Proceedings of the Tenth International AAAI Conference on
Web and Social Media (ICWSM 2016)

Who Will Respond to Your Requests for Instant Trouble-Shooting?

Kai-Hsiang Hsu*, Cheng-Te Li†, Chien-Lin Tseng‡
*Dept. of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan
†Research Center for IT Innovation, Academia Sinica, Taipei, Taiwan
‡Codementor Inc., CA, USA

Abstract

Instant mentoring services are novel social media, in which mentees can input expertise requests and wait for accepting some expert mentors who are willing to tackle the requests in an instant and one-by-one manner. While mentee's satisfaction of being mentored is determined by the matched mentor, this paper aims at analyzing and finding which mentors will respond to the given request raised by a mentee in Codementor, which is one of the popular instant mentoring services. We formulate the *Mentor Willingness Ranking* (MWR) problem. MWR is to understand whether a mentor is willing to tackle a request. We propose to deal with the task by generating a ranked list of mentors such that those mentors who are really willing to tackle the request are as many as possible. We develop three categories of features, Availability, Capability, and Activity, to model the willingness of a mentor dealing with the request. Results of analysis show the effectiveness of these features, and encourage develop learning-based method to accurately identify the willing mentors.

1 Introduction

With the maturity of Web technologies, instant mentoring services, such as Codementor[1], HackHands[2], and AirPair[3], had been established one after another recently and getting more and more attentions. These instant mentoring platforms are mainly featured by connecting programmers and coders who need help and people who can help them. In these platforms, experts with highly specialized technology/programming knowledge can contribute themselves by solving the questions raised by those in need of specialized help. People who provide help and are satisfied by those raised the questions can also get reward (e.g. earn money). Also owing to the nature of one-on-one or one-on-many instant mentoring, these services are able to not only save the time duration of trouble-shooting, which is especially useful for solving urgent issues and can speed up the development process, but also enable other advanced functions such as pair programming, code review, and mock interviews.

[1]https://www.codementor.io/

[2]https://hackhands.com/

[3]https://www.airpair.com/

The key point of ensure the quality of instant mentoring is how to recommend the list of candidate mentors so that both mentee and mentor can feel happy. The recommendation should satisfy several requirements. For the part of mentors, these candidate mentors are required to not only being capable of dealing with the request raised by the mentee, but also being willing to tackle the request. The willingness can be determined by the recent mentoring activities and the time zone of each mentor. For the part of mentees, they do not want to wait too long since they may encounter instant or urgent problems, and they may desire not to spend too much money while expecting the recommended mentors are experienced and have good records. In other words, the mentee tends to feel satisfied if the mentor has good reputation, provides cheap charging fee, and responds fast.

This work aims at analyzing the feasibility of mentor recommendation for the mentee-input request by considering the aforementioned requirements supposed to be satisfied by both mentor and mentee sides. We simplify the mentor recommendation problem as: given a set of skills that describe the mentee's request, generating a ranked list of k mentors such that the mentors who are **willing** to deal with the request are placed as high as possible in the list. We term such task as the *Mentor Willingness Ranking* problem. To deal with such task, we need to consider the skills, the time zone, and the recent mentoring activities of each mentor. A mentor whose skills cover more of the skills of the request has higher potential to be capable of tackling the request. If the time zone of the mentor is the same as the time zone of the mentee, the mentor tends to express "willing" and the mentee's waiting time can be shortened. In addition, a mentor who tackle requests frequently and recently is believed to respond to the request within a short time period. The rating score and the charging fee of a mentor are also key factors.

In the literatures, the research lines *Community Question Answering* (CQA) and *Reciprocal Recommendation* (ReRec) are relevant to this paper. The main topics in CQA are expert finding and identifying the best answers. The former find experts that can answer a given question (Bouguessa et al. 2008; Riahi et al. 2012; Yang et al. 2014) while the latter is to predict which expert can best answer the given question or which answer can have the highest rating (Tian et al. 2013; Gkotsis et al 2014). ReRec is to recommending people to people (Pizzato et al. 2010; Li and Li

Table 1: Data statistics of the used Codementor data.

