

Discount Expertise Metrics for Augmenting Community Interaction

Pei-Yao Hung¹, Mark S. Ackerman²

¹School of Information, University of Michigan, USA, ²Dept. of EECS and School of Information, University of Michigan, USA

{¹peiyao, ²ackerm}@umich.edu

Abstract. Community members would benefit by finding others who can answer questions or provide assistance. We want to find new ways to help find people who have expertise by creating metrics that can easily estimate expertise. In this paper, we explore discount expertise metrics that are easy to obtain and use. We present one such discount expertise metric that uses people's browsing history to estimate technical expertise. In a study of 26 users, we show that this metric can distinguish experts and novices from others. We also validate this measure by comparing it to the ground truth about our users. We also discuss the possibilities in using measures like this, based on digital traces, to augment interaction within online communities, for example, gauging newcomers in a community of practice or evaluating people in a learning community.

1 Introduction

Expertise finding is important for communities of practice and communities of interest. These online communities allow people around the world to connect and work for common goals. In both, members often need to identify experienced colleagues or individuals with the right expertise to answer questions or provide assistance. For example, people can find answers to technical questions (Mamykina, Manoim, Mittal, Hripcsak, & Hartmann, 2011) or connect to other patients with similar problems (Civan-Hartzler et al., 2010; Huh & Ackerman,

2012). However, finding people with the right level of expertise can be challenging in these environments.

Existing approaches to evaluate expertise often lack the flexibility to adapt across time (e.g., by using a user's static profile) or are labor intensive (e.g., having profiles constructed from user content), requiring a considerable amount of content to be contributed by users. Many existing approaches have the issue of maintenance (Ehrlich, Lin, & Griffiths-Fisher, 2007).

We would like to develop a simpler way to measure expertise. We call these measures *discount expertise metrics*. In this paper, we examine one such relatively straightforward way to infer users' expertise that is based on online browsing. We targeted technical expertise, because that is a simpler case to study. In our approach, we identified several features from browsing history data that provided a good correspondence between an individual's expertise and the content she consumes, specifically pages she visits online for information.

We collected browsing history data from 26 participants, from beginners and intermediates to experts in natural settings (e.g., working on their own technical projects). In addition, we interviewed these participants to understand their programming experience. With the understanding of participants' experience and the data about programming related sites they visited online, we demonstrated that it is possible to automatically infer people's levels of technical expertise based on their browsing history.

In the rest of the paper, we first review the related literature and discuss why a new approach is needed to estimate people's expertise. Next, we present the study's research question and methods. Finally, we discuss the results and future work.

2 Related Work

Expertise finding has been identified in the literature as an important tool for facilitating interaction within organizations or online communities. A member in a community usually specializes in certain domains and thus acquires specialized knowledge that is hard to obtain otherwise. People looking for help are likely to find it beneficial to seek help from a person with the required expertise.

McDonald and Ackerman (1998) identified three types of processes during expertise finding: expertise identification, expertise selection, and escalation. Estimating people's expertise is a major component of expert identification. To help find a suitable person with the correct level of expertise, we need to estimate expertise of possible helpers.

To estimate expertise, three types of data and corresponding mechanisms have been proposed for expertise estimation. The first type of data contains descriptions about a person, i.e., the "profile" approach mentioned above, where the profile is manually produced by the person or others. This type of data usually

includes information about the person's expertise (e.g., topics that this person specializes in), generated by the person, friends, or colleagues (Farrell, Lau, Nusser, Wilcox, & Muller, 2007).

The second type of data used to estimate expertise includes artifacts produced by a person (e.g., documentation), the "artifact" approach. This type of data such as posts on an online discussion forum (Nam & Ackerman, 2007) might not directly specify what the person is good at, but by reviewing the data, people are likely to understand what kinds of topics or areas on which this person focused.

The third type of data for expertise estimation is generated from the interaction with others in the community or outside the community (e.g., email conversation or forum discussion), the "interaction" approach. This type of data emphasizes interactions concerning a specific topic or problem, and thus will often highlight a very specific area in which the people involved in the conversation specialized. The data enables the possibility of differentiating questions askers (less expert) from answerers (more expert). Various studies (Hanrahan, Convertino, & Nelson, 2012; Zhang & Ackerman, 2005; Zhang, Ackerman, Adamic, & Nam, 2007) have shown that discussions through email, online forum or online Q&A platforms can be used to analyze the expertise of people involved. These methods have the potential to match questions and people who can answer them.

