User Name Disambiguation in Community Question Answering

Baoguo Yang
Department of Computer Science
University of York, UK
by550@york.ac.uk

Suresh Manandhar
Department of Computer Science
University of York, UK
suresh@cs.york.ac.uk

Abstract

Community question answering sites provide us convenient and interactive platforms for problem solving and knowledge sharing, which are attracting an increasing number of users. Accordingly, it will be very common that different people have the same user name. When a query question is given, some potential answer providers would be recommended to the asker in the form of user name. However, some user names are ambiguous and not unique in the community. To help question askers match the ambiguous user names with the right people, in this paper, we propose to disambiguate same-name users by ranking their tag-based relevance to a query question. Empirical studies on three community question answering datasets demonstrate that our method is effective for disambiguating user names in community question answering.

1 Introduction

In recent years, community-based question answering (CQA) sites like StackOverflow[1], Quora[2] and Yahoo!Answers[3], have achieved great success and attracted a huge number of users. It is not uncommon that some people in the CQA services share the same user names. Figure 1(a), Figure 1(b) and Figure 1(c) show three lists of user names from three different CQA communities: Travel[4], Webapps (Web Applications)[5], and Cooking[6], where each user name is shared by multiple users. In Figure 1(b), "David" is the most common and ambiguous user name related to 57 users.

In some cases, an off-line person asks people around a difficult question verbally, then he/she may be recommended by word of mouth to visit the CQA homepages of some potential answer providers. However, the links to their homepages are not provided sometimes, then the asker has to search them according to the provided user names. Some user names are unique, and they can easily access the historical QA records of these potential answer providers. However, some are very common and ambiguous, accordingly, many users with the same user name will be displayed.

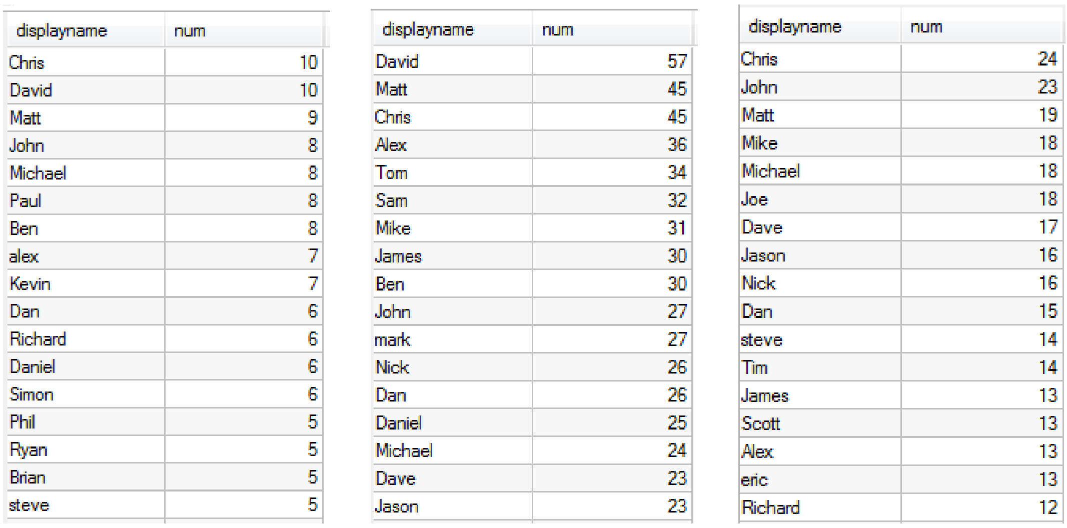
Motivated by the above scenario, it is very necessary to help askers disambiguate these users, which can release them from wondering which user should be the right one. Moreover, if the user name is not clearly given, the askers will waste a lot of valuable time on searching and visiting irrelevant users, which can cause misunderstanding and misleading. Then the asker will get puzzled.

In CQA, given a new question, the related research studies mainly fall into three areas: 1) Answer recommendation (Zhou et al., 2012b; Tian et al., 2013); 2) Similar question retrieval (Cao et al., 2010; Zhang et al., 2014b); 3) Expert user recommendation (Pal and Konstan, 2010; Liu et al., 2011; Zhou et al., 2012a). As for user recommendation, when some user names are ambiguous, the askers will be thrown into another dilemma.

