With a Little Help from My Friends

Arnab Nandi #, Stelios Paparizos, John C. Shafer, Rakesh Agrawal #The Ohio State University, 2015 Neil Avenue, Columbus, OH USA (work done while at Microsoft) Search Labs, Microsoft Research, 1065 La Avenida Street, Mountain View, CA USA arnab@cse.osu.edu, {steliosp, jshafer, rakesha}@microsoft.com

Abstract—A typical person has numerous online friends that, according to studies, the person often consults for opinions and advice. However, public broadcasting a question to all friends risks social capital when repeated too often, is not tolerant to topic sensitivity, and can result in no response, as the message is lost in a myriad of status updates. Direct messaging is more personal and avoids these pitfalls, but requires manual selection of friends to contact, which can be time consuming and challenging. A user may have difficulty guessing which of their numerous online friends can provide a high quality and timely response.

We demonstrate a working system that addresses these issues by returning an ordered subset of friends predicting (a) near-term availability, (b) willingness to respond and (c) topical knowledge, given a query. The combination of these three aspects are unique to our solution, and all are critical to the problem of obtaining timely and relevant responses. Our system acts as a decision aid – we give insight into why each friend was recommended and let the user decide whom to contact.

I. Introduction and Motivation

People often turn to their social networks, such as Facebook, Twitter, Skype, to fulfill their information needs [1]. The questions asked vary from subjective recommendations and opinions to objective factual information requests [2].

Although a search engine can be used to retrieve information, people often prefer the answers of their friends. Besides being personable, subjective and trusted, an answer from a friend can act as a time-saving shortcut to a lengthy search engine task. For example, suppose a person notices a wasp's nest in her house and would like to quickly eliminate it without using harsh chemicals. This problem is not serious enough to call a professional, and a search engine will return a ton of conflicting information, whereas a home remedy solution from a friend would suffice. In a similar vein, suppose that a person bought a new fitness device that measures how many steps they take during the day. Going through the manual and specifications is a good way to figure out how the device works, and a search engine can give useful information along the same lines. But when looking for creative ways to use this device, one could discover pleasantly surprising ideas by discussing with friends. We cover this scenario in more detail as part of our demonstration description in Section II. There are many similar tasks where an opinion or subjective advice would be preferred over impersonal information that a search engine returns.

Smartphones and tablets have become ubiquitous as they are easy to use and easy to carry around almost anywhere. These devices all have a GPS sensor and internet connection, which means they also provide access to a user's online social

network. As a result, there are new mobile scenarios that become possible. For example, a person is visiting Brisbane for a conference and decides to stroll around the town. As dinner time approaches, he decides to find a good place to eat that also offers great dessert choices. Given the mobile device, one could look for restaurants in a search engine or visit specialized tools like Yelp. However, not all tasks fit the specialized tools. Yelp does not rate the dessert part of a restaurant separately, and even if they did, the user would still have to read the reviews and calibrate for the ones that have a similar taste in desserts. On the flip side, a quick call or message to the right friend could provide an efficient and trusted solution to this task.

When reaching out to friends in a social network, people often publicly post questions to the entire network At first this seems to allow a user to reach a broader audience, thus increasing the chances of getting a good and timely response. However, only a low percentage of online questions receive a response [1]. Such an undirected request is easily lost amongst many other postings, or can lead to content-free responses by friends who are eager or feel compelled to respond, yet have little time to compose more than a cursory message. Furthermore, public broadcasting runs the danger of irritating friends when done too often on off-topic questions; and it is not suitable for some people if the topic is sensitive (e.g., advice on a medical condition).

Direct messaging avoids those pitfalls, but presents its own set of problems. The average user has hundreds of online friends, so it is difficult and time-consuming to figure out who would be best to engage for relevant and timely responses. Users may not have good insight into who is knowledgeable about a topic, especially if they have never discussed this particular topic in the past. Furthermore, even if someone is an expert, there is always the problem of catching them when they are available to respond or eager to interact.

In this paper, we present a system that aid its users in finding the ranked subset of friends whom they can interact with on a given topic. We mine existing social network data focusing on a novel set of criteria: availability, willingness and knowledge. Our system captures (a) how likely it is that a friend is online in the near future based on past activity patterns, (b) the likelihood that a friend will respond based on the strength and nature of the interpersonal connection and past interaction behavior and (c) a friend's knowledge and expertise on a topic and their potential for providing an informed response based on the past message content. We do

not initiate any conversations on the user's behalf, but provide insight and guidance to allow for informed choices on whom to engage.

