2016 IEEE International Conference on Software Maintenance and Evolution

TechLand: Assisting Technology Landscape Inquiries with Insights from Stack Overflow

Chunyang Chen, Zhenchang Xing, Lei Han
School of Computer Engineering, Nanyang Technological University, Singapore
chen0966@e.ntu.edu.sg; zcxing@ntu.edu.sg; tomhanlei@ntu.edu.sg

***Abstract*—Understanding the technology landscape is crucial for the success of the software-engineering project or organization. However, it can be difficult, even for experienced developers, due to the proliferation of similar technologies, the complex and often implicit dependencies among technologies, and the rapid development in which technology landscape evolves. Developers currently rely on online documents such as tutorials and blogs to find out best available technologies, technology correlations, and technology trends. Although helpful, online documents often lack objective, consistent summary of the technology landscape. In this paper, we present the *TechLand* system for assisting technology landscape inquiries with categorical, relational and trending knowledge of technologies that is aggregated from millions of Stack Overflow questions mentioning the relevant technologies. We implement the *TechLand* system and evaluate the usefulness of the system against the community answers to 100 technology questions on Stack Overflow and by field deployment and a lab study. Our evaluation shows that the *TechLand* system can assist developers in technology landscape inquiries by providing direct, objective, and aggregated information about available technologies, technology correlations and technology trends.**

I. Introduction

A diverse set of technologies[1] is available for use by developers and that set continues growing. Developers are expected to have a good understanding of the technology landscape in their work. However, even for experienced developers, it can be difficult to keep pace with the rapid progress in which technology landscape evolves [1]. To understand the technology landscape, developers have three types of information needs: 1) *what are the best available technologies?* 2) *how do technologies correlate with each other?* and 3) *what is the trend of technology used by the community?*

To address these information needs, developers often resort to two information sources on the Web. First, developers often share their understanding of the technology landscape in online articles, such as *best PHP framework for 2015*[2], *20 best JavaScript charting libraries*[3], *Python's SQLAlchemy vs other ORMs*[4]. Second, developers can seek answers from community-curated list of technologies (e.g., *awesome PHP*[5]), or from the Q&A websites such as Stack Overflow or Quora (e.g., *which framework is best for web development in PHP*[6]). These online technology landscape documents are indexable by search engines, thus enabling developers to find answers to their technology landscape inquiries.

However, online technology landscape documents share a set of limitations and fall short to address the developers' information needs in technology landscape inquiries. First, they lack an objective, consistent summary of the best available technologies. For example, for the query like "data visualization" or "data visualization tools", Google returns many similar online articles, such as *The 14 Best Data Visualization Tools*[7] and *8 excellent open source data visualization tools*[8]. These articles offer overlapping but different opinions for the "best" data visualization tools. Developers have to determine their agreements and discrepancies. Second, an online document usually focuses on a specific technology, not a set of correlated technologies. For example, to learn what ORM libraries popular PHP web development frameworks use, one may have to aggregate information from several articles like *best PHP framework for 2015*[9] and *best available PHP ORM libraries*[10]. Such information aggregation is opportunistic. Third, technology landscape is in a constant state of change. However, online documents usually provide a snapshot of the technology landscape at a specific time. Few of them analyze the trend of technology adoption over time. Even when they do, the analysis often lacks objective support.

To overcome the above limitations, we propose the *TechLand* system that exploits the crowdsourced knowledge from Stack Overflow, and assists technology landscape inquiries with an informative summary of available technologies, technology correlations, and technology trend. Inspired by the recent empirical studies on Stack Overflow [2], [1], [3], [4], we consider question tags as technologies used by the community, and propose association rule mining and natural language processing methods to mine relational and categorical knowledge of technologies. Our recent study suggests that tag usage statistics over time provides a good estimate of the trend of developers' interests in relevant technologies [5]. Therefore, we collect tag usage statistics over time to estimate the trend of technologies used by the community. The mined

[1]In this paper, we use the term *technology* to broadly refer to concepts, programming languages, platforms, libraries/tools/frameworks, and APIs (at class or module level) for software engineering.

[2]http://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/

[3]http://thenextweb.com/dd/2015/06/12/20-best-javascript-chart-libraries/

[4]http://pythoncentral.io/sqlalchemy-vs-orms/

[5]https://github.com/ziadoz/awesome-php

[6]https://www.quora.com/Which-framework-is-best-for-web-development-in-PHP

[7]http://thenextweb.com/dd/2015/04/21/the-14-best-data-visualization-tools/

[8]http://opensource.com/life/15/6/eight-open-source-data-visualization-tools

[9]http://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/

[10]http://www.gajotres.net/best-available-php-orm-libraries-part-1/

978-1-5090-3806-0/16 $31.00 © 2016 IEEE
DOI 10.1109/ICSME.2016.17

IEEE computer society

relational, categorical and trending knowledge of technologies constitutes a *knowledge graph* of technologies. Based on this knowledge graph, *TechLand* augments Google search with a set of libraries, languages and concepts, an interactive graph view of technology correlations, and a line chart of technology trend that are related to the search terms (See Fig. 1).

We build a prototype of the *TechLand* system (including the backend knowledge base and the frontend browser plugin and website) and release the system for public use. The Google Analytics of the website's 6-months visit data indicates the general interests in our technology landscape service. We evaluate the usefulness of the *TechLand* system in assisting technology landscape inquiries in two studies. First, we evaluate the usefulness of the *TechLand* recommendation against the community answers to 100 technology landscape questions on Stack Overflow. Second, we evaluate the design and usability of the *TechLand* system by a user study of 12 participants. It is important to note that the knowledge that *TechLand* uses to augment technology landscape inquiries is not based on one or two persons' opinions. Instead, the knowledge is aggregated from millions of questions mentioning the relevant technologies, and thus presents an objective, consistent and organized summary for addressing the information needs in technology landscape inquiries. Furthermore, the mined knowledge can be automatically updated periodically to ensure the up-to-date view of the technology landscape.

We make the following contributions in this paper:

- We present an approach to aggregating and visualizing the crowdsourced knowledge of technologies from Stack Overflow to assist technology landscape inquiries;
- We implement and release the *TechLand* system prototype for public use;
- We report our field deployment and two empirical studies to evaluate the design and usefulness of the *TechLand* system.

II. MOTIVATING SCENARIO

This section illustrates the kinds of problems a developer (say John) may encounter when exploring the technology landscape of *unfamiliar* technologies, such as data visualization, machine learning, image processing, and how our *TechLand* system could help. We use data visualization as an example. John may have technology landscape questions, such as what are the most popular data visualization tools for different program languages, and what basic concepts and techniques does he need to understand in order to learn a particular tool. To answer these questions, John needs to find out available tools for data visualization, the trend of these tools used by the community, and the correlation between relevant concepts, techniques and tools.

