Early Detection of Topical Expertise in Community Question Answering

David van Dijk[†‡]
d.v.van.dijk@hva.nl

Manos Tsagkias[§]
manos@904labs.com

Maarten de Rijke[‡]
derijke@uva.nl

[†]Create-IT, Amsterdam University of Applied Sciences, Amsterdam, The Netherlands
[‡]University of Amsterdam, Amsterdam, The Netherlands
[§]904Labs, Amsterdam, The Netherlands

ABSTRACT

We focus on detecting potential topical experts in community question answering platforms early on in their lifecycle. We use a semi-supervised machine learning approach. We extract three types of feature: (i) textual, (ii) behavioral, and (iii) time-aware, which we use to predict whether a user will become an expert in the longterm. We compare our method to a machine learning state-of-the-art method in expertise retrieval. Results on data from Stack Overflow demonstrate the utility of adding behavioral and time-aware features to state-of-the-art method with a net improvement in accuracy of 26% for very early detection of expertise.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.3 Information Search and Retrieval

Keywords

Community question answering; User profiling; Expertise finding

1. INTRODUCTION

Community Question Answering (CQA) sites such as Stack Overflow[1] provide a growing resource of information. Users contribute and interact by posting questions, answers and comments, and provide feedback by voting on questions and answers and by selecting the best answer to their question. Key to the success of CQA platforms are the users that can provide high quality answers to the more difficult questions posted, however, this type of user is scarce [10, 11]. In this setting, it becomes important to stimulate the growth of the group of users who provide the most useful answers. There are several methods for doing so; applying gamification methods on the website for incentivizing users to contribute their expertise is one [2]. Another angle to this challenge is to detect and nurture users with topical expertise early enough so we can recommend questions relevant to their expertise [7]. Central here, and our aim, is to identify users with a strong potential to become prolific users on a subject, i.e., potential topical experts, from their first interactions with the platform. The main challenge here lies in inherent data sparsity issues: how to profile an expert given only a handful of data points, i.e., questions, answers and comments.

Detecting topical expertise is a well-studied problem, which relates to expertise finding and retrieval [3]. Common to all methods is the profiling of a user and a topic for generating candidate matches. In our scenario, a user's expertise manifests itself via multiple channels, e.g., comments, questions, answers, accepted answers. Our hypothesis is that when we combine information from these channels, we can accurately detect early expertise even in scenarios where data is sparse. Focusing on early expert detection in CQA, Pal et al. [5] apply a machine learning approach to identify general experts during the first few weeks after their first answer. Bouguessa and Romdhane [4] propose a parameter-free mixture model-based approach for identifying of authorities in online communities. Pal et al. [6] observe how behavioral signals evolve over time for grouping experts. Our work differs from previous work on early expertise discovery in two ways: (i) in how we define early expertise, and (ii) we study and report on the importance of combining a large number of textual, behavioral and time-aware signals for detecting early expertise.

We cast the task of early detection of topical expertise as a classification problem: to decide whether a user will be an expert in the long-term by using evidence from increasingly long timespans of a user's early behavior. We define early expertise based on the number of best answers given by a user. A best answer is the one answer that gets accepted by a question poster as a the most useful. Users with ten or more best answers on a topic are considered experts on the topic. We engineer three feature sets to capture early expertise: (i) textual, (ii) behavioral, and (iii) time-aware. We seek to answer the following research questions: (RQ1) What is the impact on classification effectiveness when we use each feature set individually and in combination over state-of-the-art methods in expertise finding? Does performance remain stable over time? (RQ2) What is the most important feature set for early detection of topical expertise among: textual, behavioral, and temporal feature sets? (RQ3) What is the most important individual feature within and across feature sets? To answer these questions we use data from Stack Overflow, a CQA platform for programming-related topics.[2] Our experimental results show significant improvements over the state-of-the-art methods in expertise finding and validate the utility

[1] http://stackoverflow

[2] Our dataset is publicly available from http://ilps.science.uva.nl/resources

SIGIR '15, August 9–13, 2015, Chile, Santiago.
Copyright 2015 ACM 978-1-4503-3621-5/15/08
http://dx.doi.org/10.1145/2766462.2767840 ...$15.00.

of using behavioral and time-aware features from multiple behavioral channels. Results also show we can achieve high accuracy in early detection of topical expertise at relatively early stages of a user's lifespan, i.e., F_1 score 0.75 at user's first best answer.