While the above studies demonstrate how different kinds of data can be used to infer people's expertise, these approaches have several limitations. The "profile" approach requires people to contribute to their own profiles or others', thus requiring continuous effort to keep the data updated. The "artifact" approach requires people to spend a considerable amount of time documenting their work, which might not be always possible. The "interaction" approach has the drawback that estimates are not available for those who have not participated in the interactions. In addition, the use of interaction data can also raise concerns about privacy. These disadvantages might not disappear even for a hybrid approach (McDonald & Ackerman, 2000; Reichling & Wulf, 2009).

To address these limitations, our approach provides a universal way to estimate levels of expertise on different subjects using browsing history with the following advantages. First, many users of the Internet consume content but do not contribute. This type of users can be accommodated and an estimation of their expertise can be provided. Second, our approach provides continuous and updated estimation of people's expertise on different topics as long as they visit relevant web pages on the Internet. Third, applications using our approach for searching expertise do not require people to interact with one another. Using technical expertise as our target population, we seek to understand the pros and cons of such approach in this paper.

White et al. (2009) had a similar approach. They investigated how domain expertise, including that of computer scientists, might affect people's search behavior. Their definition of experts in the study was defined, however, as people

With the tool, we collected between 2 and 4 weeks browsing history from each of our participants. In total, we obtained 201,700 records. The records consisted of 1) URL, 2) title, and 3) timestamp of when each page was visited. The length of the period was selected under the assumption that programmers' levels of expertise would not change much during the period, and yet we wanted to obtain enough data to characterize the participants since some of them, especially the novices, might not perform technical work every day during the week. In addition, we had a 30-minute semi-structured interview with each participant to collect data about their programming and technical experience.

We designed the rating scheme for expertise as shown in Table I, based on earlier work (Zhang et al., 2007). This rating scheme considered technical education, internships, and professional experience. The ratings are cumulative, meaning that people who satisfied multiple criteria will increase their scores. The data were used by the research team to generate the ground truth of the participants' levels of expertise.

| Level | Experience |
|--------------|---|
| +5 | 6+ year professional programming experience |
| +4 | 4+ year professional programming experience |
| +3 | Computer Science (CS) training, or 2 - 3 years professional programming experience |
| +2 | Electrical Engineering (EE) training, 1 year professional programming experience, or 3 - 4 years assistant/part-time programming experience |
| +1 | Learning programming for the first time/year |

Table I. Rating scheme for programming expertise.

As suggested by previous research on programmers' information seeking behavior online (Brandt, Dontcheva, Weskamp, & Klemmer, 2010; Hanrahan et al., 2012), people regularly visit the following types of pages during the process of programming (or debugging): Document, Tutorial, Blog, Library/Repository, Q&A, Forum, and Search. We reviewed 1000 randomly sampled records from participants' data and confirmed that these were the types of pages that our study participants visited.

The analysis consisted of the following steps: 1) labeling a subset of data, 2) training binary classifiers for each type of web page 3) labeling all the data using the trained classifiers, 4) extracting features (i.e., page types) from labeled data, and 5) creating classifiers for the users' level of expertise.

To acquire the training data, we iteratively obtained a random sample from our data and manually labeled each page until we acquired enough training data

(about 250 records for a category). After excluding non-English pages, we calculated the tf-idf vector for each webpage using uni-grams and bi-grams. The vectors were then used as input data for Linear Support Vector Classification to generate a binary classifier. We followed this approach to train the binary classifiers for all the page types (e.g., Document).

With each record being labeled with their page types (e.g., Q&A and tutorial), we computed the page counts for each type for each programmer. We then obtained a normalized vector representation from the page counts for each programming page types for each programmer and compared it to the ground truth as rated by the research team.

4 Results

At first, we attempted to create a classifier that used logit regression (similar to simple Bayesian classifiers) to determine the participant’s level of expertise. However, the relationship between the level of expertise of a participant and the types of pages he visits is more complex than we originally thought.

