Mining Technology Landscape from Stack Overflow

Chunyang Chen
School of Computer Science and Engineering
Nanyang Technological University, Singapore
chen0966@e.ntu.edu.sg

Zhenchang Xing
School of Computer Science and Engineering
Nanyang Technological University, Singapore
zcxing@ntu.edu.sg

ABSTRACT

The sheer number of available technologies and the complex relationships among them make it challenging to choose the right technologies for software projects. Developers often turn to online resources (e.g., expert articles and community answers) to get a good understanding of the technology landscape. Such online resources are primarily opinion-based and are often out of date. Furthermore, information is often scattered in many online resources, which has to be aggregated to have a big picture of the technology landscape. In this paper, we exploit the fact that Stack Overflow users tag their questions with the main technologies that the questions revolve around, and develop association rule mining and community detection techniques to mine technology landscape from Stack Overflow question tags. The mined technology landscape is represented in a graphical Technology Associative Network (TAN). Our empirical study shows that the mined TAN captures a wide range of technologies, the complex relationships among the technologies, and the trend of the technologies in the developers' discussions on Stack Overflow. We develop a website (https://graphofknowledge.appspot.com/) for the community to access and evaluate the mined technology landscape. The website visit statistics by Google Analytics shows the developers' general interests in our technology landscape service. We also report a small-scale user study to evaluate the potential usefulness of our tool.

Categories and Subject Descriptors

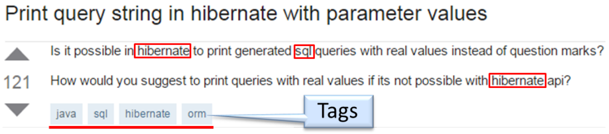
I.2.4 [**ARTIFICIAL INTELLIGENCE**]: Knowledge Representation Formalisms and Methods

Keywords

Technology Associative Network, Association Rule Mining, Community Detection, Technology Landscape

1. INTRODUCTION

A diverse set of technologies are available for use by developers and such set continues to grow. In this paper, we use the term "technology" to broadly refer to processes, methods, tools, platforms, languages, and libraries in the context of software engineering. To make the right choice for a technology in a software project, developers need to have a good understanding of the technology landscape, i.e., available technologies, the relationships among them, and the trends of them. To that end, developers often turn to two information sources on the Web [2]. First, domain experts often write articles about technology landscape, such as *"best machine learning resources for getting started"*, *"Python's SQLAlchemy vs other ORMs"*, *"20 best JavaScript charting libraries"*, Second, developers can seek answers from community-curated list of useful technologies (e.g., *"awesome PHP"*) or from Q&A websites such as Stack Overflow or Quora (e.g., *"which framework is best for web development in PHP?"*). These expert articles and community answers are indexable by search engines, thus enabling developers to find answers to their technology landscape inquiries.



Print query string in hibernate with parameter values

121

Is it possible in hibernate to print generated sql queries with real values instead of question marks?

How would you suggest to print queries with real values if its not possible with hibernate api?

Figure 1: Post #1710476 in Stack Overflow

However, there are three limitations with these expert articles and community answers. First, the technology landscape is in a constant state of change. Thus, expert articles and community answers are easily out of date. For example, the article *evaluation of .net mocking libraries* compares "how the playing field looks today" (as of December 14, 2013) and "how the playing field looked two years ago". It is in the top 10 list that Google returns for "best .net mocking framework", but it cannot reflect the state-of-the-practice "today" (as of January 2016). Second, expert articles and community answers usually focus on a specific technology, while not a set of correlated technologies. For example, reading the article *"best PHP framework for 2015"*, one cannot know that *Symfony* uses a separate ORM library *Doctrine*, while *Laravel* includes a built-in ORM library *Eloquent*. Developer needs to read another article like *"best available PHP ORM libraries"* to aggregate the information. Such information aggregation is opportunistic. Third, expert articles and community answers are often primarily opinion-based. This is why Stack Overflow usually closes technology-landscape-style questions (e.g., *"C# - Which Unit Testing Framework"*), because such questions will likely solicit debate and arguments.