To our knowledge, this is the first work on user name disambiguation in community question answering. Although there have been some studies on user name disambiguation in bibliographic citation records (Han et al., 2005; Treeratpituk and Giles, 2009; Ferreira et al., 2010), the related methods are not directly applicable to our work. In this paper, to disambiguate the same-name users, we present a simple vector-style tag-based method, *relTagVec*, to learn the relevance

[1] http://www.stackoverflow.com/
[2] https://www.quora.com/
[3] http://answers.yahoo.com/
[4] http://travel.stackexchange.com/
[5] http://webapps.stackexchange.com/
[6] http://cooking.stackexchange.com/

Proceedings of Recent Advances in Natural Language Processing, pages 707–713,
Hissar, Bulgaria, Sep 7–9 2015.



displayname	num
Chris	10
David	10
Matt	9
John	8
Michael	8
Paul	8
Ben	8
alex	7
Kevin	7
Dan	6
Richard	6
Daniel	6
Simon	6
Phil	5
Ryan	5
Brian	5
steve	5

(a) Travel community

displayname	num
David	57
Matt	45
Chris	45
Alex	36
Tom	34
Sam	32
Mike	31
James	30
Ben	30
John	27
mark	27
Nick	26
Dan	26
Daniel	25
Michael	24
Dave	23
Jason	23

(b) Webapps community

displayname	num
Chris	24
John	23
Matt	19
Mike	18
Michael	18
Joe	18
Dave	17
Jason	16
Nick	16
Dan	15
steve	14
Tim	14
James	13
Scott	13
Alex	13
eric	13
Richard	12

(c) Cooking community

Figure 1: Example of lists of most ambiguous user names in some CQA communities (all the lists are not shown completely, Figure 1(a) is based on the data between 2011-06-21 and 2013-05-09, Figure 1(b) is based on the data between 2009-07-15 and 2013-03-10, and Figure 1(c) is based on the data before 2013-03-10).

between each user and the question by comparing their tag lists, where each tag is represented by a vector. Then the one who has the highest relevance score will be the right person to recommend. Experimental results on three CQA datasets from StackExchange[7] network demonstrate that our method is very effective, and performs much better than the baseline methods.

The remainder of this paper is organized as follows. Section 2 presents the related work. Then we introduce the framework of our method in Section 3. Section 4 reports the empirical studies on real CQA datasets. Finally, we conclude this paper in Section 5.

2 Related Work

In this section, we briefly review the work that is related to some extent.

User Name Disambiguation. Han et al. (2005) present a K-way spectral clustering approach to disambiguate users in citations. In (Treeratpituk and Giles, 2009), a random forests based machine learning algorithm is introduced for pairwise user name disambiguation. A novel approach, Self-training Associative Name Disambiguator (Ferreira et al., 2010), is proposed for author name disambiguation through two steps. Recently, another method has been presented in (Zhang et al., 2014a) by exploring the link information in collaboration networks for disambiguating user names. Nevertheless, these disambiguation methods cannot be directly used for user name disambiguation in C-QA.

Expert Learning. Zhang et al. (2007) propose to use network-based ranking algorithms to find authoritative users. In (Guo et al., 2008), to recommend answer providers, a two-step method is introduced and the user profiles are also explored. Liu et al. (2011) present a pairwise competition based method for estimating user expertise scores. In (Zhou et al., 2012a), both link analysis and topical similarity are combined in a probabilistic model for experts finding in CQA. In (Yang and Manandhar, 2014), the descriptive ability of users is also studied.

3 Framework of Our Method

In this section, the concrete steps of our *relTagVec* method are presented and explained.

