IDENTIFICATION OF INFLUENTIAL USERS IN SPEECH-BASED NETWORKS

Aizaz Anwar, Department of Computer Science, Information Technology University of The Punjab, Lahore, Pakistan,aa364@itu.edu.pk
Sameen Mansha, Department of Computer Science, Information Technology University of The Punjab, Lahore, Pakistan, sameen.mansha@itu.edu.pk
Faisal Kamiran, Department of Computer Science, Information Technology University of The Punjab, Lahore, Pakistan, faisal.kamiran@itu.edu.pk
Asim Karim, Department of Computer Science, Lahore University of Management Sciences, Lahore, Pakistan, akarim@lums.edu.pk

Abstract

Online social networks act as good mediums for communication but are also becoming popular for targeting the social needs of their users. Mainstream social networks are still unable to incorporate low-literate users into their user-base as their interfaces are on the web or in Short Message Service (SMS) format, while low-literate people constitute a major portion of world's population. Speech-based networks (SBNs) overcome these limitations by providing a simple speech-based interface for users. In this work, we present a systematic analysis of SBNs designed specifically for low-literate users with a focus on identification of influential users. The task of finding influential users has not been studied for SBNs. Furthermore, knowledge of influential users can help optimize the operation of SBNs that typically run in low-income regions with low budgets. We demonstrate how a SBN is formed from call data records and define its key features or characteristics. We then propose a feature-based method for influence ranking in a SBN. Existing methods for influence maximization in social networks are not directly applicable to SBNs. Hence, we present a method for calculating influence probabilities between users in the network, enabling the application of the greedy algorithm and degree discount heuristic for influence maximization and computation of betweeness centrality in a SBN. We evaluate our methodology on data from a real-world SBN called Polly. We compare the results with those from existing methods and show that our methods are both effective and time-efficient for use in SBNs.
Keywords: Speech-based Networks, Social Networks, Influence Maximization, Betweenness Centrality.

1 INTRODUCTION

With the inception of high speed internet and fast computing resources, online social networking has become an essential tool to connect various offline communities all across the globe turning the world into a global village. Despite the growth of social networks and their emergence as an important aspect in the current society, a large portion of the world's population is still unable to be part of the online social network. The prime causes of this situation are rooted in low-literacy, poverty and in some places, lack of internet connectivity. It has been demonstrated that speech-based and non-textual systems are favored by low-literate users over text-based ones (Medhi et al., 2006). For this reason, some speech-based networks have been introduced or have adapted themselves according to the needs of low-literate users by providing speech-based interfaces for communication (Sherwani et al., 2007; Plauché and Nallasamy, 2007; Veeraraghavan et al., 2007; Patel et al., 2010, 2012; Raza et al., 2012). In terms of general user interface for developing regions, some speech-based networks focus on speech and push-button dialogue systems requiring neither literacy nor training (Raza et al., 2013). A simple telephone-based voice manipulation and forwarding system is a good medium for communication and dissemination of development-related information, e.g., provision of job ads, group messaging facility etc.

Speech-based networks (SBNs) have some issues that limit their scalability and sustainability in the current form. A major issue is that most of its users have low buying power and they cannot pay charges for the services. For example, consider Polly : Polly is a telephone-based entertainment service that allows any caller to record a short message, modify it using a choice of funny sound effects, and forward the modified recording to their friends by phone numbers. Polly was first tested in 2011 among low skilled office workers in Lahore, Pakistan. Within 3 weeks, Polly spread to 2,032 users, engaging them in 10,629 calls. However, it was shut down due to unsustainable cellular airtime cost and insufficient telephone capacity (Raza et al., 2012). This hurdle in the widespread use of SBNs can be overcome by devising a mechanism to effectively utilize the resources of the underlying network. In particular, social influence of different users of the network can be used as a decision-making factor for the allocation of resources. If a user is influential then there is a strong possibility that the information sent by him will be spread in the network far more than another, less influential, member. This can be used as a medium to make the underlying network more useful and cost effective.

In this paper, we present a methodology for ranking the influence of users in a speech-based network (SBN). More precisely, we address following problem statement: For given call data records (CDR) of a speech-based social network, extract social network graph (N_G), find the influence of users on other users in the network efficiently and use this influence as a deciding factor to achieve self-sustainability in the underlying network.

To the best of our knowledge, this is the first that tackles the problem of influence ranking in SBNs and addresses the following problems simultaneously:

- Constructing Social Network Graph N_G based on Call Data Records (CDR).
- Extracting features from (CDR) and normalizing them to calculate influence ranking.
- Computing influence probabilities based on features to apply influence maximization algorithms.
- Assigning edge labels to N_G for the calculation of betweenness centrality coefficient.
- Proposing time-efficient and data-driven techniques to utilize resources efficiently as compared to state-of-the-art baseline approaches.