2. APPROACH

Our approach for early discovery of potential experts is based on a semi-supervised machine learning method. We extract a set of features indicative of a user's expertise on a topic, which we use to train a classifier that learns whether a user shows signals of early expertise given a topic. We cater for early expertise by carefully crafting the training data used to train the classifier. Our method is semi-supervised because we automatically generate training data, by labeling experts in a data-driven manner; see Section 3.

We first need to define early expertise. Although time is a natural way for separating early from seasoned experts, the diverse behavioral patterns among experts make it hard to define early expertise using time in an experimental setting [6, 8, 9]. One future expert might submit ten best answers within two days after joining while another may post one comment during their entire first week. We define expertise based on best answers. Here, a *best answer* is one that gets accepted by the question poster. The more best answers a user gives, the more expert they are. We took as experts those users with one standard deviation number of best answers larger than the average user. On our dataset (see below) this translates into people with more than nine best answers on a topic. *Early expertise* is defined as the expertise shown by a user between the moment of joining and becoming an expert, based on the best answers provided. We interpret the values of the selected features between the moment of joining and becoming an expert as the strength of a user's early expertise, and predict future expertise based on it.

Table 1 provides a summary of the features we use.

Textual features. We build on prior work on expertise retrieval by [3]. It aggregates a user's textual relevance scores of answers as an indication of expertise. We start with generating a profile per topic—here, a topic is a tag associated with a question on Stack Overflow—by retrieving all questions that are associated with the topic along with all comments, answers, and comments to answers associated with the question. We profile terms by ranking them using tf.idf scoring and select the top-100 terms for a topic's profile. For profiling users we retrieve all answers that are posted by a certain user that are associated with the topic. Once we have topic and user profiles, we apply Model 2 [3] to determine the user's textual relevance to the topic. In particular,

$$p(q|ca) = \sum_{d \in D_{ca}} f(d, ca) \cdot p(d|ca), \qquad (1)$$

where q is a topic, ca is a candidate expert, d is a document (i.e., question, answer, comment), $f(d, ca)$ is a function denoting the textual similarity between the textual profiles of a document and a candidate user, and D_{ca} stands for the documents of the candidate, in our case the answer history of the user. We consider three readily used textual similarity functions as individual features: (i) language modeling (LM), (ii) BM25 (BM25), and (iii) tf.idf (TFIDF).

Behavioral features. On top of our textual features based on expertise retrieval, we mine a user's posting behavior to extract nine features that are indicative of their expertise. We extract these features per topic. Below, we describe them shortly.

Number of questions, answers and comments are used based on intuitions, such as that an expert is likely to ask fewer questions on his field of expertise and could be selective in what questions to answer or comment on. Z-Score, a measure to quantify expertise, is defined as $z = \frac{a-q}{\sqrt{a+q}}$ [11] and combines the number of answers and the number of questions into one score. Similarly to z-score, we engineer features that combine different behavioral signals as ratios between the number of different types of post: Nr. of questions divided by nr. of answers., Nr. of answers divided by nr. of comments and Nr. of comments divided by nr. of questions. First and timely answers a have a higher chance of becoming accepted by a questioner. Users that show timely answering behaviour are more likely to get their answers accepted by users.

Table 1: Summary of the three types of feature we consider: (i) textual, (ii) behavioral, and (iii) time-aware, 25 in total.

ID	Feature	Gloss
Textual features		
1	LM	Model 2 using language modeling scoring
2	BM25	Model 2 using BM25 scoring
3	TFIDF	Model 2 using tf.idf scoring
Behavioral features		
4	Question	Number of questions by a user
5	Answer	Number of answers by a user
6	Comment	Number of comments by a user
7	Z-Score	Question-answering ratio
8	Q.-A.	Nr. of questions divided by nr. of answers
9	A.-C.	Nr. of answers divided by nr. of comments
10	C.-Q.	Nr. of comments div. by nr. of questions
11	First Answer	Number of first answers a user has posted
12	Timely Answer	Nr. of answers posted within 4h by a user
Time-aware features		
13	Time Interval	Days between joining and N-th best answer
14	LM/T	LM / Time interval
15	BM25/T	BM25 / Time interval
16	TFIDF/T	TFIDF/ Time interval
17	Question/T	Question / Time interval
18	Answer/T	Answer / Time interval
19	Comment/T	Comment / Time interval
20	Z-Score/T	Z-Score / Time interval
21	Q.-A./T	Q.-A. / Time interval
22	A.-C./T	A.-C. / Time interval
23	C.-Q./T	C.-Q. / Time interval
24	First Answer/T	First Answer / Time interval
25	Timely Answer/T	Timely Answer / Time interval