A reasonable starting query is "data visualization" or "data visualization tools", which returns millions of blogs, tutorials, Q&A posts that are not grouped or organized in any meaningful way. John reads a Wikipedia page about data visualization from which he learns some relevant concepts and techniques, such as *information graphics*, *scientific visualization*, *scatter plot*, *scalable vector graphics* format. John finds a large number of articles summarizing "best" data visualization tools, which often list several to dozens of tools. He reads some recent articles and summarizes a few good candidates, such as *d3.js for JavaScript*, *matplotlib for Python*, *ggplot2 for R*, that are commonly mentioned in different articles. However, it is still unclear to him what is the trend of these tools used by the community and how different tools, concepts and techniques are correlated. To learn more, John needs to search more with the information scents he has just collected, such as "python scientific visualization", "d3.js data format". Certainly it is possible to issue queries that lead to the desired results quickly. However, more often than not the search and learning process is opportunistic in which John has to browse, read, compare and aggregate information from many web pages.

To assist John in technology landscape inquiries, our *TechLand* browser plugin[11] extracts the technical terms in the search query (e.g., "python", "data visualization", "tools"), and retrieves the relevant knowledge graph from the backend knowledge base. It helps address John's information needs from three perspectives (See Fig. 1). First, the plugin shows the trend of the community interest in the searched technical terms in a line chart. Second, based on the categorical knowledge of the entities in the knowledge graph, the plugin presents a set of libraries, languages and concepts related to the search terms as direct answers to the search query. These direct answers help John find available technologies related to "data visualization" without the need to read many webpages. Third, the plugin visualizes the knowledge graph in an interactive graph view. This graph view serves two purposes. First, it visualizes the correlation of relevant tools, concepts and techniques. Second, the node size indicates the community interest in the relevant technology. For example, John can observe that *JavaScript* and *D3.js* are popular language and tool for data visualization. He can also find concepts and techniques related to *D3.js*, such as *SVG*, *JSON*, *nvd3.js*, *dimple.js*. Again, John obtains these information scents without the need to read many webpages.

In addition to augmenting technology landscape queries, our *TechLand* website[12] assists developers in exploring correlated technologies via the interactive graph view of the underlying knowledge graph. For example, John can navigate from the knowledge graph of data visualization to the knowledge graph of related concepts (e.g., graph visualization) or tools (e.g., d3.js). The website also support comparing the trend of relevant technologies. For example, Fig. 2 shows the comparison of the technology trend of *JavaScript's D3.js*, *Python's matplotlib*, *R's ggplot2*.

III. MINING KNOWLEDGE GRAPH OF TECHNOLOGIES

The backend knowledge base of the *TechLand* system is a knowledge graph of technologies from Stack Overflow. In particular, we use the techniques proposed in our recent work [6], [4] to mine relational knowledge, categorical knowledge and

[11] https://github.com/tomhanlei/kg_plugin
[12] https://graphofknowledge.appspot.com

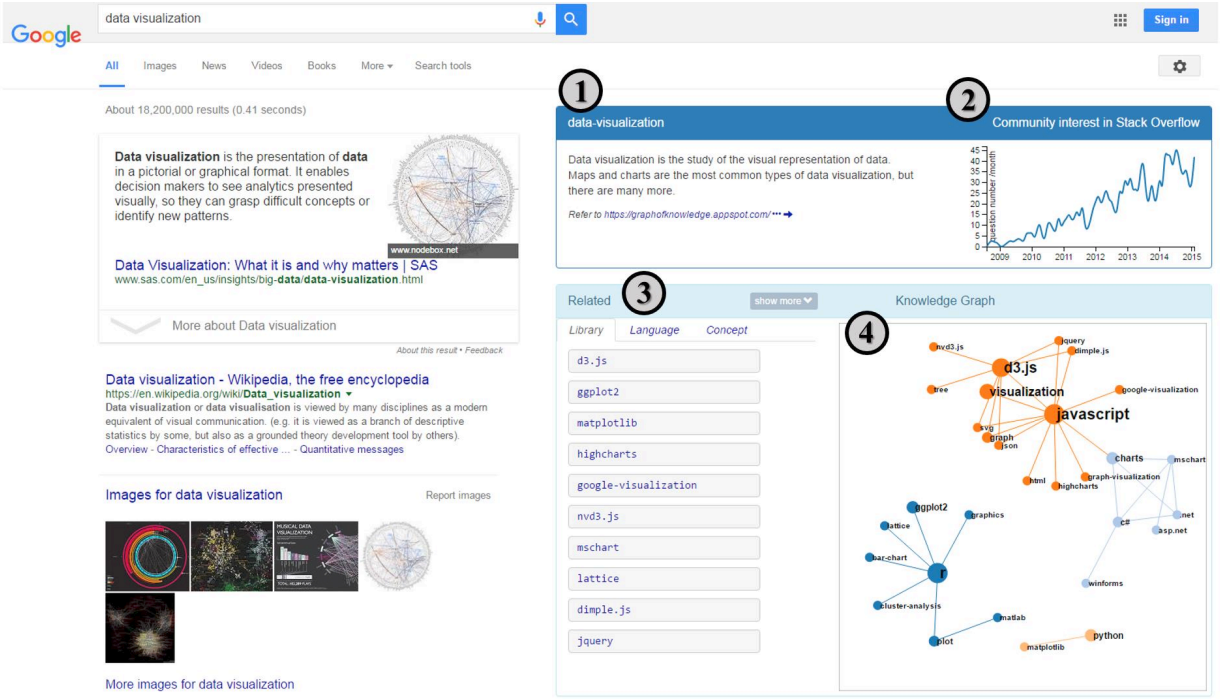

Fig. 1. Augmenting technology landscape inquiries with four pieces of information that are related to the search terms: 1) the definition of the technology given by Stack Overflow, 2) the trend of community interest in the searched technology, 3) a set of related libraries, languages and concepts, 4) an interactive knowledge graph of technology correlations.

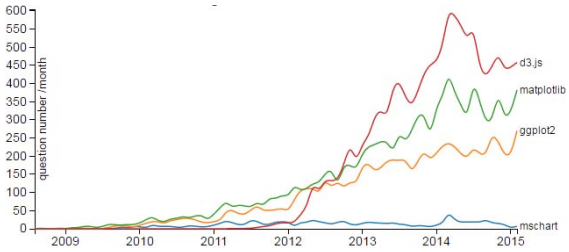

Fig. 2. The comparison of the trend of community interests

trend knowledge of technologies from Stack Overflow. In this section, we summarize the key ideas of these techniques. Interested readers are referred to our papers [6], [4] for the technical details and the evaluation of these techniques.

A. What are Technologies?

In Stack Overflow, a question must have 1-5 tags. A tag is a word (e.g., java) or a phrase (e.g., data visualization) that describes the technical term that the question revolves around [2] (see Fig. 3 for an example). Stack Overflow tags refer to a wide range of technical terms, from concepts, programming languages, platforms, libraries/tools/frameworks, to specific APIs (at class or module level). Inspired by the recent empirical studies on Stack Overflow [2], [1], [3], we consider each Stack Overflow tag as a technology. Stack Overflow enforces strict rules to create new tags[13], and the community invests huge efforts to curate the tag list (e.g., deleting unnecessary ones and merging synonyms). Thus,

[13]http://stackoverflow.com/help/privileges/create-tags

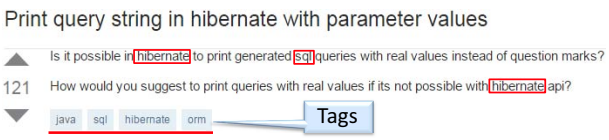

Fig. 3. Post #1710476 in Stack Overflow

Stack Overflow tags serves as a clean vocabulary of technologies for constructing the knowledge graph of technologies. For each tag, we retrieve its definition from the TagWiki as the definition of the corresponding technology (e.g., first part in Fig. 1).

