

Figure 4.16: The figure on the left shows the first query that the user has received. It also mentions the points that the user would gain in case his reply is correct and denotes that the user has decided to reply. Figure on the right shows the user's reply screen and depicts his reply.



Anindita asked you
Where is "Hair salon in tempe"
of type/cost "Ladies special"
within 11 AM (1 hour 48 minutes left)
Number of Points you gain
if you answer correctly : 12.5

Top 5 Friends for the query

Name	Expectation Value
Vivek	0.78
Alfred	0.71
Siva	0.63
Shyam	0.46
Shri	0.45

Figure 4.17: The figure on the left shows the first query that the user has received. It also mentions about the points that the user would get in case his reply is correct. Figure on the right side shows that the respondent has decided to forward and hence it shows the resulting seeds(top five) selected, from the user's mobile network, by the application for the query.

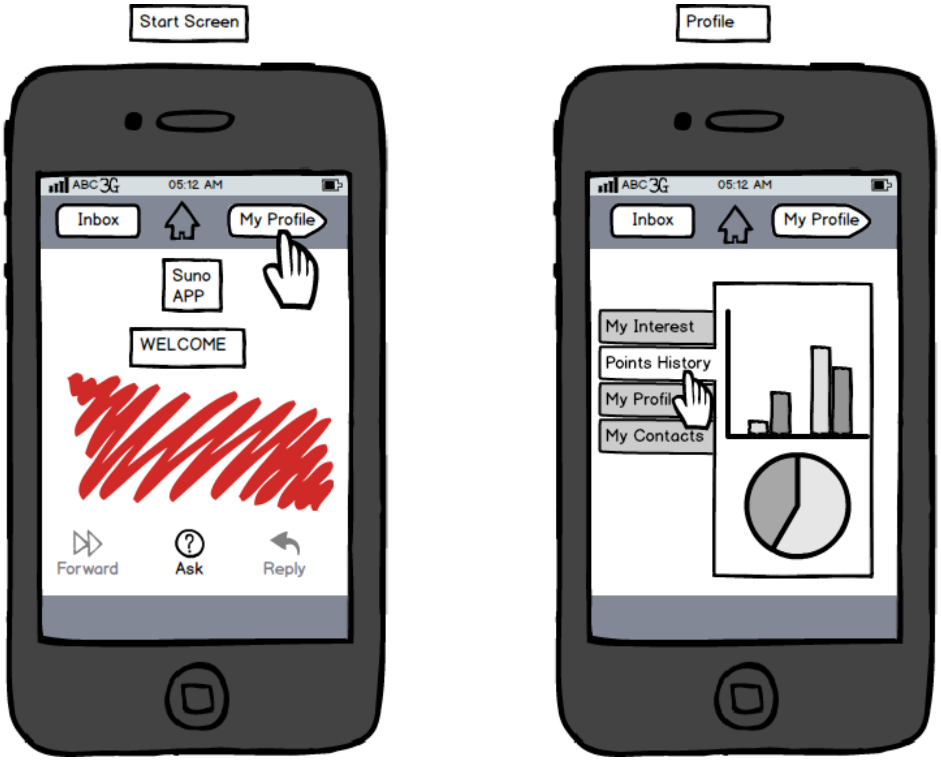


Figure 4.18: The figure on the left shows the home screen of the mobile application and the user is selecting My Profile. The Figure on the right side shows the user's profile wherein he selects his points history .

4.5 Conclusion

In this chapter, we discussed the algorithm that aids in incentivizing users to respond to the query, which in turn addresses the problem of under-contribution in the system. We designed a framework that tries to maximize the replies and hence incentivizing users to contribute would make a large difference in terms of the quantity and quality of answers received at the interrogator's end. The last section of this chapter gives an outer look on the categorization of questions and the detailed architectural design of the mobile application that uses our framework.