Several empirical studies [3, 20, 18] show that taken in the aggregate, Stack Overflow question tags provide a good estimation of technology landscape over time. Figure 1 shows an example. We can see that tags identify the main technologies that the question revolves around, even those that are not explicit in the question content (in this example *java* and *orm* (object-relational-mapping)). Technologies that tags represent are correlated. In this example, *hibernate* is an *orm* framework for accessing a *sql* database from a *java* program. However, as Stack Overflow manages the question tags as a set of words, the relationships among tags and the tag usage over time are implicit in the system.

In this paper, we propose to apply association rule mining [1] and community detection [5] techniques to mine the technology landscape from Stack Overflow question tags. The mined technology landscape is represented as a graphical Technology Associative Network (TAN). For each tag in the TAN, we use the Natural Language Processing (NLP) method [6] to analyze the tag description to determine if the tag represents a software library, programming language, or general concept. We also summarize the question asking activities of the tag over time.

We apply our approach to Stack Overflow data dump[1] and evaluate the mined technology landscape from the perspectives of tag and question coverage, semantic distance of technology associations, network structure, and network evolution. Our evaluation shows that the mined technology landscape captures a wide range of technologies, the complex relationships among technologies, and the trends of technologies. We release the mined technology landscape in our website. The website supports some basic search and exploration features. The Google Analytics results of the website usage data for about 4 months provides initial evidence of the public interests in the technology landscape[2]. A small-scale user study is conducted, which demonstrates the potentials of the mined technology landscape in assisting technology search and exploration.

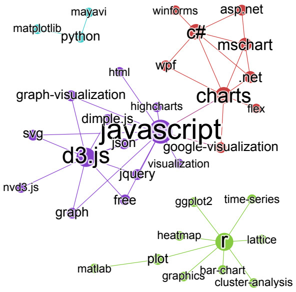
We make the following contributions in this paper:

- a systematic approach for mining and analyzing technology landscape from Stack Overflow;
- a foundational study of semantic, structural and dynamic properties of the mined technology landscape;
- a web site https://graphofknowledge.appspot.com/ for the public access of our technology landscape service;
- a user study for evaluating the usefulness of the mined technology landscape.

2. MOTIVATING EXAMPLE

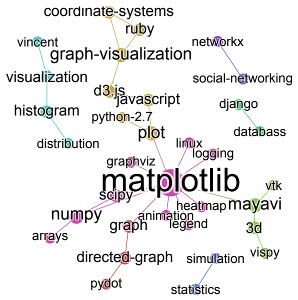
This section illustrates the kinds of problems one (say the developer John) encounters when searching unfamiliar technologies on the Web, and how an overview of technology landscape could help. Our example mimics what happens when one is new to a technology, such as machine learning, image processing, data visualization, and would like to find the available software libraries and related concepts for the technology. We use "data visualization" as an example here.

A reasonable starting query is *data visualization tools* (or *data visualization software*). The top 10 results by Google for the query *data visualization tools* include 8 reviews of data visualization tools, the Wikipedia page about data visualization, and a specific data visualization software (Tableau Software). The reviews of data visualization tools list 8 to 37 of software tools. Reading all of them and comparing the lists is a time-consuming task. After some reading, John summarizes a few good candidates, such as *d3.js, matplotlib, dygraphs, chart.js*, that are commonly mentioned in different reviews. However, there is a concern of out-of-date information in these reviews as they were posted in early 2015 or in 2014. Reading the Wikipedia page, John learns some related concepts (e.g., *information graphics*, *scientific visualization*) and several types of diagrams (e.g., *bar chart*, *scatter plot*). He then refines the query like *scientific visualization tools.* As John is interested in the tools for Python, he also tries the queries like *python data visualization tools.* The experience with the search results is more or less the same. Certainly it is possible to issue queries that lead to the desired result quickly. However, in many cases one has to browse, read, compare, and aggregate information from many web pages when he wants to explore and understand an unfamiliar technology landscape.



