# From Discussion to Wisdom: Web Resource Recommendation for Hyperlinks in Stack Overflow

Jing Li, Zhenchang Xing, Deheng Ye and Xuejiao Zhao
School of Computer Engineering, Nanyang Technological University, Singapore
{jli030, zcxing, ye0014ng, xjzhao}@ntu.edu.sg

## ABSTRACT

Stack Overflow has been providing question and answering service for 7 years. It has become a tremendous knowledge repository for developers' thoughts and practices. Hyperlinks in discussion threads of Stack Overflow are essential knowledge entities for programming on the Web, such as a software library, an API documentation, a code example, or a tutorial. Tens of millions of hyperlinks are disseminated in Stack Overflow, while wisdom on what web resources have been highly recognized by the community is implicit in millions of discussion threads. In this paper, we develop the WisLinker framework that extracts knowledge from discussion, then turns knowledge into wisdom by learning through the knowledge dissemination history. With this wisdom, for a specific hyperlink that users are concerned with, WisLinker can recommend web resources highly recognized by the Stack Overflow community. We evaluate the validity of WisLinker in an open-ended setting using Stack Overflow data dump. We also implement a browser extension for live recommendation of web resources while users browse web pages. WisLinker could enable more efficient exploratory search and information discovery of programming-related web resources.

## CCS Concepts

**•Information systems → Collaborative and social computing systems and tools; Web applications;**

## Keywords

Discussion threads, hyperlinks, knowledge and wisdom, web resource recommendation, browser extension

## 1. INTRODUCTION



*SAC 2016, April 04-08, 2016, Pisa, Italy*



DOI: http://dx.doi.org/10.1145/2851613.2851815

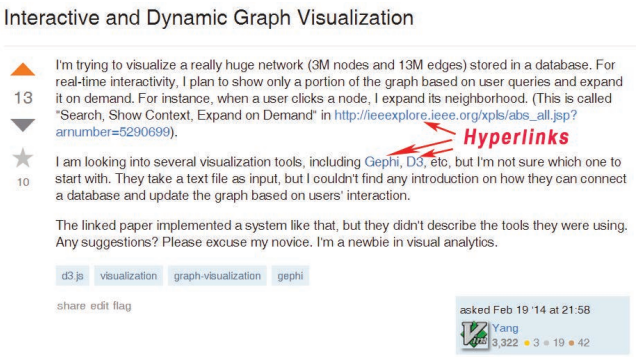



**Figure 1: Motivating example**

Today, millions of developers post questions, answers and comments related to computer programming in Stack Overflow [1, 8]. During discussion, users frequently reference to hyperlinks of web resources in their posts as shown in Figure 1. These hyperlinks are essential knowledge entities in the discussion, referencing to various types of programming knowledge on the Web, such as software, libraries, API, code examples, blogs, tutorials, and books. As data volumes increase in Stack Overflow, knowledge accumulates fast. But wisdom on what web resources are highly-recognized by the community and are pertinent to the hyperlinks at hand is buried in the sea of discussion threads.

As shown in Figure 1, the user is a "newbie in visual analytics". He is looking for scalable visualization tools for interactive and dynamic graph visualization. He references to three web resources in the post, including one paper explaining the concept "Search, Show Context, Expand on Demand", and two visualization tools "Gephi" and "D3" that he is investigating for the task. We say that this user disseminated three hyperlinks in the discussion thread, creating a publicly available record of three knowledge entities in the system. Furthermore, we see that the three knowledge entities are correlated. "Gephi" and "D3" are two alternative tools that may support "Search, Show Context, Expand on Demand" described in the paper.

In fact, as of September 2014, two of these three knowledge entities ("Gephi" and "D3") have been disseminated 107 and 517 times in 89 and 426 discussion threads, respective-

ly. They co-occur with 397 and 1579 other web resources in these discussion threads. For the question in Figure 1, answerers reference to (i.e., disseminate) 16 web resources, including 12 tools, 2 blogs, 1 Stack Overflow post, and 1 code repository in the answers and comments. 5 out of these 16 web resources frequently co-occur with the two hyperlinks of "Gephi" and "D3" in many discussion threads before the question in Figure 1 was asked. That is, knowledge that the user is asking for is highly likely already out there. Unfortunately, it is impossible for the user to distill this knowledge from thousands of web resources disseminated in hundreds of discussion threads.