While finding friends for a discussion may not have been studied in depth yet, expert finding [3], [4], [5], for question answering in the enterprise space, has a lot of similarities to this problem setting. Such approaches establish expertise by utilizing implicit information based on emails sent to mailing lists in conjunction with explicit information provided by the users in a company directory. Enterprise solutions for the most part use calendar or internal messaging tools to establish who is currently online to respond and are not usually limited to friends, but rather includes all the employees. In that sense, the knowledge is established in a similar way as our work, but with different data sources. Where we differentiate is in trying to predict a friend's availability and willingness to respond in a time-bounded ceiling.

The now defunct Aardvark [6] was an approach to this problem for the web. The system would consider both the social graph and message content, initiating 1:1 conversations on behalf of the user with prospective responders. They placed a lot of emphasis on finding who is the best person to answer by considering textual topics and workload of contributors, aspects similar to our work. However, they relied on users signing up for their service to be available as responders, since their system reached out directly to responders. Instead, we take advantage of existing networks and aid the user decision in reaching out to their friends on top of what infrastructure already exists, without requiring new memberships. Furthermore, the concept of mobile time-bound interactions and using location were not studied in depth by the authors. So the notion of predicting availability and willingness in the mobile-social era were not applicable in that system.

Last but not least, we are sensitive to the fact that mining friend information could potentially raise privacy concerns. It is a fundamental principle in our system to respect the privacy rules of the underlying social network and not allow access to restricted information. Our goal is to provide value by only aggregating friends' data already visible to the user.

In the rest of the paper, we present in Section II, a demonstration scenario that runs on our fully-implemented system. We discuss our system design choices, data model and algorithmic sketches in Section III.

II. THE DEMONSTRATION SCENARIO

As a motivating scenario, consider a user, Jess, who has just come across a good deal on a FitBit, a wearable sensor device for measuring exercise activity. She is tempted to buy one, but knowing very little about it, she would like to see if her friends can offer her any perspective. While still in the store, she pulls out her phone and enters the query "fitbit" into our app. She is presented with a ranked list of friends that the system thinks would be most helpful (Figure 1a).

The initial ranking is based on a combination of the 3 key factors: *availability*, *willingness* and *knowledge*. We visualize the individual scores for each returned friend via colored

bars, where longer bars indicate a higher score (e.g., more likely to be available). This list can also be resorted along the component scores by changing the sort selector.

Jess selects the suggested friend Maarten, who is ranked fairly high. This yields further detail as to how the system scored this friend against the three key criteria. The goal is to explain to users why a person was suggested, so that they can make an informed decision about whom to contact.

The first screen (Figure 1b) provides detail about the friend's expected availability based on past social-network activity. A series of three bar graphs illustrates the friend's predicted activity based on day of the week, hour of the day, and current location. For the first two graphs, the bars corresponding to the current day and time are highlighted in red. For the third graph, the highlighted bar is the one corresponding to the friend's predicted location. Taken together, this friend is likely to be available if Jess wishes to contact him, although hourly activity suggest he will be off-line a bit later on.

Users can then pan over to see how the system estimates the friend's Willingness (Figure 1c). As the system points out, Jess knows Maarten, as they have had direct interactions in the past. If he had not been an immediate friend, the system can show through which other friends they are connected. From the point of view of both the user and the suggested friend, we also show the fraction of all interactions that were directed at the other person. The stronger the connection and the greater the number of interactions, the greater the probability that a friend will be willing to respond. Physical distance also plays into responsiveness, and we show this both numerically, as well as plotting both the user and the suggested friend on a map. While perhaps not as relevant to this scenario, close proximity may even encourage face-to-face interactions (e.g., for a question on finding a good cafe)

On the third screen (Figure 1d), we present the user with evidence as to how knowledgeable the suggested friend is about the topic at hand. In addition to reporting the overall knowledge score, we show recent posts from the suggested friend that are relevant to the query. This not only helps to explain why the suggested friend was deemed knowledgable, but may also give deeper insight into what sort of knowledge and perspective they may have. In this case, the friend not only has first-hand experience with the FitBit, but has recently tweeted tips on how to best make use of the device.

The last screen that the user sees (not shown) is a contact page that includes any available profile information for the suggested friend. From here, the user can initiate contact with the friend, whether through phone, email or text. We provide the user with multiple means of contact, as it may depend not only on the urgency of the question, but also the friend's predicted availability. For example, if the friend is currently not active, but likely to be online later, the user may decide to send an email rather than text or call. For our user Jess, she may do better to call Maarten immediately, as he is expected to go off-line in a few hours and may not see an email before then. We do not automatically contact friends on a user's behalf, we suggests friends and provide supporting detail so





willingness kno

This person is not too far from you (82mi

Duisburg

CONNECTIONS

INTERACTIONS

PROXIMITY

LOCATION

Rotterdam

You with this user: 0.0153

This user with you: 0.0147



(d) Knowledge

users can decide for themselves whom best to contact.