3.1 Computing user relevance to the questions

For each user u, we can get a list of tags, T_u, from the questions to which he/she has recently answered. For each question q, the corresponding tag list can be represented as T_q. We use word2vec (Mikolov et al., 2013) technique to compute the

[7] `http://stackexchange.com/`

vector representation of all the tags. And then the relevance value $relevance(u, q)$ of user u over q can be represented as follows.

$$relevance(u, q) \\ = \frac{1}{|T_q|} \sum\nolimits_{i=1}^{|T_q|} \max_{j=1,2,\ldots,|T_u|} (sim(\mathbf{v}_i^{T_q}, \mathbf{v}_j^{T_u}) \cdot w_j^{T_u}), \quad (1)$$

where $\mathbf{v}_i^{T_q}$ is the vector representation for the i-th tag in the tag list of question q. Accordingly, $\mathbf{v}_j^{T_u}$ is the vector for the j-th tag in the tag list of user u. Here $sim(\mathbf{v}_i^{T_q}, \mathbf{v}_j^{T_u})$ denotes the cosine similarity between $\mathbf{v}_i^{T_q}$ and $\mathbf{v}_j^{T_u}$. In addition, $w_j^{T_u}$ is the weight of j-th tag in the tag list of user u, which can be represented as $w_j^{T_u} = 1/(1+\exp(-N_j^{T_u}))$. Here, $N_j^{T_u}$ is the number of times the j-th tag of user u appearing in the questions to which the user u has answered.

3.2 Selecting the user with highest relevance value

When we get each relevance value $relevance(u, q)$ of candidate users to the query question q, the user with highest relevance value will be considered as the right person to recommend. Here we use $u_{predicted}^{q}(username)$ to denote the predicted user with the name "username" for recommendation over question q.

3.3 Recommending ranked user list

In many cases, a considerable number of users share the same user name, then the prediction to the target person is getting difficult based on insufficient historical data, and the prediction accuracy will be low. It is very necessary to provide a ranking list to the asker.

For a query question q, we rank the candidate users to generate a ranking list based on relevance scores $relevance(u, q)$ in descending order. Then the askers just need to check the top-ranking users, which is time-saving.

4 Experimental Analysis

In this paper, two types of user names are considered.

Type 1: Each provided ambiguous user name is exactly the *DisplayName* of the target user.

Type 2: The recommendation is only given in the form of each target user's first name. For example, a user named "Tom Smith" is mentioned in the name of "Tom" instead. However, there are many members named "Tom" in the community.

4.1 Datasets and Settings

In our experiments, three Data Dumps[8] from Travel[9], Seasoned Advice (Cooking)[10] and MathOverflow communities are used to evaluate our method. Note that all the user names are case insensitive in our experiments.

Travel: We use a Travel Data Dump ranging from June 2011 to September 2014. First, the dataset is divided into two parts, the data before 2013-05-09 is viewed as historical data, while the remainder is used for evaluation.

For Type 1, firstly, from the historical set we select all the user names associated with at least two different users. Then the userIds of all the users who share the same user name will be selected, and then we collect all their previous Q&A records (833 posts associated with 231 different users). Based on the userIds of these historical Q&A records, the questions answered by the corresponding users are selected from the initial evaluation dataset. Then we build the final evaluation data in the form of triples (question, user name, userId). Here the user name is ambiguous, and the user with this userId is a **gold standard** answer provider for this question. The final evaluation dataset contains 298 (question, user name, userId) records. For each ambiguous user name, the associated users with this name form the candidates. Note that each gold standard userId is known in evaluation set without manual annotation.

As for Type 2, we first select all the one-word user names from historical set, then all the user names containing these given names are selected. And then the userIds associated with these given names are collected from historical set, the remainder steps are similar to Type 1.

Cooking: The Seasoned Advice (Cooking) Data Dump is dated from July 2010 to September 2014. For Type 1, we preprocess it in the same way as that for Travel Data Dump. Here the historical set is composed of the data before 2013-03-10, and the rest are used for evaluation. For historical set, we collect 3306 Q&A posts from 982 different users. And we get 284 (question, user name,

[8] https://archive.org/details/stackexchange

[9] http://travel.stackexchange.com/

[10] http://cooking.stackexchange.com/

Methods	User Predicting	User Ranking			
	Accuracy	avgR	MRR	CDR@2	CDR@5
random	0.4536	1.8944	0.6892	0.8284	0.9883
relTitle-Avg	0.6472	1.6296	0.7931	0.8607	0.9894
relTitle-Max	0.6986	1.5790	0.8185	0.8592	0.9894
relTagVec	0.8625	1.2747	0.9148	0.9296	0.9978