This motivating example illustrates the gap between the knowledge embodied in the discussion treads on Stack Overflow and the wisdom on what knowledge is useful. According to Bellinger et al. [3], wisdom is knowledge and judgment about excellence through users' experience. When the user is concerned with a hyperlink (e.g., the "Gephi" or "D3" tool), can we distill wisdom from the past experience and recommend some relevant web resources that are highly recognized by the community?

In this paper, we present the WisLinker framework to bridge the gap between the knowledge and the wisdom. In this work, we focus on a special type of knowledge in the discussion threads on Stack Overflow, i.e., hyperlinks. We propose a knowledge dissemination graph to model knowledge dissemination in discussion threads over time. Based on this graph model, we propose an algorithm to distill highly-recognized web resources for a given hyperlink. The algorithm takes into account both hyperlink co-occurrence and community-feedback (e.g., votes) on the posts where hyperlinks are referenced.

We implement a proof-of-concept tool of the WisLinker framework. The tool is a browser extension that supports live recommendation of web resources that are highly-recognized by the Stack Overflow community, when users browse a web page and point the mouse to a hyperlink in the page. Although the recommendation model is learnt from Stack Overflow data, the tool can make recommendation as long as the hyperlink in the page is in the recommendation model, even the web page is not a Stack Overflow page.

We use 6-years of Stack Overflow data (July 2008 - September 2014) to train the recommendation model and evaluate the validity of the model on 2 months of data spanning October 2014 to December 2014. That is, we learn wisdom from the past experience and evaluate the wisdom with the "future" data. The evaluation results show the WisLinker achieves $Pr@1 = 10.4\%$ and $Re@10 = 11.9\%$ in this open-ended setting.

Our contribution in this paper is three-fold:

- Propose a knowledge dissemination model and an algorithm to mine wisdom from the model.
- Conduct a large-scale experiment to evaluate the proposed model and algorithm.
- Implement a proof-of-concept tool of the proposed WisLinker framework.

Section 2 reviews related work. Section 3 presents the WisLinker framework. Section 4 reports the evaluation of the WisLinker framework. Section 5 presents the tool implementation. Section 6 discusses implication and insight of WisLinker. Section 7 concludes the work and discusses our future plan.

## 2. RELATED WORK

In this section, we review related works from aspects of recommendation systems for software engineering, and managing knowledge and wisdom.

**Recommendation systems for software engineering.** Question and answer (Q&A) sites have attracted much research interest [1, 11]. In particular, Stack Overflow is a popular destination where users seek help for software development problems. Luca et al [6] present Prompter, a self-confident recommender system that automatically searches and identifies relevant Stack Overflow discussions, given the code context in the IDE. Pedro et al. [7] propose RankSLDA, recommending question for collaborative question answering systems. Wang et al. [12] present an enhanced tag recommendation system to improve the quality of tags in software information sites. Alberto et al. [2] introduce Seahawk, an Eclipse plugin to integrate Stack Overflow crowd knowledge in the IDE. Alexey et al. [14] present a code search and recommendation tool which brings together social media and code recommendation systems.

Existing studies mainly focus on tag and code example recommendation. In contrast, our study distills wisdom of crowds from knowledge disseminated on Stack Overflow over time. In particular, we distill web resources that are highly-recognized by the Stack Overflow from over 10 million hyperlinks in Stack Overflow. To the best of our knowledge, this study is the first attempt to model and analyze hyperlink dissemination patterns in Stack Overflow. We also develop a web resource recommendation tool based on the mined wisdom.

Compared the recommendation by Stack Overflow which only links current post to other relevant internal posts, WisLinker can not only recommend relevant internal posts but also external resources.

**Managing knowledge and wisdom.** There are some work studying the relationship between knowledge and wisdom. Bellinger et al. [3] discuss the relationship and difference from data, information, knowledge and wisdom. Ning et al. [15] study the wisdom web to help people achieve better ways of living, working, playing, and learning in next-generation Web. Marc et al. [4] harness knowledge from online social network such as Twitter to obtain an understanding about the collective "wisdom of crowds" [9]. They leverage the wisdom in policymaking, decision support, economic analysis, epidemic behavior analysis, and various other applications.

Different from these studies on Web search and Twitter, the discussion in Stack Overflow are domain specific and carry domain-specific knowledge for programming. Distilling the collective wisdom of crowds from millions of discussions could enable more effective management and dissemina-

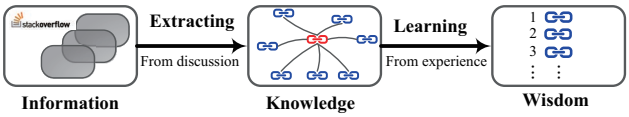



**Figure 2: The WisLinker framework**

tion of high-value programming knowledge in Stack Overflow community and beyond.