The rest of the paper is organized as follows: In Section 2, related approaches are discussed. Section 3 presents our ranking techniques in detail. Experimental results are presented in Section 4. At the end, we make some concluding remarks.

2 RELATED WORK

Studying the influence of users on one another in social networks and using it for various applications, e.g., viral marketing, computational advertising etc, has gained immense attention in the past few decades. Various algorithms have been developed to calculate the influence of users in social networks. These algorithms range from influence maximization techniques where the objective is to find influential seed users to different centrality and distance based heuristics, where the influence of a single user is calculated.

Influence maximization has been used for viral marketing where the target is to reach maximum number of users through ”word-of-mouth” effect keeping in view the budget constraints. Domingos and Richardson (2001) were the first to study influence maximization as an algorithmic problem based on probabilistic methods. Kempe et al. (2003) first formulated it as a discrete optimization problem. Leskovec et al. (2007) proposed *Cost Effective Lazy Forward* (CELF), as a solution to reduce the running time of the *greedy algorithm* significantly. CELF is said to have approximately 700 times faster running time than the *greedy approach*. It has been tested on a real-time water distribution network, where the objective was to place the water quality sensor in the network such that the spread of contamination can be detected most efficiently. This approach was also tested on blog data, where the objective was to detect blogs that could be read to a good overall summary of a subject. Although the time required to compute the influential nodes has been significantly reduced, it is nonetheless, too slow to be used where quick results are required. Goyal et al. (2011) presented CELF++ to improve the efficiency of CELF by using sub modularity from 35 to 55 percent. Heidari et al. (2015) has proposed state machine greedy algorithm to speed up existing greedy approaches. Chen et al. (2009) provided degree discount heuristics for influence spread calculation and improved the classic greedy algorithms. They experimentally showed that selecting nodes with high degree increase the performance, however, it still lags behind if it is compared with influence spread produced by greedy algorithm. Degree discount heuristics matched the performance of greedy algorithm for Independent Cascade model and improved on other degree-based heuristics used for finding influential nodes. It totally outperformed any influence propagation model in terms of time efficiency while keeping the performance matchable in terms of spread calculation. Chen et al. (2010) addressed the issue of scalability for these algorithms by presenting PMIA. They introduced a feature that could be tuned to get a balance between time efficiency and influence spread. This feature was tested on many real time and synthetic data sets and significant improvement both in term of time efficiency and influence spread was shown. Wang et al. (2010) provided a dynamic programming based technique to find communities for influence maximization. Their algorithm is an order of magnitude faster than the standard greedy approach. Jung et al. (2012) developed a technique IRIE that combined both message passing based influence ranking and influence estimation methods. They found IRIE twice faster than PMIA algorithm.

While a significant amount of work has been carried out related to influence maximization and its applications in conventional networks, no work is being done on influence maximization in speech-based networks. Lin et al. (2014) have studied influence propagation in Polly through FIZZLE, DISPERSION and RENDEZVOUS patterns. Our targeted speech based networks are significantly different from conventional networks and influence maximization can have useful implementation for these networks as well.

3 METHODOLOGY

In this section, we describe our methodology to find influential users in a speech-based network with speech and push-button interfaces requiring neither literacy nor training. We propose a new technique called Influence Ranking (INFR) to find influential users in a SBN. We also propose a new method to assign influence probabilities in a SBN so as to apply influence maximization algorithms. We assign labels to social network graphs to find betweenness centrality coefficient of users of a SBN.