#requests = 11,196	#mentors = 3,428	#mentees = 70,712
#willingness = 39,641	#matches = 4,165	#tags = 9,549
#willingness per mentor = 19.65		#tags per mentor = 17.66
#willingness per request = 3.54		#tags per request = 3.17

2012; Kutty et al. 2014), in which the preferences of both of the participated parties (e.g. male-female matching) are supposed to be satisfied. However, both CQA and ReRec consider neither the willingness of the experts (or people in one party) to respond to the requests to approve the recommendation, which is what we tackle in this paper.

The instant mentoring service we target at is Codementor. We will present the data statistics for Codementor's requests, mentors, mentees, and matched mentor-mentee pairs. Since MWR is to analyze the willingness of mentors by generating a ranked list of mentors for the given request, we propose to characterize mentors' willingness to tackle requests. We develop three categories of features, including *Availability*, *Capability*, *Activity*, and *Proximity* to model the willingness of a mentor dealing with the request. These features are extracted based on the request, the profile of a mentor, and the historical activities of a mentor. In the analysis, we will use each proposed feature score of generate the ranking of mentors. In addition, we also exhibit whether each feature is able to distinguish "willing" and "unwilling" mentors.

2 Codementor Data and Analysis

We use a collection of data from one of the largest instant mentoring services, Codementor, during the time period from April 19, 2015 to November 15, 2015. Before describing the insights of the data, we need to point out the current Codementor's mechanism for recommending mentors so that the input requests can be forwarded to them. Codementor hires an programming expert to manually generate a ranked list of 100 mentors who are most likely being willing to deal with the request and being better satisfied by the mentee. The manual generation process is based on the tags, time zones, rating, charging fee, and historical willingness and mentoring records of mentors. Therefore, we have the list of *manually-recommended* mentors for each request. By running with such lists, mentors' willingness (to requests) records can be collected. We use these records for this study. The data statistics of our used data is shown in Table 1, in which #willingness refers to the number of expressing "willing" by mentors to requests, and #matches means the number of successful matched mentor-mentee pairs.

3 Mentor Willingness Ranking in Codementor

3.1 Problem Formulation

We first provide the formulation of the **Mentor Willingness Ranking** (MWR) problem. Given a request q that is input by a certain mentee, a set of skills L_q for request q, and a collection of mentors, in which each mentor m contains a set of skills L_m, and the number of k mentors to be recommended, the goal is to return a ranked list of k mentors such that the number of recommended mentors (in the list) who are willing to tackle the request is as many as possible. That said, MWR is to find whether a mentor is willing to provide help for the request.

Since what we have is the historical willingness records for each mentor, we aim at using these records to develop features that can effectively characterize the willingness of each mentor for a request. When a new request is input, we can use the feature score to generate the ranked list of mentors such that the recommended mentors are truly willing to deal with the request. In the following, we will devise a series of features to characterize the possibility of a mentor who is willing to tackle the request. Then we conduct some empirical studies to show the effectiveness of each feature.

3.2 Predictive Features

We propose three categories of predictive features to recommend mentors: availability, capability, and activity. The general idea is to estimate the willingness that a mentor wants to tackle the request. All of these features are extracted based on the request (including the time of request, the required skills, and the budget), the profile of each mentor (including skills, time zone, rating, and charging fee), and the historical activities of each mentor (including delivered "willing" to which of past requests at what time, and successfully matched with which of past requests at what time).

Availability features aims at quantifying the availability of mentors in terms of time, and the availability of the mentee in terms of budget. Higher availability indicates higher tendency that a successful match happens between a mentor and the mentee.