## 3. METHODOLOGY

Given a hyperlink in Stack Overflow (referred to as *seed hyperlink*), our goal in this paper is to recommend top-k relevant web resources that are highly recognized by the Stack Overflow community. Figure 2 presents a high-level overview of the WisLinker framework. The WisLinker framework first extract knowledge entities (i.e., hyperlinks in this work) and model them in a knowledge dissemination graph. Then, it mines wisdom on what web resources are most recognized by the community by analyzing both hyperlink co-occurrence and hyperlink competition.

### 3.1 Knowledge Dissemination Graph

We model the dissemination of hyperlinks in Stack Overflow in a knowledge dissemination graph as follows.

DEFINITION *1. (Discussion Thread) A discussion thread consists of a question and all its answers and comments sorted by chronological order.*

Discussion thread is the basic information unit in our model. For example, Figure 3 depicts discussion thread 1 to discussion thread $M$.

DEFINITION *2. (Knowledge Cascade) A knowledge cascade consists of all discussion threads that mention the seed hyperlink, sorted by chronological order.*

Figure 3 illustrates the knowledge cascade of a seed hyperlink (highlight in red). The thread in the second row does not include the seed hyperlink, and thus is not included in the knowledge cascade of the seed hyperlink.

DEFINITION *3. (Hyperlink Associative Network $\mathcal{G}_{H_s}$) In the knowledge cascade of the seed hyperlink $H_s$, an undirected graph $\mathcal{G}_{H_s} = (V, E)$ denotes hyperlink associative network where vertex $V$ is the set of co-occurring hyperlinks (highlight in blue) of $H_s$ and $E$ represents the set of edges from node $H_s$ to nodes in set $V$.*

### 3.2 Associative-Edge Weight Computation

As shown in Figure 3, the hyperlinks may occur in the question, answer, or comment in a discussion thread. We refer to these three places as three types of hyperlink location. Thus, for the hyperlink associative network $\mathcal{G}_{H_s}$, the edges fall into three categories, i.e., $E = E_Q \bigcup E_A \bigcup E_C$, where $E_Q$,

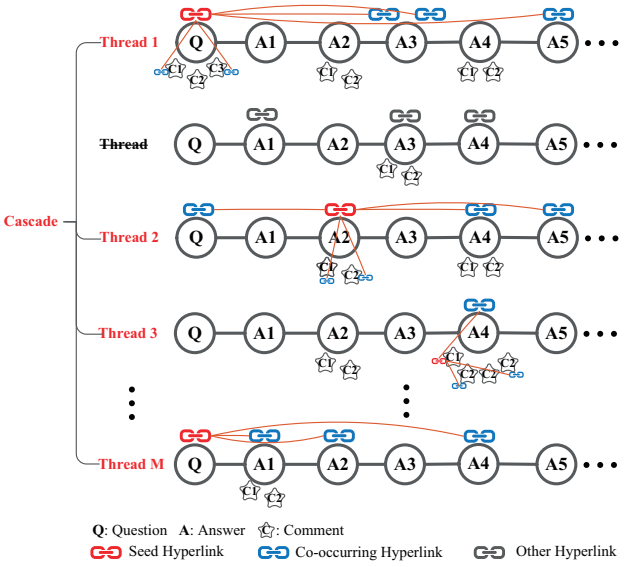



**Figure 3: Illustration of knowledge dissemination graph**

$E_A$ and $E_C$ are disjoint sets, and they represents associative edges whose seed hyperlink node $H_s$ occurs in question (e.g., Thread 1 and Thread M), answer (Thread 2), and comment (Thread 3), respectively.