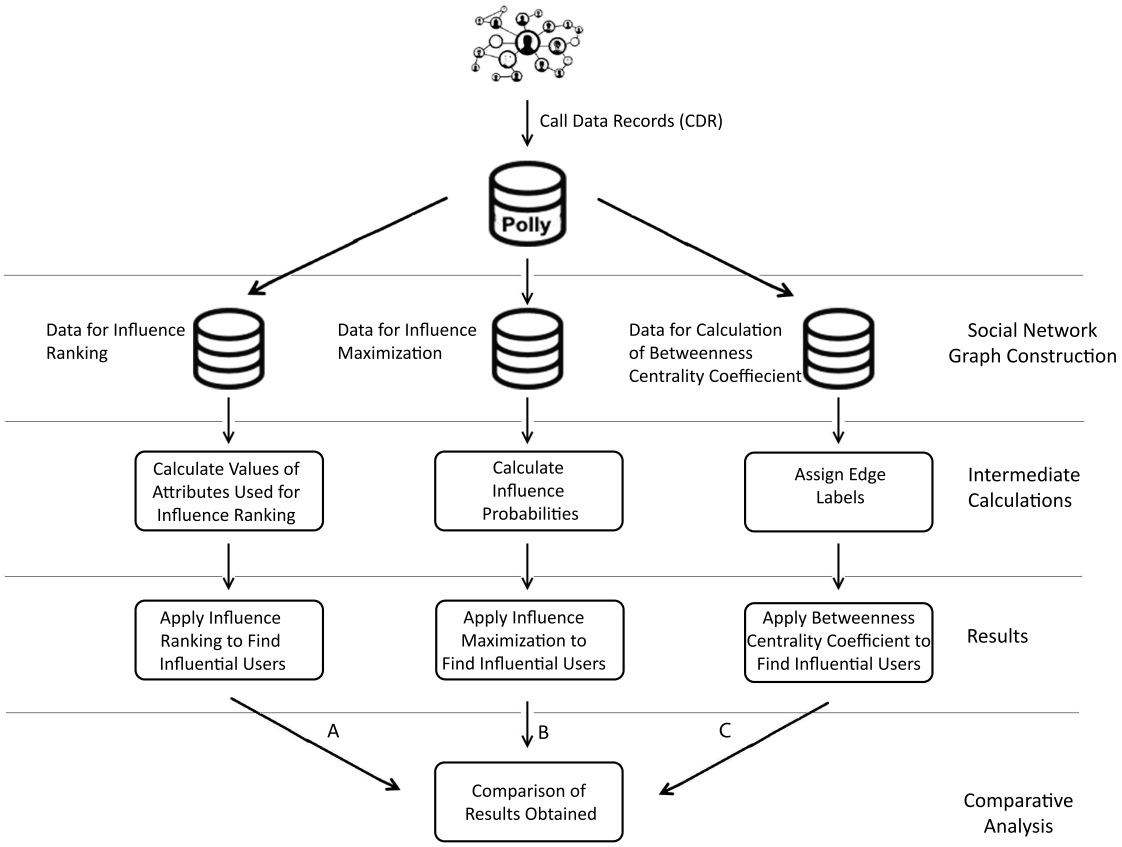

Figure 1: Process flow diagram for finding influential users in SBN

Figure 1 shows our methodology's process flow diagram. Part (A) of the figure describes the Influence Ranking technique, part (B) of the figure shows different Influence Ranking techniques adapted for SBN, and part (C) of the figure represents the calculation of betweenness centrality coefficient. Our methodology can be divided broadly into four layers. First we preprocess call data records (CDRs) for applying influence calculation techniques and constructing social network graph (N_G). Then we propose our feature based technique (INFR). For comparison purposes we apply influence calculation techniques and obtain results in the form of influential users. These techniques require preprocessed data so N_G is updated based on influence probabilities and edge labels.

Social Network Graph Construction (N_G)

We input Call Data Records (CDR) of speech-based networks for our research. (CDR) usually contain information about users, their interactions (Short Message Service (SMS) and audio files) and usage details of the resources of the networks. Social network graph (N_G) is constructed based on following attributes.

- Vertex: We assume a number of users, U=$\{u_i | i = 1, ..., z\}$ are located in a distributed fashion across a social network graph N_G. Each user u is identified as a vertex $u \in V$ of the graph based on her phone number.

- Directed Graph: In a directed graph, the edges between vertices are directed from one vertex to another. N_G is a directed graph, i.e., a vertex u is connected to other vertex v if user u has interacted with v.
- In-degree of vertex: In-degree of a vertex u is the total number of users who have initiated talk with or sent messages to u.
- Out-degree of vertex: Out-degree of a vertex u is calculated by counting the number of users to whom u has forwarded messages or calls.
- Disconnected components: Let a set of users V_1 represent one community of users based on their activities, region or lingual differences. N_G contains disconnected graph components if any users belonging to one set of vertices V_1 does not interact with any user belonging to other set of vertices V_2.

3.1 Influence Ranking

We present Influence Ranking (INFR), a technique for ranking the users according to their influence in a speech-based network. Algorithm 1 shows our influence ranking algorithm. It involves a few steps from feature extraction to the calculation of final score of each user. It takes as input (N_G) and (CDR) of the underlying network and passes them to function $ExtractFeatures()$. Relevant features required to identify influence, (e.g., number of total calls a user u has made, total ads forwarded, total ads re-forwarded, total number of messages that were re-forwarded) are extracted from *CDR* for each user of N_G. Values of these features are stored in F_{NG} for each user.