Time-aware features. We also include features with a focus on expert's activity patterns over time. We consider the time interval between a user's best answers, and we measure it as the number of days between the moment a user joined the forum and when the posted his N-th best answer ($1 \leq N \leq 9$). Our hypothesis here is that an expert is likely to take less time between posting best answers than a non-expert user. We create a time-aware version for each of the textual and behavioral features we discussed above, by dividing the respective feature value by the time interval. This provides us, e.g., with the number of answers per day. As the time interval can substantially vary between users, we expect time-aware features to be more indicative than their non-time-aware variants.

3. EXPERIMENTAL SETUP

In addressing the early detection of topical expertise problem, we concentrate on developing features and combinations of features that can be used for early detection of expertise. In this respect, our goals are comparable to those of [1, 5]. In particular, we want to know the effectiveness of our complete set of features, and of individual feature sets, for classifying users as experts and non-experts; see Table 2 for a summary of systems we consider.

Table 2: Summary of the systems we consider, and the individual features they consist of.

ID	Type	Feature	ID	Feature
A	Textual	1–3	E	C + D
B	Behavioral	4–12	F	A + B
C	Time-aware 1	13–25	G	A + B + C + D
D	Time-aware 2	1–25 per bin		

Table 3: Dataset statistics over 100 topics and 90,486 experts. A user can be expert in more than one topic, contributing more than one expert.

X per topic	Mean	Std.Dev	Min	Max
Users	15,374	13,884	2,271	73,009
Experts	905	1,383	68	6,622
Questions	7,075	15,598	10	90,998
Answers	87,273	156,226	2,150	816,662
Comments	86,959	164,600	1,553	783,430

Our dataset comes from Stack Overflow,[3] covers the period August 2008–mid-September 2014, and consists of 6,044,028 questions, 10,794,654 answers and 24,708,671 comments. We select the 100 most active topic tags in terms of number of questions and answers to maximize the number of experts we can use for training and testing. Highly semantically related topic tags are grouped together. We mark users as experts on a topic when they have ten or more of their answers marked as best by the question poster, which is one standard deviation larger than the average number of answers over all users and topics. Table 3 lists statistics for our dataset.

Machine learning. Our semi-supervised machine learning method starts out with unlabeled data and adopt a data-driven approach to labels users who provide above average best answers on a topic as topical experts. Training data for users is generated on the period between joining and becoming an expert. To prevent classification bias in the training set, we balance the number of experts and non-experts per topic by down-sampling non-experts uniformly over the number of best answers. We divide this dataset into two: we hold out 10% for feature engineering and development, and 90% for testing. We choose to evaluate the effectiveness of three classifiers: Gaussian Naive Bayes, Linear Support Vector Classification and Random Forest (RF); no parameter optimization is performed. In preliminary experiments, RF, as implemented in Scikit,[4] outperformed the other two classifiers, hence we use it for our main experiments. We use Apache Lucene[5] for extracting textual features.

We report on F1 scores over each best answer of a user starting from their first best answer and going up to their ninth answer, i.e., one best answer before they are deemed experts. At each step, we perform 10-fold cross validation on our test set. We use a two-tailed paired T-test to determine statistical significance and report on significant differences for $\alpha = .05$ and $\alpha = 0.01$.

4. RESULTS AND ANALYSIS

Our first experiment aims at answering RQ1: What is the improvement in classification effectiveness when we use each feature set individually and in combination over state-of-the-art methods in expertise finding? Does performance remain stable over time? Among individual feature sets, the textual feature set (system A) outperforms the behavioral one (system B) up to best answer three, and after that we see the reverse, i.e., the behavioral feature set outperforming the textual one, possibly due to aggregation of more behavioral data. The combination of all features sets (system G) shows the best performance among all systems peaking F1 at 0.7472, and outperforms the baseline (system A) in statistically significant way. The performance of time-aware feature sets (systems C and D) hovers around that of system G because they include all features, just like G, but normalized by time. In general, we find that incremental combinations of feature sets improve performance over the baseline in a statistically significant way (i.e., $G \geq E > F$). Fig. 1 illustrates the performance of all systems over time. Common to all systems is a big leap in performance between best answer one and two, which is likely caused by data sparsity at the first best answer.