Chapter 5

CONCLUSION

Our research aims to find answers for user's queries. We designed a mobile application framework to address this goal. As a simulation of the framework designed, we carried out the query diffusion process in various graph models to improve efficiency in seed selection. The framework has certain essential parts. The first main part of the framework is the set of attributes that are associated with the nodes(people) in the social network of the user. We fixed the attribute values based on user's actions. We used these attribute values to differentiate nodes effectively so that a subset of neighbours is picked at each level of query diffusion. We formed the attributes so that when we run a seed selection algorithm using these attributes, the quality and quantity of replies increases.

5.1 Main Contribution:

Human networks act as an essential resource when the information searched for is imprecise and qualitative. Since we might not find the answer for the query in the immediate social circle of the user, our framework utilizes the user's network efficiently so that we neither end up saturating the whole network nor end up falling short of user's expectation. We need to have an optimal seed selection process that governs the query diffusion, in order to carry out an efficient utilization of respondents in the interrogator's network. Our research work always tries to address the challenge of maximizing the replies received. We tried out several algorithms for seed selection in order to compute, compare and contrast the different metrics of measurement that accounts for the success of the overall goal. The major algorithms that we analyzed for the purpose of seed selection were Seed selection based on sum of all attribute values or each of the individual attributes, Diameter and Cluster Coefficient comparison of a typical network, Giant Component discovery in graph, Nodes picked based on summation of Topic Value and Acting Probability (finalized optimal selection process). We used three graph models in our research that simulates a real world social network. The three graph models are Watts-Strogatz (Small World), Barabasi (Preferential Attachment) and Random Graphs model. When we simulated the seed selection algorithm that selects seeds based on their probability of forwarding a query, we found that an interrogator with four neighbours received as many as twenty replies, when we diffused the query. When we simulated the seed selection algorithm that selects seeds based on their probability of replying to a query we found that an interrogator with four neighbours could spread the query to almost 40 respondents on an average. When we simulated the giant component generation algorithm, as much as 67% of total nodes are present in the giant component (formed with nodes that received query). We set the baseline algorithm (giant component generation algorithm) by selecting the entire set of neighbours of a seed and passing the query at each level. We then designed an optimal seed selection algorithm that governs the query diffusion process by selecting a proper subset of nodes at each level. Even after taking the topic interest of each node into consideration to improve the quality of replies, when we picked nodes based on summation of Topic Value and Acting Probability (finalized optimal selection process) we found that an interrogator received as many as 13 replies for a query with the average number of neighbours of the interrogator being as low as four.

Our next challenge was to avoid under-contribution in the social network. We addressed the challenge by incentivizing users to participate in the search process. We aimed to make the whole process of asking, forwarding or answering a fun activity for the user so that he/she enjoys participating. We associated a set of points for each question and split it across the people who led to/provided the correct answer. We used Point Split algorithm that carries out query diffusion and splits points amongst the nodes that are present in the path that led to the person who provided correct answer. After simulating the Point Split algorithm, in order to analyze the impact of network structure on points distribution, we analyzed three main factors that best describes the network structure of a node: Betweenness Centrality, Clustering Coefficient and Number of Neighbours. In both the cases, we found Betweenness Centrality and Number of Neighbours to have a positive correlation with the Money/Points left, whereas the Clustering Coefficient has a negative correlation.

5.2 Future Work

Our ultimate goal is to implement a mobile application framework. We have a detailed design of the architecture of the mobile application. We have also created the prototype of the mobile application. The research work extends with the creation of the mobile application that uses the optimal algorithm for query diffusion and user incentivization. We need to cross verify the results we got from the simulation with the ground truth, once the application is created. Hence implementing the whole mobile application is a work that we aim to complete in the future. We also plan to introduce a feedback system so as to improve the quality of replies. With the introduction of feedback, we need to change the seed selection algorithm to incorporate the feedback factor. This would improve the quality of seed selection. The future work would also include exploring ways to prevent the saturation of the whole network by having an upper limit of activated nodes. There are few open issues in our research. We created the points split algorithm and analyzed factors that would impact the point distribution by finding correlation values. One open issue is that we need to analyze in detail, each of the factor and modify the seed selection algorithm accordingly. There are two types of factors that impact the node's points: Network structure and Individual attributes. We need to find the factor that would have a greater impact on a node and include that in our seed selection approach. One other open issue is to analyze the impact of neighbours of a node on its points (gain/loss). If in case a node v is a neighbour of a node u with high acting probability or high betweenness centrality value, we need to research on factors that would aid or affect v.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.