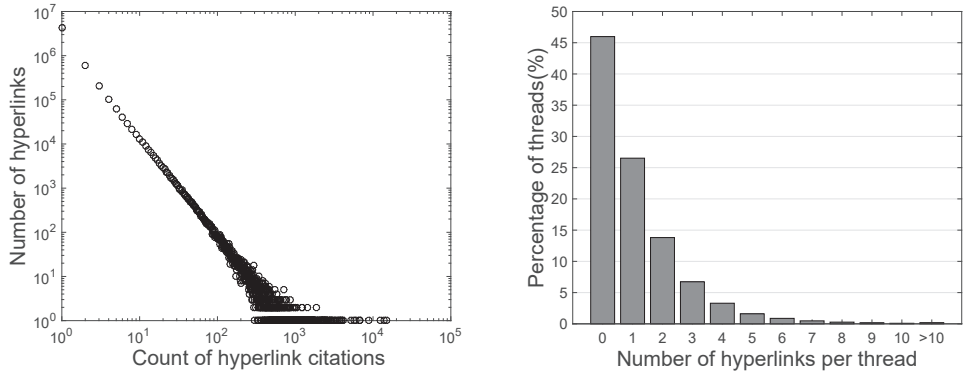
Our recommendation model is based on the following two heuristics. First, the relationships between a seed hyperlink and its co-occurring hyperlinks are different when the seed hyperlink $H_s$ occurs in different type of location. This heuristic is based on the consideration that the asker very likely follows up all the answers, but an answerer or a commenter may not pay attention to others answers. Second, for the co-occurring hyperlinks of the seed hyperlink, the scores (e.g., votes) of posts in which the hyperlinks are referenced reflect the competition among the co-occurring hyperlinks in a discussion thread. This heuristic is based on the observation that hyperlinks in high-score posts would be more valuable to the community than those in low-score posts.

Based on these two heuristics, we compute weight of three different types of associative edges as follows.

**Weights of Edges in $E_Q$.** Consider the scenario that the seed hyperlink $H_s$ occurs in a question as shown Thread 1 in Figure 3. In order to take into account the competition among co-occurring hyperlinks in different answers or comments, we assign the score of the answer or comment in which a co-occurring hyperlink is referenced as the score of this co-occurring hyperlink. In the current implementation, we use vote (i.e., upvote-downvote) as score. But WisLinker can use other scores (e.g., view count, number of favorite).

We only consider the competition of co-occurring hyperlinks in the same type of hyperlink location (i.e., competition among co-occurring hyperlinks in comments, or competition among hyperlinks in answers). This is because the scores of answers and comments are often very different in scale and most of comments have no score.

Let $w_{s,i}$ be the edge weight from node $H_s$ to node $H_i$ (the




(a) Training Dataset (b) Testing Dataset

**Figure 4: Statistics of dataset**

*i-th* co-occurring hyperlink in a discussion thread). Assume $H_i$ occurs in an answer of the discussion thread, $Score_i$ is the score of node $H_i$, $Score_A$ represents score vector of all the answers of the discuss thread, then the weight $w_{s,i}$ can be computed as $w_{s,i} = \frac{score_i - \min(score_A)}{max(score_A) - \min(score_A)}$. That is, the edge weight $w_{s,i}$ is a normalized value in $[0, 1]$ that reflects the relative score of node $H_i$ compared with other co-occurring hyperlinks. For $H_i$ occurring in a comment, the same computation can be performed.

**Weights of Edges in $E_A$.** Consider the scenario that the seed hyperlink $H_s$ occurs in an answer as shown Thread 2 in Figure 3. Let node $H_q$ be a hyperlink occurring in the question of this discussion thread. Based on the consideration that information in an answer should be directly related to information in the question, we set the weight of the edge between node $H_s$ and node $H_q$ at 1, i.e., $w_{s,q} = 1$. For the co-occurring hyperlinks $H_i$ in other answers of this discussion thread, we can obtain the edge weight between node $H_s$ and node $H_i$ as $w_{s,i} = \frac{score_i - \min(score_A)}{max(score_A) - \min(score_A)}$. For $H_i$ occurring in a comment, the same computation can be performed.

**Weights of Edges in $E_C$.** Consider the scenario that the seed hyperlink $H_s$ occurs in a comment as shown Thread 3 in Figure 3. Let $P$ be the post on which this comment is made. Let node $H_p$ be a hyperlink occurring in the post $P$. Based on the consideration that information in a comment should be directly related to information in the post $P$, we set the edge weight between node $H_s$ and node $H_p$ at 1, i.e., $w_{s,p} = 1$. For the co-occurring hyperlinks $H_i$ in other comments of the post, we obtain the edge weight between between node $H_s$ and node $H_i$ as $w_{s,i} = \frac{score_i - \min(score_C)}{max(score_C) - \min(score_C)}$, where $score_C$ represents score vector of all the comments in the post $P$.

For the hyperlink associative network graph $\mathcal{G}_{H_s} = (V, E)$ pertaining to the seed hyperlink $H_s$, WisLinker iterates through all discussion threads in the knowledge cascade. It recommends top-$k$ web resources that have the highest edge weight with the seed hyperlink $H_s$ as the community-recognized web resources relevant to the seed hyperlink $H_s$. For $H_s$, the recommendation is shown in Algorithm 1.