III. OUR SYSTEM DESIGN CHOICES

For our demonstration, we obtained three months of public social data from the Twitter firehose. This data was filtered based on the availability of location information, from either GPS-stamped tweets or a user's self-identified location, and further restricted to users having interactions with at least 10 and at most 1000 other Twitter users. The resulting data set consists of 4.2 million users with 81 million interaction edges. An interaction is either a reply, retweet or a message containing a user handle and it is directional. Furthermore, we only consider users to be friends when they have exchanged at least one message with each other in both directions.

We built a scalable offline pipeline (shown in Figure 2) that mines the social data and computes the scoring probabilities and indexes. We created a Windows Phone 7 front-end app (screenshots shown in Figure 1) that we use to demonstrate our system and illustrate how users find friends for a given information need.

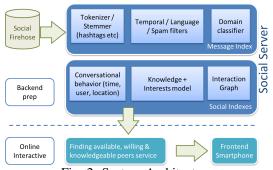


Fig. 2: System Architecture

A. Computing Utility Scores

The main insight in finding friends is encapsulated in our work on availability, willingness and knowledge. For this demonstration paper, we make the simplifying assumption of independence to focus on communicating our key features. More complex models can be applicable but their advantages and disadvantages is outside the scope of this paper.

Definition 1 (Availability): Availability $A(\phi|t)$ is the likelihood that a friend ϕ will be active in the social network and able to interact in the near future, given current time t.

In most social networks, availability is represented as a status, where a friend is either available to interact (online) or can only be reached via indirect messaging (offline). However, such an approach has its set of issues. In some cases, a friend logs onto a website like Twitter periodically to check their messages, making it challenging to catch them because the intersection of common time could be minimal. In other cases, a friend is logged-in for a long period of time on the system, like Skype, and is shown as available. But, the online status in this case is not necessarily representative of the real life situation, as they might be away from the computer, occupied with other tasks or on a mobile device that constantly appears online. Therefore, it is important to predict when a friend will actually be available and have time to interact.

The availability $A(\phi|t)$ computation for each friend ϕ relies on mining past activity patterns broken into three subcomputations (a) hour-of-day (hod), (b) day-of-week (dow) and (c) friend location (loc). Computation for (a) and (b) are intuitively similar. For each friend ϕ , compute the distribution of the number of messages m on each hour or day bucket. The current hour and day indicates which bucket to select in the histogram. To figure out the location probability, we first compute for each friend ϕ , the distribution of messages sent from each location ever. We do not care to find the actual location, but rather the distribution. Then we use the friend's most likely location on that hour / day of the week to find the corresponding bucket in the location histogram. So, assuming independence for simplicity, availability is: $A(\phi|t) = hod(\phi|t) * dow(\phi|t) * loc(\phi|t)$

Availability can potentially be zero which means the friend should not be contacted at all. However, typically it has a continuity from more to less available.

Definition 2 (Willingness): Willingness $W(\phi|u,q,t,l)$ is the likelihood a friend ϕ of a user u will respond to a message request q, given the user's current time t and location l.

Even if a friend knows about a topic or is available, it does not guarantee that they will be eager to discuss this topic with the user. Sometimes they are not that interested in a conversation or they have been 'annoyed' by multiple past requests and would rather avoid the user. Furthermore, perhaps they are not that close friends with the user or have

different views and avoid interaction. Social affinity can be thought of as a good approximation to capture such eagerness. Furthermore, we have found from analyzing past interactions that physical location proximity also increases the likelihood users will interact with each other. We attribute this to physical proximity being another form of social affinity, as users who live close together interact more frequently in the real world as well as online.

The willingness $W(\phi|u,q,t,l)$ computation relies on social affinity as well as geospatial proximity. Our experimentation has shown that the degree of reciprocal communication amongst friends when considered as a fraction of messages sent and received is a good indicator for the likelihood people will interact on a topic. Social affinity is then measured as $soc(\phi|u) = \frac{|m_{u \Rightarrow \phi}|}{|m_u|} * \frac{|m_{\phi \Rightarrow u}|}{|m_{\phi}|}$. This is extendable to cover query topics qt by only considering the messages exchanged on the same topic as the query q and thus computing $soc(\phi|u,q)$. Furthermore, we found experimentally that physical proximity tends to act as a proxy for face-to-face interactions, further strengthening the social affinity and also providing a form of homophily. We compute the distribution of the distances amongst all messages sent and received and use this to estimate the friend distance. So $loc(\phi|u,t,l) = \frac{|l-loc(hod(t),dow(t))|}{|m_u|}$. Again, the overall willingness computation assumes independence for simplicity: $W(\phi|u,q,t,l) = soc(\phi|u,q) * loc(\phi|u,t,l)$

Overall, willingness is generally above zero, as, in a social environment, friends always find some opportunity to interact.