Library	Concept
d3.js	svg
ggplot2	json
matplotlib	graph
highcharts	visualization
nvd3.js	time-series
dimple.js	heatmap
flex	plot
mschart	graphics
lattice	bar-chart
mayavi	free

Figure 2: The landscape of "data-visualization"



Library	Concept
matplotlib	histogram
d3.js	distribution
pydot	3d
graphviz	statistics
vtk	simulation
mayavi	arrarys
vispy	heatmap
numpy	animation
networkx	legend
mayavi	directed-graph
scipy	visualization
vincent	logging

Figure 3: The landscape of "python data-visualization"

Assume we can aggregate important software tools and concepts related to data visualization in an overview of technology landscape like the Figure 2. In addition to human inspection of the graphical Technology Associative Network (TAN), analyzing the description of the technologies in this TAN using NLP techniques can identify a list of software libraries for different programming languages, such as Javascript's *d3.js*, *nvd3.js*, *highcharts*, *dimple.js*, C#'s *mschart*, R's *ggplot2* and *lattice*, and Python's *matplotlib*, as shown to the right of the TAN. Assume the node size in the TAN is proportional to the number of questions tagged with the corresponding technology in a Q&A website (i.e., the larger the node, the more questions asked for the technology). This statistics indicates that Javascript's *d3.js* seems to be a hot software tool. Note that this statistics is derived from the community activities, not based on personal opinion. Furthermore, John may also observe some correlated software tools to *d3.js* in the TAN, such as *nvd3.js* and *dimple.js*, the two libraries extends *d3.js.* Apart from that, John may be specifically interested in data visualization in python and the corresponding TAN (Figure 3) can show him more detailed technologies about data visualization in python including libraries such as *vispy*, *mayavi*, *vtk*, *pydot.*

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

[2] The website serves mainly as a web portal (prototype) to demonstrate our empirical results.

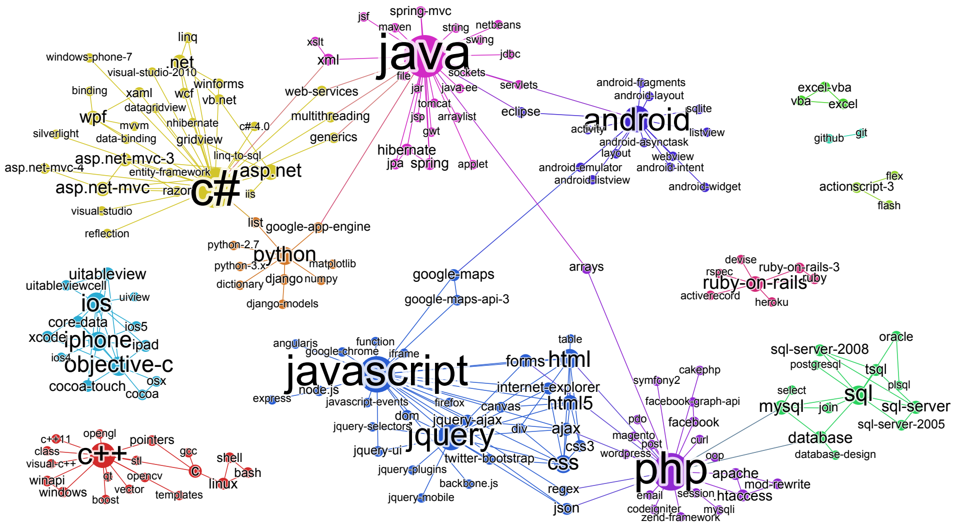


Figure 4: The general TAN mined at minimal support 0.0007 and minimal confidence 0.15

Other than software libraries, John may find other related information in the TANs, such as related concepts (e.g., *graph-visualization*, *cluster-analysis*), different types of charts (e.g., *bar-chart*, *tree*), specific data formats (e.g., *svg*, *json*), and layout algorithms (e.g., *force-layout*). Such information scents can help John who does not know the right technical terms formulate the "right" queries to obtain the results he needs. Collecting such information scents from the Google search results often requires browsing and reading many web pages.