(a) MathOverflow

Methods	User Predicting	User Ranking			
	Accuracy	avgR	MRR	CDR@2	CDR@5
random	0.2226	4.7102	0.4179	0.3957	0.6360
relTitle-Avg	0.6360	1.7138	0.7824	0.8304	0.9859
relTitle-Max	0.8551	1.3887	0.9078	0.9152	0.9859
relTagVec	0.9329	1.1166	0.9609	0.9753	0.9965

(b) Cooking

Methods	User Predicting	User Ranking			
	Accuracy	avgR	MRR	CDR@2	CDR@5
random	0.5235	1.5336	0.7535	0.9564	1.0
relTitle-Avg	0.8993	1.1376	0.9435	0.9631	1.0
relTitle-Max	0.9262	1.1107	0.9569	0.9631	1.0
relTagVec	0.9698	1.0335	0.9843	0.9966	1.0

(c) Travel

Table 1: Performance under Type 1.

userId) records for the evaluation set. The preprocessing for Type 2 is similar to that in Travel set.

MathOverflow: The Data Dump for MathOverflow ranging from September 2009 to September 2014 is also publicly available. Here the data before 2011-02-05 is formed as historical data. For Type 1, we finally collect 2770 (question, user name, userId) records for evaluation. All the preprocessing steps for both types are the same as those for Travel Data Dump.

All the experiments are performed on a PC with Pentium Dual-core 2.3 GHz CPU and 4.0 GB RAM. For the tag vector representation, word2vec continuous bag of words (CBOW) model (Mikolov et al., 2013) is used, and the vectors are got based on the question tags from the whole dataset. We set the dimension of each vector as 50, and the training is executed for 10 iterations.

4.2 Experiments on user name disambiguation in CQA

We compare our *relTagVec* method with the following three baseline methods on *Travel*, *MathOverflow* and *Cooking* datasets under Type 1 and Type 2 separately. For each type and each dataset, all the methods are run 10 times, then the averaged results are reported.

Baselines:

- *Random*: A predictor generates random ranking of candidate answer providers for each question.
- *relTitle-Avg*: Given the title $Title_q$ of a query question q, the titles $\{Title_{q_i \in Q_u}\}_{i=1}^{|Q_u|}$ of the previously asked and answered questions Q_u from each candidate user u are collected, then we compute the Jaccard similarity coefficient between $Title_q$ and each $\{Title_{q_i \in Q_u}\}_{i=1}^{|Q_u|}$, and then the averaged similarity value is calculated, which is considered as the relevance score of user u to question q.
- *relTitle-Max*: Different from *relTitle-Avg*, in *relTitle-Max*, the maximum Jaccard similarity value is computed instead of the averaged similarity value.

Metrics: We use accuracy as the metric for the most likely user prediction evaluation. The repre-

Methods	User Predicting	User Ranking			
	Accuracy	avgR	MRR	CDR@2	CDR@5
random	0.1646	9.4405	0.3408	0.3072	0.5505
relTitle-Avg	0.3648	4.8563	0.5509	0.5669	0.8084
relTitle-Max	0.4910	4.4630	0.6359	0.6504	0.8354
relTagVec	0.6947	2.1003	0.7991	0.8250	0.9413

(a) MathOverflow

Methods	User Predicting	User Ranking			
	Accuracy	avgR	MRR	CDR@2	CDR@5
random	0.1731	8.0061	0.3375	0.2933	0.5030
relTitle-Avg	0.4562	3.6558	0.6147	0.6191	0.8228
relTitle-Max	0.6680	3.1181	0.7569	0.7719	0.8391
relTagVec	0.7719	2.2546	0.8459	0.8717	0.9369

(b) Cooking

Methods	User Predicting	User Ranking			
	Accuracy	avgR	MRR	CDR@2	CDR@5
random	0.3199	3.6919	0.5230	0.5446	0.7609
relTitle-Avg	0.6987	1.6355	0.8221	0.8956	0.9646
relTitle-Max	0.8476	1.4200	0.9046	0.9326	0.9697
relTagVec	0.9217	1.1700	0.9535	0.9731	0.9899