**Algorithm 1:** Recommendation for $H_s$

**Data**: Seed hyperlink $H_s$
**Result**: Top-$k$ resources as recommendation results
$M \leftarrow$ discussion threads to be extracted;
**while** *$M$ is not empty* **do**
  thread $m \leftarrow$ pop head node of $M$;
  **foreach** *edge $E_i$ in thread $m$* **do**
    **if** $E_i \in E_Q$ **then**
      Compute $w_{s,i}$ ;
      $\mathcal{G}_{H_s}$.add_weighted_edge($w_{s,i}$);
    **else if** $E_i \in E_A$ **then**
      Compute $w_{s,q}$ , $w_{s,i}$;
      $\mathcal{G}_{H_s}$.add_weighted_edge($w_{s,q}$);
      $\mathcal{G}_{H_s}$.add_weighted_edge($w_{s,i}$);
    **else**
      Compute $w_{s,p}$ , $w_{s,i}$;
      $\mathcal{G}_{H_s}$.add_weighted_edge($w_{s,p}$);
      $\mathcal{G}_{H_s}$.add_weighted_edge($w_{s,i}$);
**return** *top-k resources in this cascade based on weighting scores*

# 4. EVALUATION

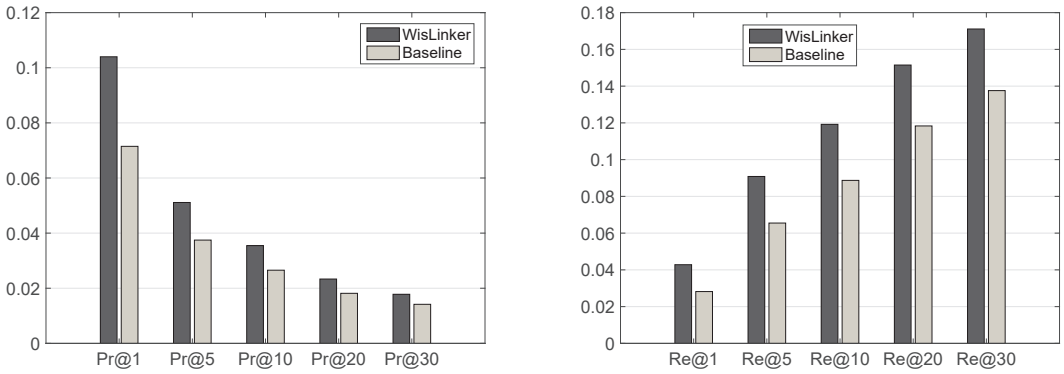
In this section, we evaluate our proposed method using Stack Overflow data dump. We first detail the the experimental settings, then report our evaluation results.

## 4.1 Experimental Setup

**Dataset.** We conducted our evaluation on the data dump of Stack Overflow made available by the site [1]. The training dataset spans 6 years from July 2008 to September 2014. In total, we collected 7,990,787 questions, 13,684,117 answers, 32,506,636 comments, and 5,522,886 unique hyperlinks. Figure 4(a) plots the hyperlink citation distribution where a power-law distribution is observed as expected.

The testing dataset spans two months October 1st, 2014 to December 1st, 2014. We collected 420,637 discussion threads and 467,413 hyperlinks. Figre 4(b) shows the statis-

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




(a) Precision@k (b) Recall@k

**Figure 5: Performance of methods with $k = \{1, 5, 10, 20, 30\}$**

tic of the number of hyperlinks per discussion thread. We can see that 45.98% of discussion threads have no hyperlinks and 26.52% of discussion threads only contain 1 hyperlink. To evaluate our recommendation method, we consider only discussion threads with 2 or more hyperlinks.

**Seed hyperlinks.** When training the recommendation model, we consider only hyperlinks that are referenced more than 4 times. In the testing dataset, 104,012 hyperlinks appear in the recommendation model and in the 62,351 discussion threads with 2 or more hyperlinks. We use these 104,012 hyperlinks as the seed hyperlinks to evaluate the performance of the WisLinker's recommendation.

**Ground truth.** For a seed hyperlink $H_s$ in a discussion thread, we collect all its co-occurring hyperlinks in the thread as the ground truth.