- *Time Zone Closeness.* Due to the daily routine and biological clock of human beings, if the time zone of the mentor is far away from the time zone of the mentee, the willingness of the mentor to tackle the request is supposed to be lower. We propose to measure the closeness between time zones of a mentor m and the request q as

$$TZClose(m,q) = 1 - \frac{\min(\Delta_T(m,q), 24 - \Delta_T(m,q))}{12}, \tag{1}$$

  where $\Delta_T(m,q) = m.ltz - q.ltz$, and $m.ltz$ is the local time zone of m. Higher $TZClose(m,q)$ values indicates higher availability and the mentor m has higher possibility to tackle the request q.

- *Online Availability.* A direct manner to find the availability of mentor q is whether he/she is *online* at the time that the request q gets posted. As Codementor allows each mentor to indicate whether he/she is online now, we devise a binary feature considering both time zone and online status to understand the availability of a mentor.

$$OA(m,q) = \begin{cases} 1, & \text{if } m.status = \text{“}online\text{”} \textbf{ and } m.ltz = q.ltz \\ 0, & otherwise \end{cases} \tag{2}$$

- *Budget.* We also need to measure the availability in terms of budget of the mentee who proposes the request. If the charging fee of mentor m is close to the budget of request

q, then m tends to be available to q. We use the difference between m's charging fee and q's budget to define the feature.

$$BA(m,q) = \begin{cases} 1, & \text{if } \Delta_B(m,q) \leq 1 \\ \frac{1}{\sqrt{\Delta_B(m,q)}}, & otherwise \end{cases}, \quad (3)$$

where $\Delta_B(m,q) = m.chargingFee - q.budget$.

Capability features are proposed to quantify whether or not the mentor q is qualified to deal with the request q considering m's expertise, rating by past mentees, and past mentoring sessions. It is believed that the capability is the most important factor affecting the mentee to select which of the recommended mentors.

- *Skill Matching.* The most intuitive way to estimate the capability of a mentor is to examine the degree of matching between the skills of the mentor and the required skills of the request. We consider two factors for skill matching. The first is skill coverage. If the skill set of a mentor covers more required skills of the request, such mentor is more eligible. The second is skill uniqueness. Some skills are common while some are unique. A mentor possesses an unique required skill can better deal with the request and satisfy the need of the mentee, compared with a common required skill. Let the skill set of mentor m as L_m, and the required skill set of the request q as L_q. We define the feature of skill matching as:

$$SkillMatch(m,q) = \frac{\sum_{l \in L_m \cap L_q} Rarity(l)}{\sum_{l \in L_q} Rarity(l)}, \quad (4)$$

where $Rarity(l)$ is the rarity of skill l, defined by $Rarity(l) = \frac{1}{\sqrt{freq(l)+1}}$, and $freq(l)$ is the number of mentors who possess skill l.

- *Rating.* The quality of the mentor can be directly reflected by his/her rating score, which is rated by mentess based on past mentoring sessions. We use the technique of *Bayesian Average* on the rating records of mentor m. Let a certain rating record score i of mentor m as $r_m(i)$, and let the total number of rating records of m as c_m. We define the *Bayesian Average Rating* (BAR) of m as:

$$BAR(m) = \frac{C \times \bar{r} + \sum_{i=1}^{c_m} r_m(i)}{C + c_m}, \quad (5)$$

where C is the total number of rating records among all the mentors, and $\bar{r}$ is the average values of rating scores among all the mentors.

- *Session Length.* When a mentee selects a recommended mentors, an instant mentoring session will be created. We consider the mentoring session to approximate the capability of a mentor. If a mentor either frequently establishes sessions with mentees, such mentor can be regarded as an experienced one. Also if the time duration of a mentoring session is longer, the mentor in such session can deliver more knowledge to the mentee, and the mentor is more trustful because the mentee is willing to pay more to have longer discussion with the mentor. The total time duration of mentoring sessions as is treated as the feature, given by $SessionLen(m) = \sum_{b \in B_m} TD(b)$, where B_m is the set of sessions of mentor m, and $TD(b)$ is the time duration of session b.