Algorithm 1 Influence Ranking (INFR)

Input: Social Network Graph N_G, Call Data Records (CDR)
Output: Influence score of each user $(INFS)$
$F_{NG} \leftarrow \varnothing$
$INFS \leftarrow \varnothing$
for each user u in social network graph N_G ***do***
    $F_{NG} \leftarrow ExtractFeatures(N_G, CDR)$
end for each
for each feature k in F_{NG} ***do***
    for each user *u* in social network graph N_G ***do***
        Normalize feature u_k using $u'_k = \frac{u_k - u_{kmin}}{u_{kmax} - u_{kmin}}$
    end for each
end for each
for each user *u* in social network graph N_G ***do***
    for each feature k in F_{NG} ***do***
        Update influence score $INFS(u) += u'_k$
    end for each
end for each
return *INFS*

Since values of different features in F_{NG} are in various ranges, they are normalized to a common range. Normalization helps to get an influence score that is not dominated by only few features. Value of k^{th} feature of user *u* is normalized after converting them into a range of [0,1] using following equation:

$$u'_k = \frac{u_k - u_{kmin}}{u_{kmax} - u_{kmin}} \tag{1}$$

u_{kmin} and u_{kmax} show minimum and maximum value of k^{th} feature for all users across N_G. Normalized value of u_k is stored in u'_k. After normalization we assign equal weights to all features and aggregate their values to calculate influence score *(INFS)* of a user using equation 2.

$$INFS(u) = \sum_{1}^{k} u'_k \tag{2}$$

Where u'_k is the range normalized value of the attributes. We rank the users based upon the values of influence score. A user with high value of influence score will be considered influential in network.

3.2 Applying Influence Maximization Techniques

Influence maximization is the process of finding a subset of users in a social network that, under a certain influence propagation model, can reach maximum number of users and influence them to perform a certain action. While a significant amount of work has been carried out related to influence maximization and its applications in conventional networks, almost no work is being done on influence maximization in speech-based networks. These networks are significantly different from conventional networks and influence maximization can have useful implementation for these networks as well. In this section, we apply influence maximization techniques on speech-based networks. We propose a method to assign influence probabilities in speech-based networks. Finally, we apply influence maximization techniques to find influential users on the N_G. Existing classical influence maximization techniques such as greedy general algorithm, degree heuristic, degree discount heuristic are discussed below.

3.2.1 Overview of Influence Propagation Models and Influence Maximization Techniques

Influence is propagated in the network according to a stochastic cascade model. We start with discussing influence propagation models and related influence maximization techniques.

1. Influence Propagation Models

Most of the influence maximization techniques require a graph N_G with edges labeled and an information propagation model that defines how users are influenced by other users in the network. In this section, we will discuss information propagation models that are an essential part of Influence maximization task. Following are two most common information propagation models that are used by different influence maximization techniques.

A. Independent Cascade Model (IC)

IC model finds the influence spread in the following way. Suppose S is the set of seed users and we want to calculate their influence. This process start at time t, at this time every seed user in S will have chance to influence its neighbours with probability equal to the edge label between them. If any new neighbour gets influenced then it is added to the set S. Now at time $t + 1$ the process is repeated again with the Set S that contains new elements. This process unfolds until no new user is added to the set S.

B. Linear Threshold Model

In linear threshold model, every node has an activation threshold t. If a fraction of neighbours of a node u with summation $\Sigma\ \rho(u, v)$ that is greater than threshold t, the user u gets activated as well. This process unfolds until there is no new node activated.

2. Influence Maximization Techniques

Influence maximization techniques are divided into two different categories, greedy and heuristic approaches. The greedy approaches have better influence spread, but lower scalability on large networks. The heuristic approaches are scalable and fast but not for all type of networks. Influence maximization techniques make use of graphs of social networks with edges already labeled. Edges represent connections between users of a social network, whereas edge labels are the influence probabilities of one user on another. We have used following greedy and heuristic as baseline influence maximization techniques:

A. Greedy Algorithm (GG)

Algorithm 2 presents the *general greedy solution* that guarantees the performance within $(1 - 1/e)$ of the optimal influence spread under both linear threshold model and independent cascade model, where e is the base of natural log.

Algorithm 2 The General Greedy Algorithm (GG)

Input: Social Network Graph (N_G)
Output: Seed Set (S) containing k selected users in Social Network Graph (N_G)
$S \leftarrow \varnothing$
$influence \leftarrow \varnothing$
R $\leftarrow$ 20000
for $i \leftarrow 1$ to k *do*
  for each user u in social network graph N_G ***do***
    for $j \leftarrow 1$ to *R* ***do***
      $influence[u] \leftarrow CalculateInfluence(S \cup \{u\})$
    end for
    $influence[u] = influence[u]/R$
  end for each
  $S \leftarrow S \cup \{argmax_{u \in V \setminus S}\{Influence[u]\}$
end for each
return S

The outer loop iterates k times to find most influential users. In each iteration most influential user is selected and added into seed set *S*. To find most influential user, an inner loop iterates over users of social network graph N_G and calculates their influence. For each user *u*, the influence spread of $S \cup u$ is estimated with *R* repeated simulations of function *CalculateInfluence()* which calculates influence propagation of users by Monte-Carlo simulation. It takes as input a set of users $S \cup u$ and returns number of users *u* that are activated upon activation of input set. After inner loop iteration, the user with maximal influence is added to *S*.