Definition 3 (Knowledge): Topical expertise or knowledge $K(\phi|q)$ is the likelihood that a friend ϕ will produce a high quality response, given a message request q.

In a social environment, it is very common that eager friends respond with an 'opinion'. However, the value of such a response can only be useful if the friend knows what they are talking about. It is also very rare for a user to know all aspects of a friend's life. Therefore it is essential to compute this for all friends. We are not looking for experts or authorities on a given topic – who knows more is a very subjective notion. We focus on friends that are familiar with a topic so the user can interact with them.

To capture knowledge, we rely mostly on message content, as we found the bio and interest information in the user profile to be incomplete. We concatenate all message and profile content into one document for each user and index them in an inverted text index. We use *Okapi BM25* [7] to score them for each query. Scores are then normalized to probabilities and are comparable across friends for the same query. Text ranking and scoring is well-understood in the IR community, and a detailed discussion is outside the scope of this paper.

Alternatively, knowledge can also be mapped to a set of topics $qt \in T$. This can be specified in a profile or learned. Knowledge then becomes: $K(\phi|q) = \sum_{qt \in T} p(\phi|qt)p(qt|q)$, given a classifier mapping a query q to all topics qt. We chose to go with the simpler and more flexible definition that allows for alternative implementations to compute topical expertise.

Query-topic knowledge can be zero, effectively filtering out friends unable to contribute something useful.

Definition 4 (Friend Utility): Given a user u, current time t, location l and query message q, the utility of a friend ϕ is: $U\phi(u,q,t,l) = f(A(\phi|t),W(\phi|u,q,t,l),K(\phi|q))$

Friend utility is a combination of the availability, willingness and knowledge main features described above. Continuing with the simplified independence assumption we compute: $U\phi(u,q,t,l) = A(\phi|t) * W(\phi|u,q,t,l) * K(\phi|q)$

Problem 1 (Friend Selection): Given a user u, time t, location l, query q and parameter k>0, return an ordering $L_k(\Phi)$ of friends $\phi_i\in\Phi$, such that $\sum_i U\phi_i(u,q,t,l)$ is maximized.

Our goal is to find a subset of friends filtered and ordered by this utility function, such that we maximize the likelihood the user will receive a timely and good quality response. We note that the output can be the empty set, as no friend may be available or have the expertise to respond. Furthermore, although we mention friends, it is possible to extend the same formulation to friends-of-friends or n-degree of separation in a social network. This extension can be used where data is available and visible to the user, while also respecting the privacy policies of the social network and its members. Finally, although this is an optimization problem, an at-most-k solution can be used instead to approximate the same behavior while still providing a good user experience.

IV. CONCLUSION AND FUTURE WORK

We presented a system that returns a filtered and ordered subset of a user's friends, by predicting availability, willingness and knowledge as proxies for estimating who is a good candidate to provide a judicious and time bound response to a query. The importance of these three dimensions is an insight that is unique to our system, as within a social context, expert-finding alone is insufficient. To truly evaluate the utility of such a system, an extended user study (or feedback from the application deployment) is needed to measure the satisfaction of the system in real-world scenarios. This is outside the scope of this demonstration paper; we leave it as future work, along with evaluating the various methods used by the individual components in our system.

REFERENCES

- S. A. Paul, L. Hong, and E. H. Chi, "Is twitter a good place for asking questions? a characterization study," in *Proc. ICWSM Conf.*, 2011.
- [2] M. R. Morris, J. Teevan, and K. Panovich, "What do people ask their social networks, and why?: a survey study of status message q&a behavior," in *Proc. CHI Conf.*, 2010, pp. 1739–1748.
- [3] C.-Y. Lin, N. Cao, S. Liu, S. Papadimitriou, J. Sun, and X. Yan, "Smallblue: Social network analysis for expertise search and collective intelligence," in *ICDE*, 2009, pp. 1483–1486.
- [4] R. W. White, M. Richardson, and Y. Liu, "Effects of community size and contact rate in synchronous social q&a," in *Proc. CHI Conf.*, 2011, pp. 2837–2846.
- [5] D. Yimam-Seid and A. Kobsa, "Expert-finding systems for organizations: Problem and domain analysis and the demoir approach," *Journal of Org. Computing and Elec. Commerce*, vol. 13, no. 1, pp. 1–24, 2003.
- [6] D. Horowitz and S. D. Kamvar, "The anatomy of a large-scale social search engine," in *Proc. WWW Conf.*, 2010, pp. 431–440.
- [7] K. Jones, S. Walker, and S. Robertson, "A probabilistic model of information retrieval: development and comparative experiments," *Information Processing and Management*, vol. 36, no. 6, pp. 779–840, 2000.