(c) Travel

Table 2: Performance under Type 2.

sentation of accuracy is shown as follows.

$$Accuracy = \frac{N_{(u_{predicted}==u_{true})}}{N_{records}},$$

where $N_{records}$ denotes the number of (question, user name, userId) records in the evaluation set, and $N_{(u_{predicted}==u_{true})}$ is the number of records whose answer providers have been correctly matched. Here $u_{predicted}$ denotes the predicted userId, and u_{true} is the ground-truth userId of a user name for a record. The higher accuracy, the better performance is.

Because some user names are shared by many users, we also evaluate the predicted ranking of the ground-truth[11] user by our method and baselines in terms of the following metrics.

- The average rank of ground-truth users (*avgR*): the average rank of ground-truth users among the candidate users for the query questions.
- Mean reciprocal rank (*MRR*): the average of the reciprocal ranks of ground-truth users for the query questions.
- Cumulative distribution of ranks (*CDR*): *CDR@m* is the percentage of query questions whose ground-truth answer providers are in the top m of the ranking list of candidate users.

The mathematical expressions for *avgR*, *MRR* and *CDR@m* are shown as follows.

$$AvgR = \frac{1}{|Q|}\sum_{q\in Q} r^q_{u_{true}}$$

$$MRR = \frac{1}{|Q|}\sum_{q\in Q} \frac{1}{r^q_{u_{true}}}$$

$$CDR@m = \frac{|\{q \in Q | r^q_{u_{true}} \leq m\}|}{|Q|}$$

Here, q is the query question from the question set Q. The expression $r^q_{u_{true}}$ denotes the rank of the ground-truth user u_{true} among the candidate users for question q.

The higher the values of *MRR* and *CDR*, the better the performance is, while it is contrary for *avgR*.

[11] The real ranking for ground-truth user should be 1.

4.2.1 Performance under Type 1

In Type 1, the candidate users share the same names. Table 1(a) shows the results for all the methods on *MathOverflow* dataset, as for the most likely user prediction, *relTagVec* method performs best with promising accuracy value 0.8625, which is much more competitive than the baselines. For the performance on the ranking of ground-truth users, *relTagVec* is still superior to others in terms of avgR, MRR, CDR@2 and CDR@5. In addition, both *relTitle-Max* and *relTitle-Avg* methods perform better than *random* method. And *relTitle-Max* method can yield more accurate results than *relTitle-Avg*.

In Table 1(b), we can observe that *relTagVec* method still performs better than the baselines on *Cooking* dataset, and *random* method is the worst choice again. As for Title-based methods, *relTitle-Max* is still superior to *relTitle-Avg* especially on accuracy.

As for the performance on *Travel* dataset shown in Table 1(c), it can be seen that *relTagVec* method still yields superior results in terms of all the metrics. By contrast, *random* is less competitive. Note that their CDR@5 values are all 1, which means that all the questions whose ground-truth answer providers are in the top 5 of the candidate list.

It is obvious from Table 1 that *relTagVec*, *relTitle-Max* and *relTitle-Avg* can effectively disambiguate the user names given the query question with regard to different evaluation metrics. By contrast, *relTagVec* performs best in Type 1.

4.2.2 Performance under Type 2

Different from Type 1, given a question, under Type 2, the querying user name only contains one word, which is usually viewed as the first name of a user. In such case, the candidate set is composed of all the users with the same first name. Accordingly, the user name will be more ambiguous with larger candidate set.

As can be seen from Table 2(a) that our *relTagVec* method still shows very promising performance, which outperforms the baseline methods in terms of all the listed evaluation metrics on MathOverflow dataset. Among the baselines, *random* method yields very low accuracy. As for the two title-based methods, *relTitle-Max* is still better than *relTitle-Avg*.

From Table 2(b) and Table 2(c), it tends to the similar conclusion that our *relTagVec* method performs better than the baselines on both *Cooking* and *Travel* datasets with acceptable performance.