B. Mining Relational Knowledge

Tags of a question are correlated. In the example in Fig. 3, *hibernate* is an *orm* framework for accessing a *sql* database from a *java* program. However, Stack Overflow manages the question tags simply as a set of terms. Therefore, the correlation of technologies is implicit and need to be discovered.

Given a technology t, to mine the correlation of its relevant technologies, we first collect all the questions that are tagged with t. Each question is considered as a transaction and the question tags as the items in the transaction. The given technology t itself is removed from the transactions. We use association rule mining [7] to discover technology correlations from tag co-occurrences in questions. As we want to mine the pair-wise correlation of technologies, we find frequent pairs of tags. Given a frequent pair of tag t_1, t_2, association rule mining generates an association rule $t_1 \Rightarrow t_2$ if the confidence of the rule is above the minimal confidence threshold.

In association rule mining, users set minimal support threshold and minimal confidence threshold to distill the most important association rules. In our application to discover technology correlations, the number of questions that are tagged with a given technology *t* vary greatly from hundreds of times to hundreds of thousands of times. That is, the number of transactions for association rule mining vary greatly from one technology to another. This makes it impractical to use a uniform minimal support threshold and minimal confidence threshold for all the technologies, because a uniform threshold will include too many associations for some technologies, but too few associations for others. Therefore, in this work, we sort the association rules by their confidence values and include association rules from the top until *N* unique tags are included. These *N* tags are considered as technologies that are related to the given technology *t*, and the included association rules capture the correlation of these *N* technologies.

C. Mining Categorical Knowledge

A technology can be of different categories. Based on our observation of technology landscape questions on Stack Overflow and Quora, we consider three categories in this work, i.e., programming language, library/tool/framework, or general concept. To determine the category of a technology, we resort to the tag definition in the TagWiki. The TagWiki of a tag is collaboratively edited by the Stack Overflow community. The TagWiki description usually starts with a short sentence to define the tag. Typically, the first noun phrase just after the *be* verb defines the category of the tag. For example, the tag *java* is defined as "Java (not to be confused with JavaScript) is a general-purpose object-oriented programming language ...". The tag *hibernate* is defined as "Hibernate is an object-relational mapping (ORM) library for the Java language ...". The tag *orm* is defined as "Object-relational mapping (ORM) is a technique for mapping object-oriented systems to relational databases". Based on these TagWiki definitions, we can categorize *java*, *hibernate* and *orm* as language, library and concept, respectively.

Based on this heuristic, we use the NLP methods (similar to the methods used in [8] for named entity recognition) to extract the noun phrase from the tag definition sentence as the category of a tag. Given the TagWiki of a tag in Stack Overflow, we extract the first sentence of the TagWiki description, and clean up the sentence by removing hyperlinks and brackets such as "{}", "()". Then, we apply Part of Speech (POS) tagging and phrase chunking to the extracted sentence. POS tagging is the process of marking up a word in a text as corresponding to a particular part of speech, such as common noun, verb, adjective. Phrase chunking [9] is the process of segmenting a sentence into its subconstituents, such as noun phrases, verb phrases. We use the Python NLTK library[14] for POS tagging and phrase chunking. Fig. 4 shows the results for the tag definition sentence of *java*. Based on the POS tagging and phrase chunking results, we extract the first noun phrase (NP) (*programming language* in this example) after the *be* verb (*is* in this example). That is, the category of *java* is *programming language*. More details can be found in [6], [10].

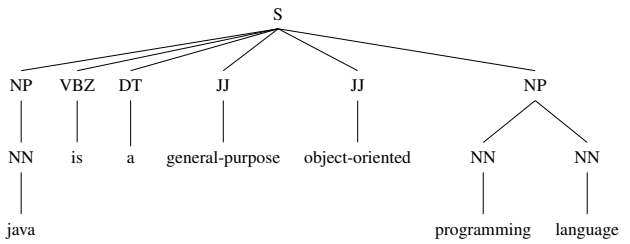

Fig. 4. POS tagging and phrase chunking results of the definition sentence of the tag *java*

After processing all the tags, we manually normalize the obtained category labels, such as normalizing uppercase and lowercase (e.g., *API* and *api*), and merging synonym terms such as *programming language and language* as *language*, *library, tool, framework, extension and API* as *library*. At the end, we obtain three categories of tags: *language*, *library*, and *concept* (i.e., all others that are not *language* or *library*)

D. Mining Trend Knowledge

Our recent work [5] studies the correlation between what developers search on Google and what developers ask on Stack Overflow. We show that tag usage statistics over time provide a good estimate of the trend of developers' interest in certain technology. In this work, we summarize the number of questions that are tagged with a given technology and that have been asked per month. This statistic over time generates a trend of the community interest in the given technology.

IV. THE TECHLAND SYSTEM

Developers' technology landscape queries usually contain some technical terms they are interested in. To help address the information needs in technology landscape queries, the *TechLand* system extracts the technical terms that are being referenced in a query and links them to the backend knowledge base, and then it displays additional facts about the technical terms, recommends related technologies based on the relationships encoded in the knowledge base, visualizes technology correlations and assists exploration of correlated technologies, and assists comparison of technology trend.

A. Extracting Technical Terms in Search Query

TechLand implements a greedy matching method to identify the technical terms in a search query. A technical term can consist of a single word (e.g., python, sockets) or a phrase (i.e., data visualization, web scraping). A query can consist of more than one technical term (e.g., python data visualization). Given a search query, *TechLand* first reduces plural words into singular words by stemming. Then, it splits the query into a list of words by space. Next, it starts with the whole query and recursively searches phrases of consecutive words with shorter length in the vocabulary of technologies of the backend knowledge base (see Section III-A). The recursion

[14]http://www.nltk.org/_modules/nltk/tag.html

stops when the phrase matches a technology or has a single word. For example, for the query *python data visualization*, *TechLand* identifies two technical terms, i.e., *python* and *data visualization*. As the search is greedy, once a phrase matches a technology (e.g., *data visualization*), *TechLand* will not further process subsequences of the phrase (e.g., *data* and *visualization*). Relying on community-curated tag synonyms list[15], *TechLand* can recognize synonyms of technical terms, such as *js* for *javascript*, *dotnet* for *.net*.

In addition to technical terms in a query, *TechLand* also attempts to identify the category labels in the query (e.g., *programming language* or *tool*) based on the vocabulary of category labels identified in Section III-C. The identification method is the same as the identification of technical terms. *TechLand* relies on the category synonyms identified in Section III-C to abstract category labels into the two general categories, i.e., *language* or *library*.