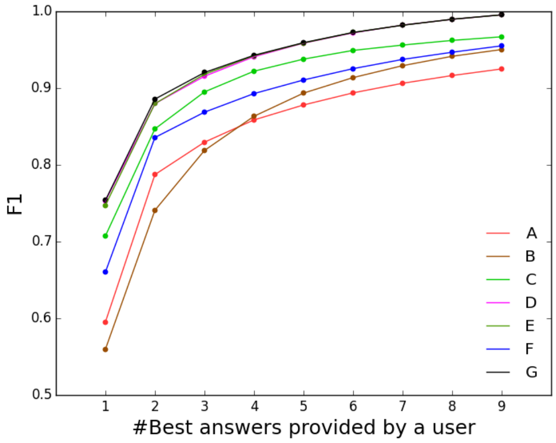

Figure 1: System performance in F1 using individual feature sets and their combination.

Turning to our next two research questions, RQ2 and RQ3: What is the most important feature set among textual, behavioral, and temporal feature sets, and the most important features within and across feature sets? We perform an ablation study, where we remove one feature set at a time from our best performing system (system G); see Fig. 2. The removal of the two time-aware feature sets leads to the largest drop in effectiveness. This shows that normalizing features by time helps detecting early expertise in a statistically significant way. Next, we aim to find out what individual features in our feature sets have the largest impact in performance. By incrementally removing individual features, we found a small set of features that when combined, they perform similarly to our best system (G): TFIDF (feature 3 in Table 1), Answer (feature 5), BM25/T (feature 15), TFIDF/T (feature 16), Answer/T (feature 18), Time Interval (Time-Aware 2, feature 13). This optimal feature set is a combination of textual features, time-aware features, and behavioral features; the kappa-statistic over all best answers is 0.8705, which supports the strong predictive power of this optimal set of features. An interesting finding is that the z-score, a widely-used measure for quantifying expertise, although it has performed well in similar tasks [4–6], in our setting was not selected as an important individual feature.

To better understand our results, we perform an analysis of the performance of our best system (G). First, we look at the time difference between a user's first best answer and their tenth best answer. Fig. 3 (left) shows that a large number of users become experts within the same month they post their best answer but there is also a long tail of users who become experts many months later. Second, we plot system performance ,on the first best answer, on the selected feature set, over the time difference between a user's

[3] https://archive.org/details/stackexchange
[4] http://scikit-learn.org
[5] http://lucene.apache.org

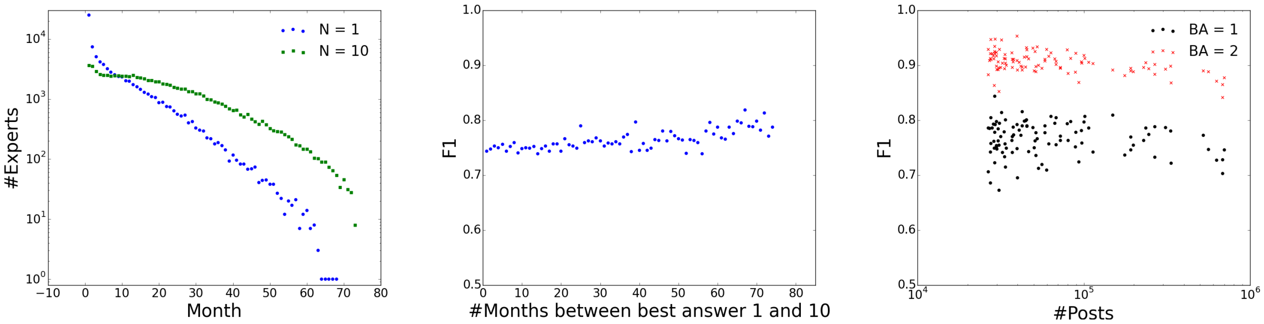

Figure 3: Number of experts that provided their N-th best answer before month X (left). Performance in F1 over experts binned per time between first and tenth best answer (middle). Performance in F1 over size of topics on first and second best answer (right).