Activity features are to capture the mentoring activities of a mentor. A mentor who either frequently express "willing" to tackle requests or frequently being accepted by mentees is considered as an enthusiastic one, and has higher potential to provide instant trouble-shooting. We quantify the mentoring activities from two temporal aspects: instant time and recent time. Instant-time activities of mentor m refer to the historical interactions between m and mentees *within one certain time difference (e.g. one hour) before and after a request gets posted*, while recent-time ones represent the recent interactions between m and mentees *in the past week*. Assume a request q is posted at timestamp T_q (e.g. 201512241002) whose time of day is denoted by t_q (e.g. 1002), let $N_{ins}(m,q)$ and $N'_{ins}(m,q)$ be respectively the times of expressing "willing" and the times of establishing mentoring sessions by mentor m to any historical requests within time period $t_q - \Delta_A \leq t \leq t_q + \Delta_A$, where Δ_A is the instant time difference and empirically set as 60 minutes in this study. In addition, let $N_{rec}(m,q)$ and $N'_{rec}(m,q)$ be respectively the times of expressing "willing" and the times of establishing mentoring sessions by mentor m to any historical requests within time period $T_q - \Delta_R \leq T \leq T_q$, where Δ_R is the recent time difference and empirically set as 168 hours in this study. We define the following four activity features.

- *Instant Willingness.* A mentor m delivering more instant-time "willing" around t_q has higher possibility to tackle request q within a short time period. We define the instant-willingness feature as: $WI(m,q) = \frac{N_{ins}(m,q)}{\sum_{u \in M \setminus m} N_{ins}(u,q)}$, where M is the set of all the mentors in the data.
- *Instant Sessions.* A mentor m accepted by more mentees around T_q tends to be accepted to deal with new request q within a short time period. The feature is defined as: $SI(m,q) = \frac{N'_{ins}(m,q)}{\sum_{u \in M \setminus m} N'_{ins}(u,q)}$
- *Recent Willingness.* If mentor m very actively expresses "willing" to requests in recent days before T_q, he/she has higher potential to tackle request q. We define the recent-willingness feature as: $WR(m,q) = \frac{N_{rec}(m,q)}{|Q_T|}$, where Q_T is the set of all the requests posted within $T_q - \Delta_R \leq T \leq T_q$.
- *Recent Sessions.* Likewise, if mentor m is very frequently accepted by mentees in recent days before T_q, he/she can be very popular recently and can be accepted by mentees to deal with new request q. The feature is defined as: $SR(m,q) = \frac{N'_{rec}(m,q)}{|Q_T|}$.

4 Results of Analysis

To verify the effectiveness of the proposed predictive features in differentiating mentors with and without willingness, we plot the cumulative distribution functions (CDF) of various features, as shown in Figure 1. The plot reflects

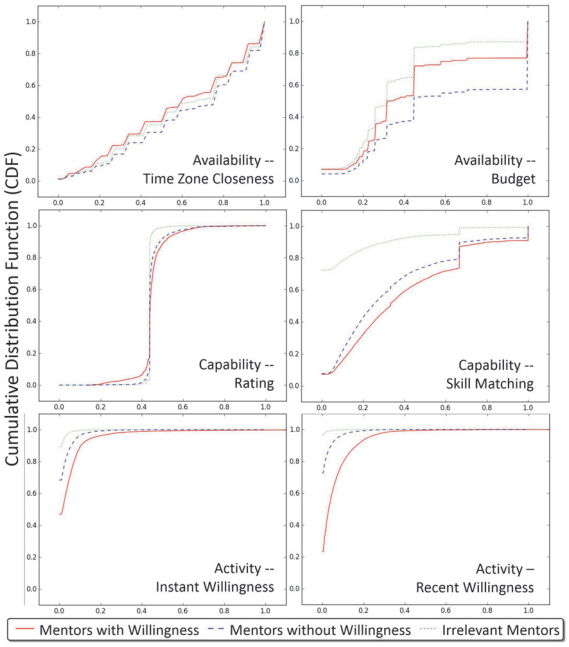

Figure 1: Comparison of the cumulative distribution functions of various features (values) for mentors with willingness, mentors without willingness, and irrelevant mentors.

the effectiveness of a feature, and consists of three curves corresponding to mentors with and without willingness, and irrelevant mentors who are randomly selected from the collection of all mentors. The bigger the area between curves of mentors with and without willingness, the more effective the feature is in differentiating the willingness or not. We can see that the proposed features are generally able to distinguish mentors with willingness from those without willingness. It is especially effective for availability and proximity features. Note that due to page limit, we do not present the results for all of the features, but they exhibit similar trends.