B. Displaying Additional Facts about Technical Term

For a technical term in the query, *TechLand* links it to the corresponding technology in the knowledge base and display a short definition of the technology extracted from the TagWiki. It also provides the link to navigate to the community-curated TagWiki page which contains more information about the technical term. Furthermore, *TechLand* displays a line chart of the trend of the technology used by the Stack Overflow community, which is aggregated from the tag usage statistics over time (see Section III-D)

In the current implementation, when there are two or more technical terms in the query, *TechLand* displays only the definition and the trend line chart of the technical term whose corresponding tag is least used in Stack Overflow. The underlying intuition is that the least used term is more specifically related to the developer's information needs.

C. Visualizing and Exploring Technology Correlations

For the technical terms in the query, *TechLand* constructs a knowledge graph of relevant technologies based on the relational knowledge encoded in the knowledge base (see Section III-B). The nodes of the graph represent the relevant technologies, while the edges represent the correlation between technologies. *TechLand* clusters technologies in the knowledge graph as technology clusters using community detection technique [11]. Technologies within a cluster are highly correlated, while technologies across clusters are not or loosely correlated.

TechLand visualizes the knowledge graph in an interactive graph view. Different technology clusters are visualized in different colors. Two technologies may be correlated, but as the association between them may not be strong enough, the correlation may not be included in the knowledge graph. Users can interact with the knowledge graph as follows: hover mouse over a technology to highlight its directly correlated technologies; right-click a technology to view the technology definition; double-click a technology to navigate to the corresponding webpage for that technology on the *TechLand* website. On the *TechLand* webpage, users can find detailed information about that technology and further explore the knowledge graph of that technology via the interactive graph view. This serves as an alternative to searching for more information about some correlated technology on the Web.

[15]http://stackoverflow.com/tags/synonyms

D. Recommending Related Technologies

Based on the knowledge graph for the technical terms in the query, *TechLand* categorizes the entities in the knowledge graph into three categories, i.e., *language, library, and concept*. If the search query contain the category label *language* or *library*, *TechLand* recommends only related languages or related libraries for the search query. Otherwise, *TechLand* recommends related languages, related libraries, and related concepts for the search query. Related technologies are ranked by their association strength with the searched technical terms. That is, the higher the rank, the stronger the association between a related technology and the searched technical terms. Recommendations of related technologies provide direct answers to available technologies and direct information scents for query reformulation.

E. Comparing Community Interests in Technologies

Inspired by recent empirical studies of Stack Overflow data [2], [1], [3], including our own work on technology trend [5], *TechLand* supports two ways to compare community interests in relevant technologies.

First, *TechLand* computes the co-occurrence frequency of the searched technical terms and a technology in the knowledge graph for these searched technical terms. It normalizes the co-occurrence frequency over all the technologies in the knowledge graph as a value in $(0, 1]$, and visualizes the normalized value by the node size of the technologies in the graph. For example, comparing the node size of *d3.js* in the knowledge graph in Fig. 1 and that of other tools (e.g., *matplotlib*, *ggplot2*), users can see that *d3.js* associate with the term *data visualization* more frequently in Stack Overflow questions than other tools. This is why *d3.js* is ranked at the top in the recommended libraries for *data visualization*. Although by no means conclusive, this indicates that *d3.js* seems to be a more popular tool for data visualization than other tools.

Second, *TechLand* allows users to compare the trend of several technologies in one chart. For example, Fig. 2 shows the comparison of the technology trend of *JavaScript's D3.js*, *Python's matplotlib*, *R's ggplot2*. We can see that *D3.js* was not very popular in 2012/2013 compared with the other tools, but it rises very fast since 2014.

V. IMPLEMENTATION

Our *TechLand* system includes a backend knowledge base, a browser plugin and a website.

A. TechLand Knowledge Base

The backend knowledge base is mined from the 6.5-years Stack Overflow data dump that contains 8,978,719 questions

from July 2008 to March 2015. The resulting *TechLand* knowledge base contains 39,948 technologies and 3,016,987 correlations. For each technology, we aggregate monthly tag usage statistics from July 2008 to March 2015 to generate its technology trend on Stack Overflow. From TagWiki, we collect tag definitions for 30,632 technologies, while the rest 9,316 tags either do not have TagWiki or do not have a clear definition sentence. Among these 30,632 technologies, 537 are categorized as *language*, 7,783 as *library*, and the rest as general *concept*.

B. TechLand Browser Plugin and Website

The browser plugin[16] is written in JavaScript for TamperMonkey for Chrome and GreaseMonkey for Firefox . It is currently integrated with the Google search, but can be easily extended to other search engines. The browser plugin analyzes the search query and augments the search result page with an informative summary of available technologies, technology correlations and technology trend. We use the D3 [12] library to plot the interactive knowledge graph view. The website[17] is written in Python and deployed on Google App Engine. The website presents essentially the same information for a given technology as that used to augment the search results page of the technology. In addition, the website supports the exploration of correlated technologies by navigating the underlying knowledge graph, and supports the comparison of technology trends of several technologies[18].

C. Determining Knowledge-Graph Size Empirically

Given a technology, *TechLand* constructs a knowledge graph of N relevant technologies and their strongest associations (see Section III-B). One one hand, technologies in the knowledge graph should reach a good coverage of the Stack Overflow questions that are tagged with these technologies. One the other hand, the knowledge graph should be clear for interactive use and human inspection. Both question coverage and graph clarity are affected by the number of technologies (i.e., N) included in the knowledge graph. We empirically determine an appropriate N to achieve a good trade off between the question coverage and the graph clarity.

Let S be the set of all the questions that are tagged with a given technology t, and let G be the knowledge graph of size N for the given technology t. For each technology *tag* in the knowledge graph G, we find the subset S_{tag} of questions in S that are tagged with *tag*. We union the S_{tag} for all the technologies in the knowledge graph, $\bigcup_{tag \in G} S_{tag}$. The question coverage of the knowledge graph G is computed as $|\bigcup_{tag \in G} S_{tag}|/|S|$. Finally, we average the question coverage of the knowledge graph of size N over all the technologies in the vocabulary. Figure 5 shows the average question coverage by the knowledge graph with 15 to 45 technologies. The coverage increases as the number of technologies in the knowledge graph increases. For the knowledge graph with 25 technologies

[16]https://github.com/ccywch/knowledgeGraph_plugin
[17]https://graphofknowledge.appspot.com
[18]http://graphofknowledge.appspot.com/tagcompare

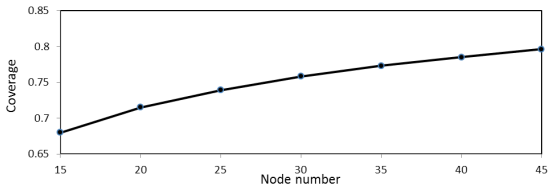

Fig. 5. The average question coverage by the knowledge graph with different number of technologies

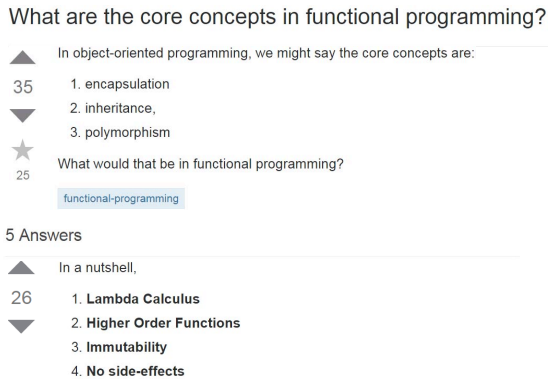

Fig. 6. An example of technology landscape question and its answer

or more, the average question coverage is above 75%, which we regard as a reasonable coverage for the knowledge graph to be representative of the crowdsourced knowledge.