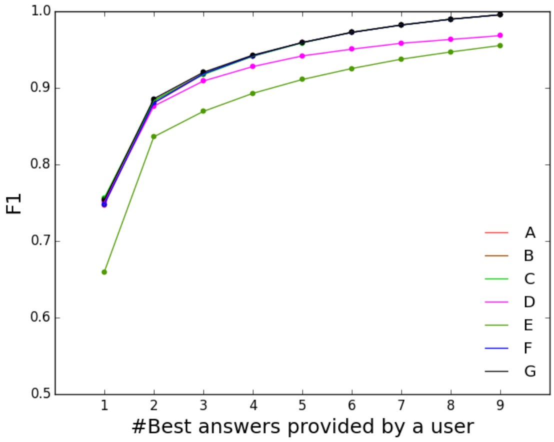

Figure 2: Ablation study on feature sets.

first and tenth best answer in Fig. 3 (middle). System performance is stable over all time bins, suggesting that our classifier is not overfitting for users who become experts in short time, and also performance increases for users who become experts many months after their first best answer. This fact provides evidence that we can accurately detect experts early on (e.g., from 10 to 70 months in advance). Finally, we look at system performance with regards to the size of a topic in terms of number of answers for the first and second best answer. Fig. 3 (right) illustrates that sparser topics have a larger variance in performance, which is likely due to data sparsity itself. However, an interesting finding is that our system performs on average better on sparser topics than on non-sparse ones. This may be attributed to that topics with many answers may contain more noise than sparse ones. In sum, we find that topic size has little influence on system effectiveness, which provides evidence that our system can effectively detect experts for both popular and non-popular topics at an early stage.

5. CONCLUSIONS

We addressed the task of early detection of topical expertise. We proposed a robust way for defining early expertise based on the number of a user's best answers rather than time, catering for different user activity behaviors. We presented a semi-supervised approach using textual, behavioral and temporal feature sets, and demonstrated the effectiveness of our method over a machine learning, state-of-the-art method for expertise finding. We found that behavioral and temporal features when combined with textual features significantly boost effectiveness peaking F1-score at 0.7472; an 26% improvement over the state-of-the-art method. Results demonstrated that our system can accurately predict whether a user will become an expert from a user's first best answer; projected in time, our system makes correct predictions even in 70 months before a user becomes an expert. Although the features to be used may vary, accepted answers are a common phenomenon in QA sites. We therefore expect the method to generalise well within this domain. In future work, we plan to evaluate our method on other corpora as well as extend our features to capture more aspects of early expertise, e.g., answer quality, diversity, novelty, and also track how a user's expertise evolves from one topic to another over time, which can yield a strong predictor of early expertise.

Acknowledgments. This research was supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 312827 (VOX-Pol), the Netherlands Organisation for Scientific Research (NWO) under project nrs 727.011.005, 612.001.116, HOR-11-10, 640.006.013, 612.066.930, CI-14-25, SH-322-15, Amsterdam Data Science, the Dutch national program COMMIT, the ESF Research Network Program ELIAS, the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences (KNAW), the Netherlands eScience Center under project nr 027.012.105, the Yahoo! Faculty Research and Engagement Program, the Microsoft Research PhD program, and the HPC Fund.

REFERENCES

[1] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. Finding high-quality content in social media, with an application to community-based question answering. In *WSDM '08*, pages 183–194. ACM, 2008.

[2] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. Steering user behavior with badges. In *WWW '13*, pages 95–106. ACM, 2013.

[3] K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, and L. Si. Expertise retrieval. *Foundations and Trends in Information Retrieval*, 6(2–3): 127–256, August 2012.

[4] M. Bouguessa and L. B. Romdhane. Identifying authorities in online communities. *TIST*, 6(3), 2015.

[5] A. Pal, R. Farzan, J. Konstan, and R. Kraut. Early detection of potential experts in question answering communities. In *ICUMAP '11*, pages 231–242, 2011.

[6] A. Pal, S. Chang, and J. A. Konstan. Evolution of experts in question answering communities. In *ICWSM '12*. AAAI, 2012.

[7] J. San Pedro and A. Karatzoglou. Question recommendation for collaborative question answering systems with rankslda. In *RecSys '14*, pages 193–200. ACM, 2014.

[8] G. Szabó and B. A. Huberman. Predicting the popularity of online content. *Comm. of the ACM*, 53(8):80–88, 2010.

[9] M. Tsagkias, W. Weerkamp, and M. de Rijke. News comments: Exploring, modeling, and online prediction. In *ECIR '10*, pages 191–203. Springer, 2010.

[10] J. Yang, K. Tao, A. Bozzon, and G.-J. Houben. Sparrows and owls: Characterisation of expert behaviour in stackoverflow. In *UMAP '14*, volume 8538, pages 266–277. Springer, 2014.

[11] J. Zhang, M. S. Ackerman, and L. Adamic. Expertise networks in online communities: Structure and algorithms. In *WWW '07*, pages 221–230. ACM, 2007.