**Metrics.** We use two metric to evaluate the quality of web resource recommendation: $Precision@k$ and $Recall@k$ ($Pr@k$ and $Re@k$ for short). $k = \{1, 5, 10, 20, 30\}$ is the number of top-ranked recommended web resources. Let $H_r$ be the set of top-$k$ recommended web resoruces and $H_g$ be the set of ground-truth web resources for the seed hyperlink $H_s$. $Pr@k$ for $H_s$ is $\frac{H_r \cap H_g}{k}$. $Re@k$ for $H_s$ is $\frac{H_r \cap H_g}{H_g}$. The values reported are averaged values over all 62,351 discussion threads in the testing dataset.

**Baseline.** Our proposed method takes into account competition of web resources among posts. Thus, the baseline method is that the recommendation considers only the absolute score of a post and does not consider the competition among posts. We name our proposed method as WisLinker method as opposed to the baseline method.

## 4.2 Evaluation Results

First, we report the overall performance of the proposed WisLinker method and the baseline method. Then, we evaluate the impact of different citation levels.

### 4.2.1 Overall Performance

Figure 5 reports $Pr@k$ and $Re@k$ of the WisLinker method and the baseline method. From this figure, we make following observations:

- The WisLinker method outperforms the baseline method at all different $k$. This shows that taking into account competition among posts in which hyperlinks are referenced can improve the quality of web resource recommendation.
- The best precision is 10.4% at $k = 1$ and the best recall is 17.4% at $k = 30$. As expected, the precision drops as the value of $k$ increases, while the recall increases as the value of $k$ increases.
- Although the precision and recall are small, we deem this is a satisfactory and acceptable performance, because we use real posts in Stack Overflow are testing set. The reference of hyperlinks can be affected by many factors, such as variation of question topics, the expertise of answerers, and the emergence of new technologies. In such a open-ended setting, our WisLinker method achieves precision 10.4%@1 and recall 11.9%@10.

### 4.2.2 Impact of Citations

The WisLinker method is based on hyperlink co-occurrence in discussion threads. As such, the number of hyperlink citations may affect the performance of recommendation.

As shown in Figure 4(a), the number of hyperlinks by citation count obeys a pow-law distribution. It reveals that most of hyperlinks are only referenced in a small number of times and a fraction of hyperlinks are referenced in a large number of times. In this study, we split the number of citations into 4 levels: "$5 - 50$", "$51 - 100$", "$101 - 500$" and "$> 500$". In the testing dataset, the number of seed hyperlinks at these 4 different levels are 57886, 11736, 20232 and 14158, respectively.

Table 1 and Table 2 shows the precision and recall at the 4 citation levels, respectively. From these two tables, we can observe that:

- The precision and recall increases as the number of hyperlink citations increases. For hyperlinks referenced

Table 1: Precision at different citation levels

| Citation levels | | 5-50 | 51-100 | 101-500 | >500 |
|---|---|---|---|---|---|
| # of hyperlinks | | 57886 | 11736 | 20232 | 14158 |
| Pr@1 | WisLinker | 0.078 | 0.10 | 0.12 | 0.17 |
| | Baseline | 0.050 | 0.056 | 0.079 | 0.15 |
| Pr@5 | WisLinker | 0.037 | 0.049 | 0.061 | 0.089 |
| | Baseline | 0.028 | 0.030 | 0.039 | 0.074 |
| Pr@10 | WisLinker | 0.026 | 0.031 | 0.041 | 0.062 |
| | Baseline | 0.021 | 0.021 | 0.026 | 0.046 |
| Pr@20 | WisLinker | 0.018 | 0.018 | 0.024 | 0.036 |
| | Baseline | 0.014 | 0.016 | 0.017 | 0.027 |
| Pr@30 | WisLinker | 0.014 | 0.015 | 0.018 | 0.025 |
| | Baseline | 0.012 | 0.013 | 0.013 | 0.019 |

Table 2: Recall at different citation levels

| Citation levels | | 5-50 | 51-100 | 101-500 | >500 |
|---|---|---|---|---|---|
| # of hyperlinks | | 57886 | 11736 | 20232 | 14158 |
| Re@1 | WisLinker | 0.031 | 0.041 | 0.051 | 0.077 |
| | Baseline | 0.019 | 0.021 | 0.031 | 0.064 |
| Re@5 | WisLinker | 0.067 | 0.091 | 0.11 | 0.15 |
| | Baseline | 0.052 | 0.056 | 0.069 | 0.11 |
| Re@10 | WisLinker | 0.088 | 0.11 | 0.14 | 0.19 |
| | Baseline | 0.074 | 0.078 | 0.092 | 0.14 |
| Re@20 | WisLinker | 0.11 | 0.13 | 0.17 | 0.22 |
| | Baseline | 0.10 | 0.10 | 0.11 | 0.16 |
| Re@30 | WisLinker | 0.13 | 0.14 | 0.18 | 0.24 |
| | Baseline | 0.11 | 0.12 | 0.13 | 0.17 |

more than 500 times, the WisLinker method achieves 17.83%@1 precision and 19.45%@10 recall.