Instead, we found that we could obtain a useful indicator by simplifying the problem. We defined a simple ‘novice’ classifier based on whether over 80% of their page visits were in the tutorial category. All participants classified as novices by this classifier, were rated by the research team as novices (level 1 or 2), as shown in row 1 in Table II. This gave us a conservative classifier, in that there were no false positives, although at the cost of having many false negatives (i.e., it missed people who should have been classified as novices). The results are consistent with our intuition that, for beginners, step-by-step tutorials would likely be useful for them as they are trying to figure out not only the practices but also the general knowledge of programming.

| Novice Classifier | Ground Truth | Probability |
|--------------------------|---------------------|--------------------|
| Yes | Novice | 4 of 4 |
| No | Novice | 4 of 13 |

Table II. The classification result by the ‘novice’ classifier. All the programmers classified as novices were also rated as novices by the research team.

Similarly, we were able obtain a classifier to conservatively identify people with high expertise. All the participants classified as experts by this classifier were also rated by the research team as experts (level 4 or 5), as shown in row 1 in Table III. This classifier primarily looked at code libraries and Q&A websites. One possible explanation is that experienced programmers would want to seek

libraries and packages written by others to avoid reinvention, and they are likely to consult Q&A websites for advanced and specific questions that are not addressed in tutorial pages.

| Expert Classifier | Ground Truth | Probability |
|--------------------------|---------------------|--------------------|
| Yes | Expert | 4 of 4 |
| No | Expert | 5 of 13 |

Table III. The classification result by the ‘expert’ classifier. All the programmers classified as experts were also rated as experts by the research team.

While we could infer whether a programmer was a novice or an expert, it was difficult to infer whether a programmer was what we called ‘intermediate’. For programmers who were rated as level 3, their visit patterns were mixed. For example, out of 9 participants who were rated as 3, three of them spent over 60% of their visits on tutorial pages (2 of them over 80%), while 4 of them spent over 60% of their visits on library, repository, and Q&A pages. This is not surprising since intermediates consist of people who have left their status as beginners and are still in the process of being proficient in programming. As a result, their visit patterns would involve traits of beginners, but also display some traits of experts gradually as their expertise increases.

5 Augmenting Community Interaction

With these initial results, we would like to discuss a few opportunities that the similar expertise metrics might present for augmenting communities as we mentioned in the introduction. First, the metrics could provide an initial estimation before a new member initiates any interactions within a community of practice or a community of interest. When someone wants to join a community of practice, for example, existing community members could use the metrics to more easily gauge the expertise of a new member. This could smooth the process of entering a Q&A community of practice since the metrics could quickly create a representation of expertise using only the user’s browsing history.

Second, similar metrics could track the progress of expertise development after entering into Q&A or learning (e.g., MOOC) communities to provide an up-to-date estimation. For instance, the metrics could help to track learning progress or expertise development as students progress throughout courses. As well, these metrics could be used to pair students for discussion or group projects based on their levels of expertise.

The above uses of expertise are available with any measure of expertise, but *discount* metrics of expertise allow many more people to be ranked for a community of practice or a community of interest. Since many users of the Internet consume content but do not contribute, discount metrics would lead these communities to be more inclusive.

Finally, it should be noted that similar metrics could be used across communities. The metrics could also help create ad-hoc networks of people. For instance, a platform might use discount expertise metrics to create teams of suitable programmers to solve technical problems or to facilitate collaborative problem solving.

6 Future Work

The initial analysis that applied logit regression, a form of linear regression, to the data suggested that the relationship between programmers' levels of expertise and their page visits might be more complex than a linear relationship. For future work, we want to consider ensemble methods such as boosting so as to develop better classifiers for level of expertise and also for programming-relevant web pages, focusing on hard-to-classify examples. It may also be possible to use the visible content (e.g., words) and the page structure (e.g., HTML DOM tree) as training data to improve our classifiers.

In this study, we could not use the search category because search engine optimization prevented us from obtaining the same content as programmers and the proportion of search queries did not correlate well with the levels of expertise. Future work could consider including factors such as search result snippets, search sessions, and longer-term search patterns or trajectories into new metrics.