We also verify whether the proposed features are effective in ranking the final accepted mentors (by mentees) at high positions of the recommended list of mentors. We generate the box plots of acceptance rank of the ground-truth accepted mentor using each feature, as shown in Figure 2. The higher the rank (the lower rank value) is, the more effective the feature is in finding the mentors really accepted by mentees. We can find activity features can produce better rankings, and capability features work well, too. These observations from feature analysis encourages us to combine these features using supervised learning in the following.

5 Conclusion

This paper proposes to analyze which mentors would like to respond the request input by a mentee in an instant mentoring service, Codementor. We formulate the Mentor Willingness Ranking problem. Based on the given requests, we devise three diverse sets of features to model the mentors' willingness, including Availability, Capability, and Activity. These feature sets are validated to be effective for both tasks.

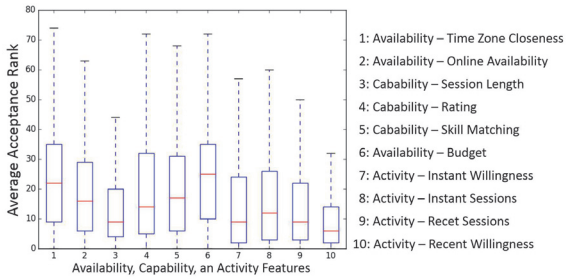

Figure 2: Comparison of the average acceptance ranks (using individual feature values to generate a ranked list of mentors) for availability, capability, and activity features.

This study is first attempt to explore mentors' response willingness in instant mentoring services, and the results encourage advanced analysis and developing complicated models to generate more satisfying mentors to mentees.

6 Acknowledgements

This work was sponsored by Ministry of Science and Technology (MOST) of Taiwan under grant 104-2221-E-001-027-MY2. This work is also supported by Multidisciplinary Health Cloud Research Program: Technology Development and Application of Big Health Data, Academia Sinica, Taipei, Taiwan under grant MP10212-0318.

7 References

Bouguessa, M.; Dumoulin, B.; and Wang, S. 2008. Identifying authoritative actors in question-answering forums: the case of yahoo! answers. In *Proc. of KDD*, 866-874.

Gkotsis, G.; Stepanyan, K.; Pedrinaci, C.; Domingue, J.; and Liakata, M. 2014. It's all in the content: state of the art best answer prediction based on discretisation of shallow linguistic features. In *Proc. of ACM WebSci*, 202-210.

Kutty, S.; Nayak, R.; and Chen, L. 2014. A People-to-People Matching System Using Graph Mining Techniques. *World Wide Web Journal*, 17:311-349.

Li, L; and Li, T. 2012. MEET: A Generalized Framework for Reciprocal Recommendation Systems. In *Proc. of ACM CIKM*, 35-44.

Pizzato, L.; Rej, T.; Chung, T.; Koprinska I.; and Kay, J. 2010. RECON: A Reciprocal Recommender for Online Dating. In *Proc. of ACM Recommender Systems*, 207-214.

Riahi, F.; Zolaktaf, Z.; Shafiei, S.; and Milios, E. 2012. Finding Expert Users in Community Question Answering. In *WWW 2012 Companion* (CQA workshop), 791-798.

Tian, Q.; Zhang, P.; and Li, B. 2013. Towards Predicting the Best Answers in Community-Based Question-Answering Services. In *Proc. of ICWSM*, 725-728.

Yang, B.; and Manandhar. S. 2014. Tag-based expert recommendation in community question answering. In *Proc. of IEEE/ACM ASONAM*, 960-963.