Time complexity of greedy algorithms is $O(knRm)$ where *k* is the number of influential users to select, *n* is the number of total users in N_G, *R* is number of times simulations will be run and *m* is the number of edges in N_G. Although *greedy approach* tends to give better results, it becomes quite difficult to calculate the desired seed users even using modern servers in a reasonable amount of time. A major chunk of time is spent on the running Monte Carlo simulations needed to calculate reasonably accurate influence spread. This serious drawback makes the greedy approach almost impossible to be used in any scenario where seed users are required to be computed in real time, e.g., finding priority users for call resource allocation on the fly in a speech-based network.

B. Degree Discount Heuristics (DDH)

Chen et al. (2009) provided degree discount heuristics for influence spread calculation. They also present some improvements to the classic greedy algorithm. Algorithm 3 presents the general idea as follows. Let *u* be a neighbor of user *v*. If *v* has been selected as a seed, then when selecting *u* as a new seed based on its degree, edge *uv* towards its degree should not be counted. Thus degree is discounted by one due to the presence of *v* in the seed set. Same discount is performed on degree of *u* for every neighbor of *u* that is already in seed set. For the IC model with a small propagation probability *p=0.1*, When *p* is small, indirect influence of *u* to multi-hop neighbors is ignored and direct influence of *u* to its immediate neighbors is considered, which makes degree discount calculation manageable. m_u is the number of neighbors of vertex *u* that are already selected as seeds.

Algorithm 3 DegreeDiscount: Degree discount algorithm for the Independent Cascade Model

Input: Social Network Graph $(N_G), k$
Output: Seed Set (S) containing *k* selected users in Social Network Graph (N_G)
$S \leftarrow \varnothing$
for each user u in social network graph N_G ***do***
    Compute its degree u_d
    $ud_d = u_d$
    $m_u = 0$
end for each
for $i \leftarrow 1$ to *k* ***do***
    Select *v* = $argmax_u$ { $ud_d \mid u \in V \backslash S$}
    $S = S \cup \{v\}$
    for each neighbor *u* of *v* and $u \in V \backslash S$ **do**
        $m_u = m_u + 1$
        $ud_d = u_d - 2m_u - (u_d - m_u)m_u p$
    end for each
end for each
return *S*

Chen et al. (2009) experimentally showed that selecting nodes with high degree increase the performance, however, it still lags behind if it is compared with influence spread produced by greedy algorithm. Degree discount heuristics matches the performance of greedy algorithm for Independent Cascade model and also improves on other degree-based heuristics used for finding influential nodes.

The major advantage degree discount heuristics provides is time efficiency. It totally outperforms any influence propagation model in terms of time efficiency while keeping the performance matchable in terms of spread calculation. N_G is directed due to the fact that there is two ways communication between users of the network. Figure 1 illustrates that N_G is used as an input to influence maximization techniques to find influential users.

3.2.2 Finding Influence Probabilities for the Graph of Speech-based Social Network

Most of the work on influence maximization ignored the problem of assigning edge probabilities which could have huge impact on the selection of seed users and their subsequent influence spread. Although (Goyal et al., 2010) tried to address the issue, however, they did not pay attention to devising methods for assigning probabilities in speech-based networks. This paper proposes a different way to assign edge probabilities that can be used according to the characteristics of the underlying speech-based social network. Both of these methods extract required data from the available call records of the users.

Although, these methods are not time-aware, however, they can be made time-aware by using time-stamp on the communication between the users.

Equation 3, $\rho(u, v)$ shows the influence probability of user u over user v, where S is the set of voice messages that user u has forwarded to user v and T is the set of voice messages that user u has forwarded to other users from S or listened for S. Voice messages can be any audio recording, e.g., songs, job ads etc.

$$\rho(u, v) = \frac{|T|}{|S|} \tag{3}$$

To elaborate the calculation of influence probability in 3, consider the following example:

Suppose a user u forwards 3 voice messages m_1, m_2, m_3, in the form of audio recording of job ads, to user v and after listening to these messages user v forwards m_2, m_3 to other users or listens from others then

$$\rho(u, v) = 2/3 = 0.67.$$

After extracting N_G and assigning influence probabilities, we use these parameters to apply influence maximization techniques. Influence maximization techniques give top k influential users as output, where k is the number of influential users to be extracted from the social network.

3.3 Influence Calculation Using Betweenness Centrality Coefficient

In our work, we use Betweenness Centrality coefficient for both directed and un-directed graph of the speech-based network as proposed by (Goh et al., 2003). For the directed graph of speech-based network, we propose method to find edge labels. These edge labels are further used to find Betweenness Centrality coefficient of users of the network. Figure 1(C) summarizes whole procedure.