Next, we randomly select 50 technologies and construct the knowledge graphs with 25-45 technologies for these selected technologies. We recruit 3 PhD students in our school to evaluate the clarity of the knowledge graphs when they are visualized in the search results page and the *TechLand* webpage. Based on the user feedback, we empirically set $N = 25$ for the knowledge graph shown in the search results page and $N = 30$ for the knowledge graph shown in the TechLand webpage. We let the knowledge graph in the *TechLand* webpage contain more technologies, because the *TechLand* webpage has larger space for visualizing the knowledge graph, compared with the search results page.

VI. Evaluation of TechLand Recommendation

In this section, we evaluate the usefulness of the *TechLand*'s recommendations of related technologies for technology landscape inquiries by using them to answer real technology landscape questions from Stack Overflow.

A. Experiment design

We observe two types of technology landscape questions on Stack Overflow. First, some novice developers may ask concept-related questions, i.e., some core concepts that they need to know in order to learn certain technology (see Fig. 6 for an example). Second, developers often ask for library recommendations for certain tasks, such as "*What is a good RDF library for .net?*" and "*What Java XML library do you recommend (to replace dom4j)?*".

Based on our observations, we define several heuristics rules (e.g., question title contains "what ... concepts" or "what ...

libraries") to collect a set of candidate technology landscape questions. We then manually filter inappropriate questions, and finally randomly select 50 concept-related and 50 library-related technology landscape questions with more than one answers[19]. For the technology landscape questions, answers often list related concepts or libraries (see Fig 6). Or answers often provide a hyperlink to the related concepts or libraries so that readers can access the relevant resources for more details. Based on such heuristics, we read the answers of the selected questions to compile a set of related concepts and libraries as the ground truth for the questions. As tags usually consist of 4 or less words, we consider only phrases with 4 or less words as the ground truth in this study. Although concept-related questions ask for concepts, question answers sometimes include libraries which are also very relevant to the question. Similarly, for library-related questions, question answers sometimes include relevant concepts. As it is often difficult to separate concepts from libraries, we do not distinguish the two categories of information in the ground truth.

For each selected question, we extract technical terms in question title in the same way as *TechLand* extracts technical terms from search query. For example, for the question "*what are the core concepts in functional programming?*", we extract *functional programming*. For the question "*what is a good RDF library for .net?*", we extract *.net* and *RDF*. We then construct the knowledge graphs with 30 technologies for the extracted technical terms, and use the technologies in the knowledge graphs as potential answers to the given technology landscape question. We examine how many related concepts and libraries in the ground truth are covered by the technologies in the knowledge graph. In our analysis, we consider technology synonyms as a match, such as *oop and object oriented programming*, *javascript and java-script*. We average the coverage over the 50 concept-related questions and the 50 library-related questions.

B. Results

For concept-related questions, on average, 48.3% of related concepts and libraries in the ground truth are covered by the knowledge graphs. For library-related questions, the average coverage is much higher and 63.5% of related concepts and libraries in the ground truth are covered by the knowledge graphs. The coverage for concept-related questions is lower because many concepts mentioned in the answers are not used as tags. Thus, they will not appear in the knowledge graph mined from tags. In contrast, libraries are commonly used as tags, and thus are more likely covered by the knowledge graph.

We further investigate why our knowledge graph fails to cover some related concepts and libraries mentioned in the answers. First, some concepts refer to good programming practices (e.g., "program mindfully") which is never used as tags. Second, some answers are very old and related concepts or libraries in the answers are no longer widely used or change to new names, such as *symbian* and *nokia qt*. Third, some answers contain specific version of some libraries or concepts such as *python 3.4, nltk 3.0*, while the knowledge graph contains the general ones (e.g., *python-3.x, nltk*). Forth, tags of some concepts or libraries are not used frequent enough to be included in knowledge graph, such as *wt* and *libwww*.

[19]The question list can be found at https://graphofknowledge.appspot.com/question

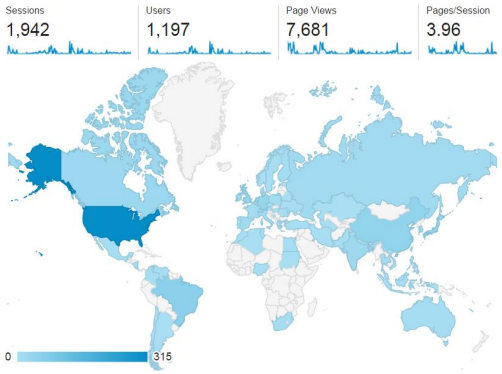

Fig. 7. Visit statistics of the *TechLand* website by Google Analytics

VII. User Evaluation

We evaluate the *TechLand* system in two complementary ways. First, we want to deploy it to real users to understand if they would actually use it and, if so, how they would use it. We post announcements on programming forums (e.g., StackApp), and collect the usage statistics of the *TechLand* website by Google Analytics. Second, we want to directly compare how searching for technology landscape information with and without the help of the *TechLand* system differs. To do this, we conduct a lab study for which we recruit 12 participants. Each of them performs 6 technology landscape inquiry tasks inspired from our observation of Stack Overflow technology landscape questions and from usage patterns of our website in field deployment. In the below, we present the details of each of the evaluations and discuss our findings.

A. Field Deployment

We release our website to the public and post this news on several programming-related forums (e.g., http://stackapps.com/questions/6569). As shown in Figure 7, according to the Google Analytics of the website traffic data[20], 1,197 users from 79 countries visited our site from Sept 4, 2015 to March 29, 2016. These users on average browse 3.96 technology pages in each session for 6 minutes and they browse 7,681 technology pages in total (including the homepage). These usage statistics provide some initial evidence of the developers' interests in technology landscape services.

To investigate the user navigation pattern on our website, we analyze the web logs in detail. Approximately 700 users just came to have a look at our homepage or visited just one or two technology pages, and subsequently did nothing. We discard these users from our analysis, obtaining 290 users who at least visited three technology pages in one session. Among

[20]As most search engine robots do not activate Javascript, robot traffic is not counted in Google Analytics [13]

these 290 users, about 50 users returned days or weeks later to use our website again.