- At all the citation levels, the WisLinker method outperforms the baseline methods.
- Comparing the lowest and highest citation levels "5 − 50" versus "> 500", the performance of citation level "> 500" significantly better than that of the level "5 − 50". In most cases, the precision and recall are doubled between the two levels.

## 5. LIVE WEB RESOURCE RECOMMENDATION

We apply WisLinker to train a web resource recommendation model based on the Stack Overflow data dump from July 2008 to September 2014. Our goal is to recommend high-quality and trustworthy web resources while users browse web pages and would like to see some relevant web resources. In order to make real-time recommendations for the hyperlinks in Stack Overflow, we implement a browser extension integrating the recommendation model of WisLinker [2].

Figure 6 shows the interface of the WisLinker browser extension. When the user hovers the mouse over the hyperlinks of "D3" in the question, a popup view shows the WisLinker recommendations for the hyperlinks of "D3". Part 1 is the basic information about the citation history of the hyperlink. The number "517" shows that this hyperlink was referenced 517 times in Stack Overflow. The following bar chart represents the temporal trend of these 517 citations. Each bar in

[2]A demonstration video of the tool can be found at https://www.youtube.com/watch?v=3W9SFvu9EXI

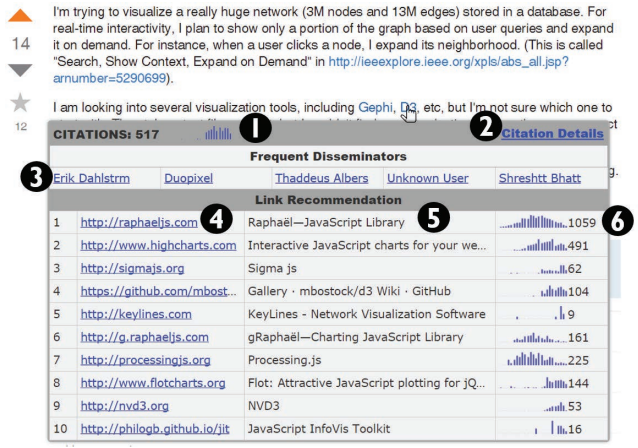



**Figure 6: The interface of the WisLinker browser extension**

the chart represents the number of citations in every three months. From the chart, users can clearly see the temporal pattern of the hyperlink of "D3", for example, when is the web resource frequently referenced, is it becoming more popular or less popular in Stack Overflow.

Part 2 links to the WisLinker website for viewing citation details, such as distribution of time in a day when the hyperlink is referenced, and tag clouds of the posts referencing to the hyperlink. Part 3 lists the top-5 frequent disseminators of the hyperlink. These disseminators may be experts in the domain relevant to the hyperlink. WisLinker provides links to the home pages of these users on Stack Overflow.

Link Recommendation section presents web resources recommended by the WisLinker recommendation model. The WisLinker browser extension only list the top-10 resources. For each web resource, WisLinker displays the following information: the hyperlink to the web resource in part 4, the title of the link in part 5, and the temporal citation trend of the recommended web resource in part 6. The title allows users to quickly judge the relevance of the recommended web resource. The temporal citation trend help users compare the popularity of the recommended web resources. In this example, the user can find many alternative JavaScript data visualization libraries. These tools are frequently referenced in Stack Overflow discussions. He can also observe the popularity of these tools. For example, "Raphael" has been frequently referenced in the past, but its popularity seems dropping recently.

## 6. DISCUSSION

In this section, we discuss some implications and insights of our study.

**Supporting exploratory search.** Gary et al. [5] describe three kinds of search activities: lookup, learn, and investigate. They highlight exploratory search [13] as especially pertinent to the learn and investigate search activities. A person's knowledge is limited, compared with that of the crowd. WisLinker turns knowledge embodied in crowd-generated content into wisdom of crowds. It allows the user

to easily reach more web resources relevant to the content he is viewing. Note that the recommendation is backed by the wisdom of crowds. As such, WisLinker could enhance the user experience during exploratory search, by bringing serendipitous discovery in the learn and investigate search activities.