3.3.1 Betweenness Centrality (BC)

Different centrality measures have been proposed to find influential users in social networks. Betweenness Centrality is also one of these measures. Betweenness Centrality coefficient can be measured for both directed and un-directed graphs of the social networks. Betweenness centrality is fraction of all shortest paths between all pair of nodes in a network that pass through a single node. A user with high between centrality is expected to have larger influence, however, there is an inherent assumption that influence will pass through the shortest path. This assumption might not be true in different cases when influence doesn't propagate through the shortest path. Betweenness centrality has a time complexity of $\Theta(V^3)$ where V is the number of nodes.

3.3.2 Determining Edge Labels

We assign edge labels to social network graph (N_G) based CDR and calculated both weighted betweeness centrality (BCW) and un-weighted betweeness centrality (BCU) of all the nodes in the network. To calculate BCW, we assign edge labels to N_G to make it suitable for finding weighted Betweenness Centrality.

$edgeweight = 1 - \rho(u, v)$

After calculating BC coefficient of each user, we have used it to rank users.

3.3.3 *Calculating Influential User Through Betweenness Centrality Coefficient*

After assigning edge labels to N_G , we calculate all pairs shortest path for both weighted and un-weighted graph. Then we used these shortest paths to calculate BC coefficient for each user. Finally influential users are selected with higher values of BC coefficient.

4 EXPERIMENTAL SETUP AND RESULTS

In this section, we explain the data set we have used for our experiments and the results that we have obtained running all experiments. Comparison of the time complexity of our technique with other influence calculation techniques is also given. To analyze the comparison of the time complexity of all techniques, run time analysis is also given.

4.1 Dataset Description

We have used dataset of Polly a telephone-based application, with a large influence network of 213196 users, 2522981 interactions, spanning around 700MB of log data. Based on the characteristic Call Data Records *(CDR)* of Polly, we have extracted a social network graph N_G, which is directed graph with every node labeled with an ID and every edge labeled with influence probabilities. Some of the key attributes of the N_G are listed in the Table 1.

Key Statistics about The Usage of Polly Based on Call Data Records *(CDR)*			
Number of users	213196	Number of calls made	2522981
Number of requests for calls	1051669	Number of requests for job delivery	33597
Number of times job ads listened	386199		
Key Characteristics of Social Network Graph (N_G)			
Number of edges in N_G	13694	Number of connected components	3272
Nodes in biggest connected component	2407	Edges in biggest connected component	3332
Diameter of the biggest connected component	10		

Table 1: Description of Call Data Records (CDR) *and Social Network Graph* (N_G)

To target requirements of speech based networks, *INFR* extract following features from N_G and *CDR*:

- Out-degree : Out-degree of a user u is the number of users to whom user v forwarded a message, e.g., a job ad etc.
- In-degree: In-degree of a users u is the number of users who have forwarded messages to user u.
- Total Calls by User: Number of total calls a user u has made to other users for delivery of messages.
- Total Ads Forwarded: Total number of ads a user u has forwarded to other users in the network.
- Total ads re-forwarded: Total number of messages that were re-forwarded from the users to whom user u has forwarded messages.

4.2 Discussion on Time Complexity and Run-time Analysis

In the section, we will discuss and compare the theoretical complexities of different methods for finding influential users. Table 2 lists the complexities of different methods to find influential users. k is the number of influential users to select. n is the number of nodes the network. m is the number of edges in the network. R is the number of Monte Carlo (MC) simulations required for greedy algorithm. As we can see from Table 2 that finding influential users through Betweenness Centrality coefficient and greedy algorithm for influence maximization has the highest complexity among all methods to find influential

Name of Algorithm	Complexity
General Greedy Algorithm	O($knRm$)
Degree Discount Algorithm	O($klog(n) + m$)
Degree Heuristic	O(m)
Random Heuristic	O(k)
Betweenness Centality	O($n^2log(n) + mn$)
Influence Ranking	O(kn)

Table 2: Complexities of different influence calculation techniques

users. Degree heuristics method has complexity equal to the number of edge in the graph and random heuristics has complexity equal to the number of nodes to select. Time complexities of degree discount algorithm and our Influence Ranking method has complexities that are comparable. For a graph to be fully connected, minimum number of edges m should be equal to $n-1$, however, we know that the value of m in a social network is several times greater than value of n. In this case, complexity of the degree discount algorithm would be greater than our Influence Ranking technique.