We observe some interesting technology exploration history in the web logs of these 290 users. For example, the user 162 first visited the *nlp* page and double-clicked the *machine learning* node in the *nlp*'s knowledge graph. This leads the user to the *machine-learning* page. As *machine learning* is frequently tagged together with *nlp* in Stack Overflow questions, the *machine learning* node is one of the biggest nodes in the *nlp*'s knowledge graph. Then, the user further double-clicked the *neural network* node in the *machine learning*'s knowledge graph to navigate to the *neural networdk* page. From the *neural networdk* page, the user finally navigated to the *theano* page. Theano is an efficient numerical computation library for Python. Such technology exploration history indicates that a graphical view of correlated technologies could provide guided navigation for the users to explore the technology landscape and find the desired information.

When designing the *TechLand* website, we expect that users would first search and view some technology pages and then compare the relevant technologies they are interested in. Indeed, 72 of 290 users (24.8%) used our website in this manner. For example, the user 164 first visited the *chef*, *ansible* and *puppet* pages (several configuration management tools) and compared the technology trend of these tools[21].

On the whole, the field deployment did not lead to as much usage as we had hope. However, the usage statistics of our *TechLand* website, albeit very limited, are promising, given that we post only brief announcements, perform no training, and many users likely just visit the website to satisfy their curiosity. This initial results demonstrate both the needs and the interests in technology landscape services that our *TechLand* system supports. We are now investigating search engine optimization to promote the rank of our technology pages in the search results for certain technology landscape queries (e.g., *data visualization tools*). If ranked high in the search results, our technology pages could complement existing online documents with an objective and informative summary of technology landscape.

B. Lab Study

In the lab study, we compare Google search with the TechLand system support (i.e., experimental system) against using Google search only (i.e., baseline system). The participants are provided with technology landscape inquiry tasks describing the information needs and are asked to use the experimental or baseline system to acquire information.

1) Tasks: We construct 6 independent tasks based on our observation of Stack Overflow questions and usage patterns of our *TechLand* website in field deployment. 6 tasks cover three categories of technology landscape questions: practice-related (Task1 and Task2), concept-related (Task3 and Task4), and library-related (Task5 and Task6). Each task has three questions. The tasks and their questions are shown in Table I.

[21]http://graphofknowledge.appspot.com/tagcompare/chef&ansible&puppet

Task 1: bdd (behavior driven development) (1) List 5 related concepts; (2) Find 5 PLs and corresponding frameworks/libraries to support BDD practice; (3) From the languages and tools in question (2), please determine BDD is widely adopted for applications developed in which programming languages?
Task 2: data visualization (1) List 5 related concepts; (2) Determine 4 PLs and corresponding tools/libraries for data visualization; (3) Assume that you know all these languages, which language and tool in Question (2) will you use based on the technology trend?
Task 3: encryption (1) List 5 related concepts; (2) Find 5 PLs and corresponding frameworks/libraries to implement encryption methods; (3) Assume that you know both java and c#, is there any tool that can be used for both languages?
Task 4: statistics (1) List 5 related concepts; (2) Select 2-3 PLs and corresponding tools/libraries that support statistical analysis; (3)Assume that you know all these languages, which language and tool in Question (2) will you use based on the technology trend? Please explain your reasons such as documentation, community adoption.
Task 5: beautifulsoup (1) List 5 related concepts; (2) List 2-3 tasks that beautifulsoup can support; (3) For tasks that you find in question (2), are there any alternative tools to beautifulsoup? (not limited to the same programming language)
Task 6: numpy (1) List 5 related concepts; (2) List 3 libraries that are highly related with numPy; (3) What tasks can these libraries in question (2) implement? (please list them one by one)

TABLE I
TASK DESCRIPTIONS (PL DENOTES PROGRAMMING LANGUAGE)

The first two questions of all the tasks are general, which ask the participants to find related concepts, languages, and libraries for a given practice, concept, or library. The third question of the tasks are specific to the given practice, concept, or library, which requires deeper understanding of available technologies, technology correlations, and technology trends.

2) Participants: We recruit 12 participants by promoting the experiments via emails and word-of-mouth. The recruited participants are all PhD students in our school majoring in computer science and computer engineering. The participants have diverse research background and they use different software tools and programming languages in their work. All the participants use Google regularly. However, none of the participants claim to be familiar with the technologies in the experimental tasks. The reason for selecting this set of participants is that in this study we would like to focus on the ability of the *TechLand* system to support users who are unfamiliar with a technology. The participants receive $10 shopping coupon as a compensation of their time.

3) Experiment Procedures: The experiment begins with an introduction to the study. Then, we explain and walk through all of the features of the *TechLand* system that are discussed in Section IV. Next, participants perform a training task with the *TechLand* system to familiarize the features. After the training session, each participant is asked to work on the six tasks. All of the six tasks are completed with no interventions by the experimenters. For each category of the tasks, participants use the experimental system for one task and the baseline system

Measures	Google	Google + TechLand
Average confidence	3.56 (0.93)	4.0 (0.75)*
Average completeness	0.922 (0.113)	0.988 (0.056)**
Average validity	0.860 (0.139)	0.958 (0.074)**

TABLE II
THE MEAN (STANDARD DEVIATION) OF THE THREE MEASURES, * DENOTES $\rho < 0.1$ AND ** DENOTES $\rho < 0.05$

for the other. The order of task category, the order of tasks for each category, and the order of using the experimental or baseline system are rotated based on the Latin Square [14], which help reduce the learning and fatigue effects.

Participants are given the description of the tasks and up to 10 minutes to complete each task. After completing the task, participants are asked to rate their confidence in the information they collect (on 5-point likert scale with 1 being least confident and 5 being most confident). If participants use the experimental system for a task, they are also asked to rate the helpfulness of the *TechLand* system for the task (5-point likert scale with 1 being least helpful and 5 being most helpful). At the end, participants fill in the System Usability Scale (SUS) questionnaire [15]. The questionnaire also asks participants to select the *TechLand* system features that they deem most useful or least useful for the tasks. In addition, we conduct a semi-structured interview focusing on their perceptions and their use of features during the study. Participants record the screen while they work on the tasks. We analyze the task videos to compare the search behaviors with and without the *TechLand* system.

4) Overall Performance: We use participants' confidence in the task results, the completeness of the results, and the validity of the results as metrics to evaluate the participants' performance. We use the Wilcoxon Signed-Rank test to check the significance of the differences in participants' confidence, result completeness and result validity between the experimental system and the baseline system.

For the first and second question of each task, we compute the result completeness score as the number of entries that participants collect for the question divided by the number of entries requested by the question. For example, if the question asks for 5 related concepts and the participant finds 3 concepts, the result completeness for this question is 0.6. For the third question of each task, if the participant answers the question, we set the result completeness score as 1, otherwise 0. We then obtain the results completeness score for the task by averaging the result-completeness scores of the three questions. We find a domain expert in our school for each task to examine the validity of the results submitted by the participants. The rating procedure is similar to result completeness, but the expert needs to examine whether the collected information is correct and assign a score in $[0-1]$ by 0.2 increment. That is, the rating of result validity is 5-point scale.