**Encouraging web resource dissemination.** Stack Overflow encourages users referencing relevant web resources in their questions and answers, as this practice can enhance question and answer quality and user satisfaction [1, 10]. From the Figure 4(b), 26.52% of discussion threads only have 1 hyperlink and 45.98% of discussion treads have no hyperlinks. We hypothesize that certain percentage of these discussion threads could be enhanced by including some relevant web resources to support the discussions. The knowledge base of WisLinker could support this task. However, WisLinker currently requires a seed hyperlink to trigger the recommendation. This limitation could be addressed by linking a hyperlink to a named entity. For example, by analyzing the hyperlink anchor text, we can know that the hyperlink represents a tool, such as "Gephi" or "D3" in the motivating example. Then, WisLinker could make recommendation when the user points to a word that matches the tool name. Of course, entity linking is a challenging task as entities and hyperlinks in Stack Overflow discussions are often mentioned in many variations, such as "d3.js", "d3js", or even "this", "here".

# 7. CONCLUSION AND FUTURE WORK

In this paper we present our WisLinker framework for turning knowledge entities (e.g., hyperlinks) disseminated in Stack Overflow discussions into wisdom of crowds. WisLinker is based on the fact that similar but non-duplicated programming issues are frequently discussed in Stack Overflow, and relevant web resources are repeatedly referenced in the discussions. Taken in aggregate, wisdom of crowds can be learned from the past knowledge dissemination patterns.

WisLinker captures the knowledge dissemination history using a graph model. Its mining algorithm takes into account both co-occurrence and competition of web resources in the discussion threads to discern web resources that are highly-recognized by the community. We evaluate the validity of WisLinker assumption and the quality of its recommendation using a large-scale Stack Overflow data dump. The results show that WisLinker achieves promising recommendation accuracy in an open-ended test dataset. We also presented the WisLinker browser extension that can enhance the user experience via live recommendation of web resources in web search and browsing.

In the future, we will investigate the use of WisLinker for exploratory search and domain-specific named entity recognition and linking methods for enhancing web resources dissemination in Stack Overflow.

## Acknowledgments

The authors thank the reviewers for their helpful comments. This work is partially supported by MOE AcRF Tier 1 grant M4011165.020 and MOE Scholarships.

# 8. REFERENCES

[1] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. Discovering value from community activity on focused question answering sites: a case study of stack overflow. In *Proc. of KDD*, pages 850–858. ACM, 2012.

[2] A. Bacchelli, L. Ponzanelli, and M. Lanza. Harnessing stack overflow for the ide. In *Proc. of the Third International Workshop on Recommendation Systems for Software Engineering*, pages 26–30, 2012.

[3] G. Bellinger, D. Castro, and A. Mills. Data, information, knowledge, and wisdom. 2004.

[4] M. Cheong and V. Lee. Integrating web-based intelligence retrieval and decision-making from the twitter trends knowledge base. In *Proc. of the 2nd ACM workshop on Social web search and mining*, pages 1–8, 2009.

[5] G. Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.

[6] L. Ponzanelli, G. Bavota, M. Di Penta, R. Oliveto, and M. Lanza. Prompter: A self-confident recommender system. In *Proc. of ICSME*, pages 577–580, 2014.

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

[8] D. Schenk and M. Lungu. Geo-locating the knowledge transfer in stackoverflow. In *Proc. of the International Workshop on Social Software Engineering*, pages 21–24. ACM, 2013.

[9] J. Surowiecki. *The wisdom of crowds*. Anchor, 2005.

[10] C. Treude, O. Barzilay, and M.-A. Storey. How do programmers ask and answer questions on the web?: Nier track. In *Proc. of ICSE*, pages 804–807, 2011.

[11] S. Wang, D. Lo, and L. Jiang. An empirical study on developer interactions in stackoverflow. In *Proc. of SAC*, pages 1019–1024, 2013.

[12] S. Wang, D. Lo, B. Vasilescu, and A. Serebrenik. Entagrec: an enhanced tag recommendation system for software information sites. In *Proc. of ICSME*, pages 291–300, 2014.

[13] R. W. White and R. A. Roth. Exploratory search: Beyond the query-response paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 1(1):1–98, 2009.

[14] A. Zagalsky, O. Barzilay, and A. Yehudai. Example overflow: Using social media for code recommendation. In *Proc. of the Third International Workshop on Recommendation Systems for Software Engineering*, pages 38–42, 2012.

[15] N. Zhong, J. Liu, and Y. Yao. In search of the wisdom web. *Computer*, (11):27–31, 2002.