We have also recorded average running time of each technique on the dataset of Polly for finding top 50 influential users. We have performed our experiments on core i7 machine. We have used Sqlyog for relevant data extraction from SQL dump file. For the tasks related to graph theory, we have used Networkx that is a python library for such tasks. For influence maximization tasks, we have used standard implementation provided by different researcher for their algorithms. We have used Rapidminer for different data manipulation tasks. Figure 2 shows the average running time of each algorithm. Greedy algorithm and BC have highest average running time among all techniques to find influential users. For greedy algorithm, most of the running time is spent on running MC simulations to find influence spread in the network and for BC coefficient the time is spent on finding all pairs shortest path. Average running time of degree discount algorithm is almost comparable to our Influence Ranking methods, however, it will go up if the number of edges in the graph of the social network increases. Influence ranking is producing results closer to betweenness centrality and degree discount heuristic.

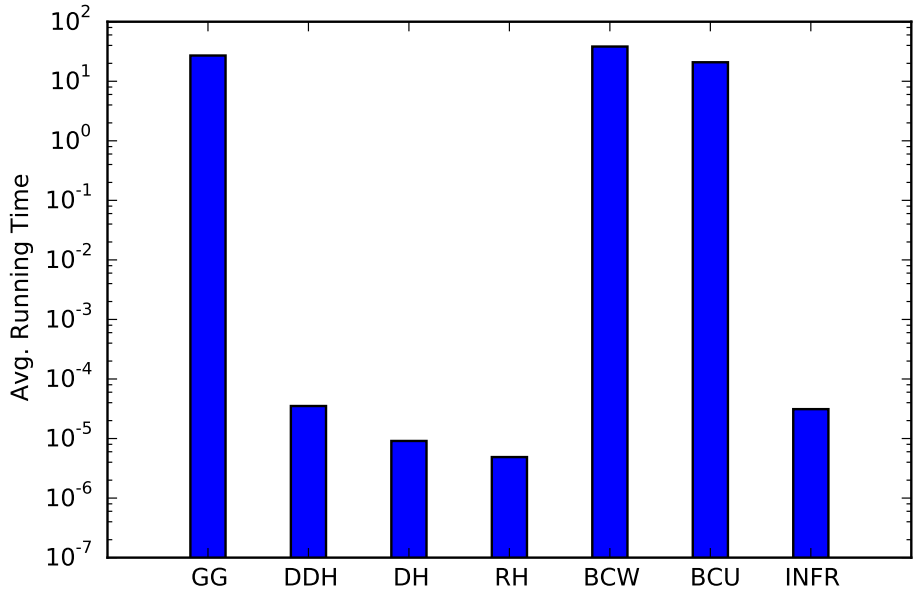

Figure 2: Average running time of algorithms to find influential nodes

	BCW	BCU	GG	DDH	DH	RH	INFR
INFR	16	14	0	20	27	1	50
RH	0	0	1	1	2	50	1
DH	13	11	1	30	50	2	27
DDH	14	12	2	50	30	1	20
GG	0	0	50	2	1	1	0
BCU	48	50	0	12	11	0	14
BCW	50	48	0	14	13	0	16

Table 3: Number of similar users found by different algorithms

4.3 Efficiency Analysis of Algorithms

Table 3 shows the comparison of results produced by all algorithms discussed in methodology section. However, Random Heuristic (RH) shows randomly selected users in N_G. We compare top 50 influential users selected by each algorithm and rank performance considering similar users. As we can see from Table 3 that Influence Ranking produces results similar to degree-based and degree discount heuristics. Influence Ranking has almost similar results for both weighted and un-weighted Betweenness Centrality coefficient. Influence Ranking is based on the fact that how much a user is spreading information in the network. Different heuristics-based techniques produce results that are similar to each other, however, almost none of these techniques have results that are more similar to general greedy algorithm for influence maximization. Even results obtained though general greedy algorithm are not similar to results obtained through degree-discount heuristics. Betweenness Centrality and Degree Discount algorithm for Influence Maximization produces results that are similar to each other.

5 CONCLUSION AND FUTURE WORK

We present an Influence Ranking technique to find influential users in speech-based networks (SBNs). Furthermore, we present methods for estimating influence probabilities between users thus enabling the application of in influence maximization algorithms to SBNs. Influence Ranking is significantly faster than any influence maximization algorithm and it produces better results than other heuristic-based techniques for finding influential users in the network. The calculated influence of users can be used as a decision-making factor while allocating resources of the network efficiently. Based on the results of our experiments on the dataset of Polly, we conclude that Influence Ranking is a suitable technique for situations where results are required at run-time and input data for the experiments is continuously changing.

There are several future directions to this research. One of the important direction is to improve the quality of influential users in terms of real-life influence spread. Although, Polly gained huge success in short period of time, however, the data generated during this period is quite meagre to apply techniques to find influential users. To calculate influence, the data about information transferred among the users in the network is required. We have this information in the form of number of job ads forwarded; however, this data is not enough to get any solid conclusion. A major limitation that we faced during experimentation was the sparsity of the available dataset. This makes the influence calculation task a bit unreliable. Another future direction is to test this technique on various other datasets of networks of similar type.