Table II shows the average user confidence in their results, and the average result completeness, and the average result validity between using the experimental system (Google+TechLand) and using the baseline system (Google). For participants' confidence in the results, participants are more confidence in their results when using

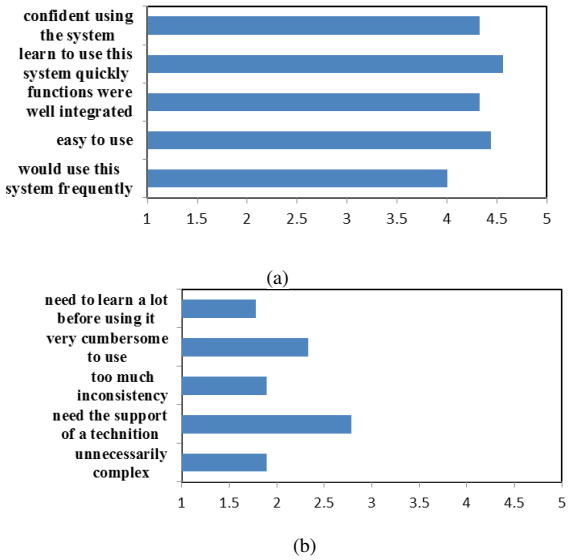

Fig. 8. Average score of SUS results

Google+TechLand, compared with using Google only. From the post-study interviews, participants indicate that this is mainly because they feel that the *TechLand* information is backed by the crowdsourced knowledge from Stack Overflow. However, the difference between the participants' confidence is not significant. For result completeness and result validity, there are significant difference in the two measures between using Gogole+TechLand and using Google. On average, the result completeness and result validity for the tasks using Google+TechLend increase 7.2% and 11.4% respectively, compared with those tasks using Google only.

5) System Usability: Figure. 8 summarizes the users' ratings of the 10 system design and usability questions in the System Usability Scale questionnaire. Figure 8(a) shows that users agree or strongly agree that our system is easy to use and the features of the Techland system are well integrated. Figure. 8(b) further confirms the simplicity and consistency of our Techland system. Furthermore, the average helpfulness of the *TechLand* system for the tasks is 4.09, which indicates that participants appreciate the help of the *TechLand* system in the tasks.

6) Search Behavior: As *TechLand* provides direct information for the tasks, especially the first two questions, we expect that participants would issues less queries and visit less webpages with the help of *TechLand*. However, we do not observe this difference by analyzing screen-recorded task videos. In fact, participants sometimes issues more queries or visit more webpages when they have the *TechLand* recommendations for the tasks. Interviews with participants suggest two reasons. First, participants are new to the *TechLand* system and do not fully trust the quality of its recommendations. Therefore, they search and read relevant information about the recommended information to verify the recommendations. This also shows that participants are rigorous in completing the tasks. Second, with the *TechLand* information, participants do not need to spend much time in searching and collecting

the initial information scents in the beginning of the tasks. Instead, they can directly search technologies recommended by the *TechLand*, which could also lead to more searches and more webpage visits.

7) User Feedback: In the post-study interviews, participants suggest that augmenting search results page with the *TechLand* information is more effective in assisting technology landscape queries than accessing the information from the *TechLand* website. Although the browser plugin and the website present essentially the same information, augmenting search results page provides better integration of the *TechLand* information in the search process.

In terms of useful features, participants rate recommendations of related technologies most useful. They suggest that the top recommendations are usually very relevant to their information needs, and thus provide useful information scents for them to decide what to search or which webpage to read.

Participants suggest that the technology trend of a technology bears little meaning, unless it is compared with the trend of similar technologies, which is needed to determine trendy technologies. However, participants feel inconvenient to switch to the *TechLand* website to compare similar technologies. They suggest that the system could automatically generate trend comparison chart for the recommended technologies.

Participants suggest that the graph view of the knowledge graph is moderately useful for understanding technology correlations. A major limitation of the current knowledge graph is that it captures only general correlation between technologies. The knowledge graph could become more useful if it can explain technology relationships with more specific semantics, for example, the library *nvd3.js* extend the library *d3.js*, or the library *nltk* is analogical to the library *opennlp*.

8) Limitations: This lab study has several limitations. The participants are PhD students. First-use studies make it difficult to understand how a user might use the system longitudinally. Use feedbacks may be biased by the experimental tasks. The experimental results may not be generalizable to other populations or longitudinal use in the field. We release the *TechLand* system to the public to collect real-world usage data to answer these questions.

VIII. Related Work

Stack Overflow has been the focus of many studies[22], including discover topics and trends of developers' discussions [1], [3], predict answer quality [16] or user participation [17], identify experts [18], analyze social interactions inside the cooperative community of Stack Overflow [19], [20], and study technology trends [1], [16]. These studies show that Stack Overflow is a high-quality knowledge repository of developers' thoughts, needs and practices. Nasehi et al. shows that the main technologies that the question revolves around can usually be identified from question tags [2]. These empirical studies inspire our *TechLand* system which exploits the crowdsourced knowledge from Stack Overflow to address the information needs in technology landscape inquiries.

Many studies have shown that structured knowledge can emerge from social tagging systems [21], [22], [23]. Hierarchical clustering techniques have been applied to induce taxonomies from collaborative tagging systems [24], [25], [26], and from software project hosting site Freecode [27]. Schmitz analyzes association rule mining results to infer a subsumption based model from Flickr tags [28]. In this work, we also use association rule mining to discover relational knowledge between technologies from tag co-occurrence patterns. In addition, we incorporate relational knowledge as well as categorical and trend knowledge from Stack Overflow into a knowledge graph, which can be exploited to provide not only additional facts about the technology in a query, but also extended suggestions for users.

One main driver for enhancing search experience has been the ability to incorporate structured data in search results page [29], [30] for named entity queries [31], [32]. In the realm of web search, the information is organized into a knowledge graph of real-world objects or concepts commonly referred to as entities. Mining knowledge graphs from structured (e.g., Freebase, Linking Open Data, DBpedia) and unstructured (e.g., wikipedia, social media) data is a vibrant area in database and data mining research [33], [34], [35]. Knowledge-graph based applications, such as query understanding and reformulation [36], entity profiling [37], exploratory search [38], serendipitous search [39] have also been actively researched. Our knowledge graph is a domain-specific knowledge graph for software engineering technologies mined from unstructured Stack Overflow data. Based on this knowledge graph, we build the *TechLand* system to augment search results page and assist technology landscape inquiries.

IX. Conclusion and Future Work

In this paper, we present the *TechLand* system that aggregates and visualizes the relational, categorical and trend knowledge from Stack Overflow. *TechLand* helps address the information needs in technology landscape inquiries by providing an objective and informative summary of available technologies, technology correlations and technology trend. Our evaluation shows that 1) *TechLand* can assist in answering technology landscape questions on Stack Overflow; 2) *TechLand* can guide developers' exploration of large information space of correlated technologies; 3) *TechLand* can help address the cold start problem in technology landscape inquiries for which developers who are unfamiliar with a technology do not know what to search for. The user study shows that the *TechLand* information can increase participant's confidence in the information collected for technology landscape questions, and improve the completeness and validity of the information collected. Our work demonstrates a direction for enhancing developers' lives on the Web by incorporating structured data in search results. We envision more research on mining knowledge graph from software engineering social data and more entity-centric search applications for software engineering.