Overall, the overview of the technology landscape helps John find answers to the three questions regarding *data visualization*: what are available software libraries? what are their trends? what are related concepts? In addition, during his exploration of the TAN, John may also find technologies he is not aware of. Such serendipitous discoveries would help extend his knowledge.

3. THE APPROACH

In this work, we focus on how we can obtain a technology landscape like those shown in Figure 2. Manually creating a technology landscape of tens of thousands of technologies obviously would require significant time and human efforts. In this section, we introduce our approach to automatically mine technology landscape from Stack Overflow question tags. Our approach leverages the fact that structured knowledge of technologies can emerge from the tagging practices of millions of Stack Overflow users taken together [14, 19].

3.1 Mining Technology Associations

In this work, we consider Stack Overflow question tags as *technologies* for computer programming. Given a set of Stack Overflow questions, we use association rule mining [1] to mine technology associations from tag co-occurrences in questions. If the input set of questions contains all the Stack Overflow questions, we refer to the resulting TAN as the general TAN. If the input set of questions contains only questions that are tagged with some technologies, we refer to the resulting TAN as technology-specific TAN. If the input set of questions contains questions that are asked during a period of time, we refer to the resulting TAN as time-specific TAN.

In this work, a Stack Overflow question is considered as a transaction and the question tags as items in the transaction. As we are interested in constructing a TAN, we need to find frequent pairs of technologies, i.e., frequent itemsets that consist of two tags. A pair of tags is frequent if the percentage of how many questions are tagged with this pair of tags compared with all the questions is above the minimal support threshold t_{sup}. Given a frequent pair of tags $\{t_1, t_2\}$, association rule mining generates an association rule $t_1 \Rightarrow t_2$ if the confidence of the rule is above the minimal confidence threshold t_{conf}. The confidence of the rule $t_1 \Rightarrow t_2$ is computed as the percentage of how many questions are tagged with the pair of tags compared with the questions that are tagged with the antecedent tag t_1.

Given the mined tag association rules, we construct a TAN. A TAN is an undirected graph $G(V, E)$, where the node set V contains the tags (i.e., technologies) appearing in the association rules, and the edge set E contains undirected edges $< t_1, t_2 >$ (i.e., technology associations) if the two tags has the association $t_1 \Rightarrow t_2$ or $t_2 \Rightarrow t_1$[3]. Each edge has a confidence attribute indicating the strength of the technology association.

3.2 Detecting Technology Communities

A TAN can consist of large numbers of technologies and the associations among technologies. Some relevant technologies would be strongly connected to each other, but loosely connected to those irrelevant technologies. In graph theory, a set of highly correlated nodes is referred to as a community (cluster) in the network. In this work, we use Louvain method [5] implemented in the Gephi [4] tool to detect communities of highly correlated technologies in a TAN. The Louvain method does not require users to specify the number of communities to be detected. It uses an iterative modularity maximization method to partition the network into a finite number of disjoint clusters that will be considered as communities. Each node must be assigned to exactly one community. Intuitively, any edge in a given community has both ends in the same community contributes to increasing modularity, while any edge that cuts across communities has a negative effect on modularity.

[3] The edge is undirected because association rules indicate only the correlations between antecedent and consequent.