Acknowledgment

We would like to show our gratitude to Dr. Agha Ali Raza(Information Technology University, Pakistan) and Prof. Roni Rosenfeld(Carnegie Mellon University, Pittsburgh, USA) for their cooperation and providing us access to Polly's Call Data Records and log files.

References

Chen, W., Wang, C., and Wang, Y. (2010). Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th International Conference on Knowledge Discovery and Data Mining, SIGKDD*, pages 1029–1038. ACM.

Chen, W., Wang, Y., and Yang, S. (2009). Efficient influence maximization in social networks. In *Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining, SIGKDD*, pages 199–208. ACM.

Domingos, P. and Richardson, M. (2001). Mining the network value of customers. In *Proceedings of the seventh International Conference on Knowledge Discovery and Data Mining, SIGKDD*, pages 57–66. ACM.

Goh, K.-I., Oh, E., Kahng, B., and Kim, D. (2003). Betweenness centrality correlation in social networks. *Physical Review E*, 67(1):017101.

Goyal, A., Bonchi, F., and Lakshmanan, L. V. (2010). Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250. ACM.

Goyal, A., Lu, W., and Lakshmanan, L. V. (2011). Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th International Conference Companion on World Wide Web*, pages 47–48. ACM.

Heidari, M., Asadpour, M., and Faili, H. (2015). Smg: Fast scalable greedy algorithm for influence maximization in social networks. *Physica A: Statistical Mechanics and its Applications*, 420:124–133.

Jung, K., Heo, W., and Chen, W. (2012). Irie: Scalable and robust influence maximization in social networks. In *Proceedings of the 12th IEEE International Conference on Data Mining (ICDM)*, pages 918–923. IEEE.

Kempe, D., Kleinberg, J., and Tardos, É. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the ninth International Conference on Knowledge Discovery and Data Mining, SIGKDD*, pages 137–146. ACM.

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., and Glance, N. (2007). Cost-effective outbreak detection in networks. In *Proceedings of the 13th International Conference on Knowledge Discovery and Data Mining, SIGKDD*, pages 420–429. ACM.

Lin, Y., Raza, A. A., Lee, J.-Y., Koutra, D., Rosenfeld, R., and Faloutsos, C. (2014). Influence propagation: Patterns, model and a case study. In *Advances in Knowledge Discovery and Data Mining*, pages 386–397. Springer.

Medhi, I., Sagar, A., and Toyama, K. (2006). Text-free user interfaces for illiterate and semi-literate users. In *Proceedings of the International Conference on Information and Communication Technologies and Development, ICTD*, pages 72–82. IEEE.

Patel, N., Chittamuru, D., Jain, A., Dave, P., and Parikh, T. S. (2010). Avaaj otalo: A field study of an interactive voice forum for small farmers in rural india. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 733–742, New York, NY, USA. ACM.

Patel, N., Shah, K., Savani, K., Klemmer, S. R., Dave, P., and Parikh, T. S. (2012). Power to the peers: authority of source effects for a voice-based agricultural information service in rural india. In *Proceedings of the Fifth International Conference on Information and Communication Technologies and Development*, pages 169–178. ACM.

Plauché, M. and Nallasamy, U. (2007). Speech interfaces for equitable access to information technology. *Information Technologies & International Development*, 4(1):pp–69.

Raza, A. A., Pervaiz, M., Milo, C., Razaq, S., Alster, G., Sherwani, J., Saif, U., and Rosenfeld, R. (2012). Viral entertainment as a vehicle for disseminating speech-based services to low-literate users. In *Proceedings of the Fifth International Conference on Information and Communication Technologies and Development*, pages 350–359. ACM.

Raza, A. A., Ul Haq, F., Tariq, Z., Pervaiz, M., Razaq, S., Saif, U., and Rosenfeld, R. (2013). Job opportunities through entertainment: virally spread speech-based services for low-literate users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2803–2812. ACM.

Sherwani, J., Ali, N., Mirza, S., Fatma, A., Memon, Y., Karim, M., Tongia, R., and Rosenfeld, R. (2007). Healthline: Speech-based access to health information by low-literate users. In *Proceedings of the International Conference on Information and Communication Technologies and Development, ICTD*, pages 1–9. IEEE.

Veeraraghavan, R., Yasodhar, N., and Toyama, K. (2007). Warana unwired: Replacing pcs with mobile phones in a rural sugarcane cooperative. In *Proceedings of the International Conference on Information and Communication Technologies and Development, ICTD*, pages 1–10. IEEE.

Wang, Y., Cong, G., Song, G., and Xie, K. (2010). Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *Proceedings of the 16th International Conference on Knowledge Discovery and Data Mining, SIGKDD*, pages 1039–1048. ACM.