[22] A community-curated list of publications using Stack Overflow data can be found at http://meta.stackexchange.com/q/134495.

ACKNOWLEDGMENT

The authors would like to thank the experiment participants for their patience. This work was partially supported by MOE AcRF Tier1 Grant M4011165.020.

REFERENCES

[1] A. Barua, S. W. Thomas, and A. E. Hassan, "What are developers talking about? an analysis of topics and trends in stack overflow," *Empirical Software Engineering*, vol. 19, no. 3, pp. 619–654, 2014.

[2] S. M. Nasehi, J. Sillito, F. Maurer, and C. Burns, "What makes a good code example?: A study of programming q&a in stackoverflow," in *Software Maintenance (ICSM), 2012 28th IEEE International Conference on*. IEEE, 2012, pp. 25–34.

[3] C. Rosen and E. Shihab, "What are mobile developers asking about? a large scale study using stack overflow," *Empirical Software Engineering*, pp. 1–32, 2015.

[4] C. Chen and Z. Xing, "Mining technology landscape from stack overflow," in *10th International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE/ACM, 2016.

[5] C. Chen and Z. Xin, "Towards correlating search on google and asking on stack overflow," in *40th Annual Computer Software and Applications Conference (COMPSAC)*. IEEE, 2016, pp. 83–92.

[6] C. Chen, S. Gao, and Z. Xing, "Mining analogical libraries in q&a discussions–incorporating relational and categorical knowledge into word embedding," in *23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, vol. 1. IEEE, 2016, pp. 338–348.

[7] R. Agrawal, R. Srikant *et al.*, "Fast algorithms for mining association rules," in *Proc. 20th int. conf. very large data bases, VLDB*, vol. 1215, 1994, pp. 487–499.

[8] J. Kazama and K. Torisawa, "Exploiting wikipedia as external knowledge for named entity recognition," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 698–707.

[9] S. Bird, "Nltk: the natural language toolkit," in *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics, 2006, pp. 69–72.

[10] C. Chen and Z. Xing, "Similartech: Automatically recommend analogical libraries across different programming languages," in *31st IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE/ACM, 2016.

[11] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.

[12] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 data-driven documents," *IEEE transactions on visualization and computer graphics*, vol. 17, no. 12, pp. 2301–2309, 2011.

[13] "Traffic from search engine robots," https://support.google.com/analytics/answer/1315708?hl=en.

[14] B. J. Winer, D. R. Brown, and K. M. Michels, *Statistical principles in experimental design*. McGraw-Hill New York, 1971, vol. 2.

[15] J. Brooke, "Sus-a quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.

[16] 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 *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 850–858.

[17] P. Fugelstad, P. Dwyer, J. Filson Moses, J. Kim, C. A. Mannino, L. Terveen, and M. Snyder, "What makes users rate (share, tag, edit...)?: predicting patterns of participation in online communities," in *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*. ACM, 2012, pp. 969–978.

[18] A. Pal, S. Chang, and J. A. Konstan, "Evolution of experts in question answering communities." in *ICWSM*, 2012.

[19] C. Treude, O. Barzilay, and M.-A. Storey, "How do programmers ask and answer questions on the web?: Nier track," in *Software Engineering (ICSE), 2011 33rd International Conference on*. IEEE, 2011, pp. 804–807.

[20] L. Guerrouj, S. Azad, and P. C. Rigby, "The influence of app churn on app success and stackoverflow discussions," in *Software Analysis, Evolution and Reengineering (SANER), 2015 IEEE 22nd International Conference on*. IEEE, 2015, pp. 321–330.

[21] H. Halpin, V. Robu, and H. Shepherd, "The complex dynamics of collaborative tagging," in *WWW*. ACM, 2007, pp. 211–220.

[22] G. Macgregor and E. McCulloch, "Collaborative tagging as a knowledge organisation and resource discovery tool," *Library review*, vol. 55, no. 5, pp. 291–300, 2006.

[23] V. Robu, H. Halpin, and H. Shepherd, "Emergence of consensus and shared vocabularies in collaborative tagging systems," *ACM Transactions on the Web (TWEB)*, vol. 3, no. 4, p. 14, 2009.

[24] J. Gemmell, A. Shepitsen, B. Mobasher, and R. Burke, "Personalizing navigation in folksonomies using hierarchical tag clustering," in *Data Warehousing and Knowledge Discovery*. Springer, 2008, pp. 196–205.

[25] D. Helic, M. Strohmaier, C. Trattner, M. Muhr, and K. Lerman, "Pragmatic evaluation of folksonomies," in *WWW*. ACM, 2011, pp. 417–426.

[26] E. Simpson, "Clustering tags in enterprise and web folksonomies." in *ICWSM*, 2008.

[27] S. Wang, D. Lo, and L. Jiang, "Inferring semantically related software terms and their taxonomy by leveraging collaborative tagging," in *ICSM*. IEEE, 2012, pp. 604–607.

[28] P. Schmitz, "Inducing ontology from flickr tags," in *Collaborative Web Tagging Workshop at WWW2006*, vol. 50, 2006.

[29] R. Blanco, P. Mika, and S. Vigna, "Effective and efficient entity search in rdf data," in *The Semantic Web–ISWC 2011*. Springer, 2011, pp. 83–97.

[30] K. Haas, P. Mika, P. Tarjan, and R. Blanco, "Enhanced results for web search," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 725–734.

[31] E. Meij, K. Balog, and D. Odijk, "Entity linking and retrieval for semantic search." in *WSDM*, 2014, pp. 683–684.

[32] X. Yin and S. Shah, "Building taxonomy of web search intents for name entity queries," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 1001–1010.

[33] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning, "Multi-instance multi-label learning for relation extraction," in *Proceedings of the 2012 Joint Conference on EMNLP-CoNLL*. Association for Computational Linguistics, 2012, pp. 455–465.

[34] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland, "Textrunner: open information extraction on the web," in *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the ACL: Demonstrations*. Association for Computational Linguistics, 2007, pp. 25–26.

[35] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J.-R. Wen, "Statsnowball: a statistical approach to extracting entity relationships," in *WWW*. ACM, 2009, pp. 101–110.

[36] N. Sarkas, S. Paparizos, and P. Tsaparas, "Structured annotations of web queries," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 771–782.

[37] S. R. Yerva, Z. Miklos, F. Grosan, A. Tandrau, and K. Aberer, "Tweetspector: Entity-based retrieval of tweets," in *SIGIR*. ACM, 2012, pp. 1016–1016.

[38] Y. Genc, "Exploratory search with semantic transformations using collaborative knowledge bases," in *WSDM*. ACM, 2014, pp. 661–666.

[39] I. Bordino, Y. Mejova, and M. Lalmas, "Penguins in sweaters, or serendipitous entity search on user-generated content," in *Proceedings of the 22nd ACM international conference on information and knowledge management*. ACM, 2013, pp. 109–118.