While the results generally aligned with our intuition, there were some interesting nuances about technical expertise that we noticed that we would like to follow up. For example, in our study, there were three programmers, identified as intermediates or experts, who exhibited behaviors belonging to what we expected from novices (e.g., visiting tutorial pages more) as they were learning a new programming language. To incorporate this variation, we have considered two additional studies: (1) a longitudinal study (e.g., 6-12 months) that could monitor the possible changes of expertise and (2) a study that separated estimating expertise for different programming languages. Combining these two approaches might allow us to better examine the nuance of how programmers' expertise might change as reflected in their browsing traces.

7 Conclusion

In this study, we have demonstrated the potential of using people’s browsing history to provide a *discount expertise metric* estimating their levels of expertise. By collecting and analyzing the browsing history of 26 programmers, we showed that people’s visits to topic-relevant web pages could be one useful indicator of their expertise. We discussed several opportunities of using these metrics to augment community interaction at different stages and for different purposes, for example, gauging newcomers in a community of practice or evaluating people in a learning community. Future work includes improving classification performance, performing a longitudinal study to monitor expertise changes, using programming language detection for estimating technical expertise about different programming languages, and analyzing query logs.

8 References

- Brandt, J., Dontcheva, M., Weskamp, M., & Klemmer, S. R. (2010): ‘Example-centric programming: integrating web search into the development environment’, *ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 513–522.
- Civan-Hartzler, A., McDonald, D. W., Powell, C., Skeels, M. M., Mukai, M., & Pratt, W. (2010): ‘Bringing the field into focus: user-centered design of a patient expertise locator’, *ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 1675–1684.
- Ehrlich, K., Lin, C.-Y., & Griffiths-Fisher, V. (2007): ‘Searching for Experts in the Enterprise: Combining Text and Social Network Analysis’, *ACM SIGGROUP Conference on Supporting Group Work*, pp. 117–126.
- Farrell, S., Lau, T., Nusser, S., Wilcox, E., & Muller, M. (2007): ‘Socially augmenting employee profiles with people-tagging’, *ACM symposium on User interface software and technology*, pp. 91–100.
- Hanrahan, B. V., Convertino, G., & Nelson, L. (2012): ‘Modeling Problem Difficulty and Expertise in Stackoverflow’, *ACM Conference on Computer Supported Cooperative Work Companion*, pp. 91–94.
- Huh, J., & Ackerman, M. S. (2012): ‘Collaborative help in chronic disease management: supporting individualized problems’, *ACM conference on Computer Supported Cooperative Work*, pp. 853–862.
- Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., & Hartmann, B. (2011): ‘Design lessons from the fastest Q&A site in the west’, *ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 2857–2866.
- McDonald, D. W., & Ackerman, M. S. (1998): ‘Just Talk to Me: A Field Study of Expertise Location’, *ACM Conference on Computer Supported Cooperative Work*, pp. 315–324.
- McDonald, D. W., & Ackerman, M. S. (2000), ‘Expertise recommender: a flexible recommendation system and architecture’, *ACM conference on Computer supported cooperative work*, pp. 231–240.
- Nam, K. K., & Ackerman, M. S. (2007): ‘Arkose: Reusing Informal Information from Online Discussions’, *ACM SIGGROUP Conference on Supporting Group Work*, pp. 137–146.

- Reichling, T., & Wulf, V. (2009): 'Expert recommender systems in practice: evaluating semi-automatic profile generation', *ACM SIGCHI Conference on Human Factors in Computing Systems*, pp. 59–68.
- White, R. W., Dumais, S. T., & Teevan, J. (2009): 'Characterizing the Influence of Domain Expertise on Web Search Behavior', *ACM International Conference on Web Search and Data Mining*, pp. 132–141.
- Zhang, J., & Ackerman, M. S. (2005): 'Searching for expertise in social networks: a simulation of potential strategies', *ACM SIGGROUP conference on Supporting group work*, pp. 71–80.
- Zhang, J., Ackerman, M. S., Adamic, L., & Nam, K. K. (2007): 'QuME: a mechanism to support expertise finding in online help-seeking communities', *ACM symposium on User interface software and technology*, pp. 111–114.

9 Acknowledgements

This work was funded, in part, by the National Science Foundation (0905460) and the University of Michigan School of Information. We also thank Qiaozhu Mei, Mark Newman, and our participants for their help.