[2] Arun Kumar Agrahri, Divya Anand Thattandi Manickam, and John Riedl. Can people collaborate to improve the relevance of search results? In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 283–286. ACM, 2008.

[3] Jilin Chen, Werner Geyer, Casey Dugan, Michael Muller, and Ido Guy. Make new friends, but keep the old: recommending people on social networking sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 201–210. ACM, 2009.

[4] Ido Guy, Naama Zwerdling, David Carmel, Inbal Ronen, Erel Uziel, Sivan Yogev, and Shila Ofek-Koifman. Personalized recommendation of social software items based on social relations. In *Proceedings of the third ACM conference on Recommender systems*, pages 53–60. ACM, 2009.

[5] F Maxwell Harper, Daniel Moy, and Joseph A Konstan. Facts or friends?: distinguishing informational and conversational questions in social q&a sites. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 759–768. ACM, 2009.

[6] F Maxwell Harper, Daphne Raban, Sheizaf Rafaeli, and Joseph A Konstan. Predictors of answer quality in online q&a sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 865–874. ACM, 2008.

[7] Przemyslaw Kazienko, Katarzyna Musial, and Tomasz Kajdanowicz. Multidimensional social network in the social recommender system. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(4):746–759, 2011.

[8] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[9] Vinod Krishnan, Pradeep Kumar Narayanashetty, Mukesh Nathan, Richard T Davies, and Joseph A Konstan. Who predicts better?: Results from an online study comparing humans and an online recommender system. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 211–218. ACM, 2008.

[10] Kimberly Ling, Gerard Beenen, Pamela Ludford, Xiaoqing Wang, Klarissa Chang, Xin Li, Dan Cosley, Dan Frankowski, Loren Terveen, Al Mamunur Rashid, et al. Using social psychology to motivate contributions to online communities. *Journal of Computer-Mediated Communication*, 10(4):00–00, 2005.

[11] Joseph F McCarthy. Pocket restaurantfinder: A situated recommender system for groups. In *Proceedings of the Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems, Minneapolis*, 2002.

[12] Joseph F McCarthy, Ben Congleton, and F Maxwell Harper. The context, content & community collage: sharing personal digital media in the physical workplace. In *Proceedings*

of the 2008 ACM conference on Computer supported cooperative work, pages 97–106. ACM, 2008.

[13] Aditya Pal and Scott Counts. Identifying topical authorities in microblogs. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 45–54. ACM, 2011.

[14] Aditya Pal and Joseph A Konstan. Expert identification in community question answering: exploring question selection bias. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1505–1508. ACM, 2010.

[15] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Application of dimensionality reduction in recommender system-a case study. Technical report, DTIC Document, 2000.

[16] Upendra Shardanand and Pattie Maes. Social information filtering: algorithms for automating Şword of mouthŤ. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217. ACM Press/Addison-Wesley Publishing Co., 1995.

[17] Duncan J Watts, Peter Sheridan Dodds, and Mark EJ Newman. Identity and search in social networks. *science*, 296(5571):1302–1305, 2002.

[18] Wikipedia. BarabásiŰalbert model — wikipedia, the free encyclopedia, 2013. [Online; accessed 21-July-2013].

[19] Wikipedia. Betweenness centrality — wikipedia, the free encyclopedia, 2013. [Online; accessed 9-July-2013].

[20] Wikipedia. Clustering coefficient — wikipedia, the free encyclopedia, 2013. [Online; accessed 9-July-2013].

[21] Wikipedia. Giant component — wikipedia, the free encyclopedia, 2013. [Online; accessed 20-May-2013].

[22] Wikipedia. Random graph — wikipedia, the free encyclopedia, 2013. [Online; accessed 21-July-2013].

[23] Wikipedia. Small-world network — wikipedia, the free encyclopedia, 2013. [Online; accessed 21-July-2013].