Overall, *relTagVec* outperforms baseline methods under both types. Comparing Table 1 with Table 2 on each dataset, we can easily notice that the performance under Type 2 is reduced on each dataset with regard to nearly all the metrics, which is in accord with the fact that the user names (only given names) are more ambiguous. Moreover, the performance on Travel dataset is better than that on Cooking set in both types, which can be partly explained by Figure 1(a) and Figure 1(c), where the user names are less ambiguous in Travel community than Cooking Community, hence the performance is better on Travel dataset.

Error Analysis: We perform error analysis for *relTagVec* method and find that some candidate users share very similar values of $relevance(u, q)$, which can increase error rate and the difficulty in identifying target users.

5 Conclusions

The rapid growth of social question answering services comes with the contributions from the increasing number of registered members. Accordingly, the phenomenon about users with the same user names is getting more and more prevalent. If a user name is shared by many people in the community, once you input the user name, the system will display all the related users, in this case, it will get difficult to find out the target user. In this paper, given a question, we focus on the user name disambiguation of potential answer providers in CQA. We utilize the tag information of both users and the query question to compute the relevance values. Then the user with highest relevance is viewed as the target user. We also recommend the possible ranked user list when there are a great number of candidates. In addition, the title-based methods are introduced in evaluation. Experimental analysis on three CQA datasets show that our *relTagVec* method is simple but very effective in user name disambiguation.

There are some directions needing further investigation. First, there are other kinds of ambiguous types to consider, like misspelling. Second, it is interesting to try other ways to compute the relevance between a user and a question.

References

Xin Cao, Gao Cong, Bin Cui, and Christian S Jensen. 2010. A generalized framework of exploring category information for question retrieval in community question answer archives. In *Proceedings of the 19th international conference on World wide web*, pages 201–210. ACM.

Anderson A Ferreira, Adriano Veloso, Marcos André Gonçalves, and Alberto HF Laender. 2010. Effective self-training author name disambiguation in scholarly digital libraries. In *Proceedings of the 10th annual joint conference on Digital libraries*, pages 39–48. ACM.

Jinwen Guo, Shengliang Xu, Shenghua Bao, and Yong Yu. 2008. Tapping on the potential of q&a community by recommending answer providers. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 921–930. ACM.

Hui Han, Hongyuan Zha, et al. 2005. Name disambiguation in author citations using a k-way spectral clustering method. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05)*, pages 334–343.

Jing Liu, Young-In Song, and Chin-Yew Lin. 2011. Competition-based user expertise score estimation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 425–434. ACM.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Aditya Pal and Joseph A Konstan. 2010. Expert identification in community question answering: exploring question selection bias. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1505–1508. ACM.

Qiongjie Tian, Peng Zhang, and Baoxin Li. 2013. Towards predicting the best answers in community-based question-answering services. In *Seventh International AAAI Conference on Weblogs and Social Media*, pages 725–728.

Pucktada Treeratpituk and C Lee Giles. 2009. Disambiguating authors in academic publications using random forests. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 39–48. ACM.

Baoguo Yang and Suresh Manandhar. 2014. Exploring user expertise and descriptive ability in community question answering. In *Proceedings of 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 320–327. IEEE.

Jun Zhang, Mark S Ackerman, and Lada Adamic. 2007. Expertise networks in online communities: structure and algorithms. In *Proceedings of the 16th international conference on World Wide Web*, pages 221–230. ACM.

Baichuan Zhang, Tanay Kumar Saha, and Mohammad Al Hasan. 2014a. Name disambiguation from link data in a collaboration graph. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 81–84. IEEE.

Kai Zhang, Wei Wu, Haocheng Wu, Zhoujun Li, and Ming Zhou. 2014b. Question retrieval with high quality answers in community question answering. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, pages 371–380. ACM.

Guangyou Zhou, Siwei Lai, Kang Liu, and Jun Zhao. 2012a. Topic-sensitive probabilistic model for expert finding in question answer communities. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1662–1666. ACM.

Zhi-Min Zhou, Man Lan, Zheng-Yu Niu, and Yue Lu. 2012b. Exploiting user profile information for answer ranking in cqa. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 767–774. ACM.