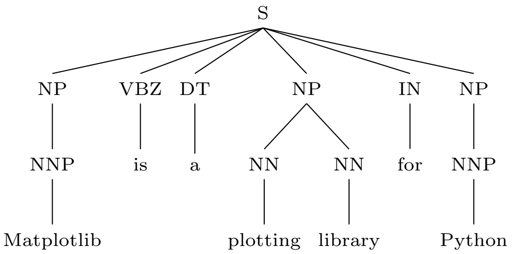


Figure 5: POS tagging and phrase chunking results of the definition sentence of the tag *Matplotlib*

3.3 Determining Technology Categories

In Stack Overflow, most tags have a brief definition called TagWiki which is collaboratively edited by the community. This mechanism is similar to Wikipedia. According to our observation, the first sentence of the tagWiki always defines the category of this tag. For example, the first sentence of the tag *Matplotlib* is "Matplotlib is a plotting library for Python"[4]. In our recent work [6], we develop the NLP methods to analyze such tag definition sentence to determine the category of a tag. We first carry out Part-of-speech (POS) tagging and phrase chunking to the sentence to get the first noun phrase after the be verb (*is/are*) and then take the last word in the phrase as the category label of the tag. As seen in Figure 5, the first phrase after *is* is *plotting library* and the last word in that phrase (i.e., library) is regarded as the category of the tag *Matplotlib*. Interested readers can refer to our paper [6] for the technical details and the evaluation of tag category analysis. As there are hundreds of fine-grained categories which will be distractions for users if we display all categories, we manually categorize them into three general categories: "library", "language" and "concept" (see Figure 2[5] for examples). The library category broadly refers to software library, framework, api, toolkit, wrapper, etc., the language category includes different programming languages, and all others are regarded as concept category such as data structures, algorithms.

3.4 Summarizing Technology Activity

For each technology in a TAN, we summarize the frequency of the corresponding tag used in the set of Stack Overflow questions. We then normalize the frequency over all the technologies in the TAN as a technology activity metric in (0, 1]. This technology activity metric is an indicator of the relative community attention to a technology in the TAN, compared with other technologies in the TAN.

3.5 Visualizing TAN

We use the Gephi tool [4] to visualize the TAN as follows (see Figure 2, Figure 3 and Figure 4 for examples). Nodes and edges in one community are shown in the same color [6]. Forceatlas2 layout [16] is used for network spatialization. This layout is especially suitable for inspecting clustering results (i.e., technology communities). The node size represents the technology activity metric. That is, the larger the node is, the more questions are tagged with the corresponding technology. The edge length can represent the strength of the corresponding technology associations. Due to the use of Forceatlas2 layout, the edge length bears no meaning in the examples of this paper.

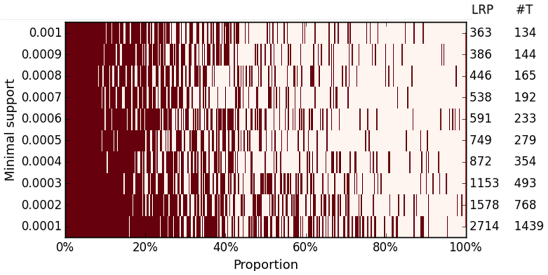


Figure 6: Coverage of tags

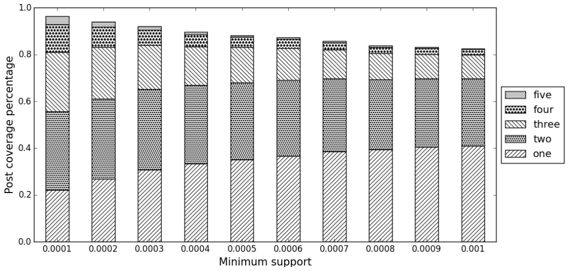


Figure 7: Coverage of questions

4. EMPIRICAL STUDY

We conduct empirical evaluation of our approach and the mined TANs using Stack Overflow data dump. In particular, we investigate the following research questions:

- RQ1: Can the mined TAN capture the important technologies from a majority of Stack Overflow questions?
- RQ2: How do different mining thresholds affect the size and modularity of the mined TAN?
- RQ3: Are the mined technology associations semantically related?
- RQ4: What are structural properties of the mined TAN?
- RQ5: How does the technology landscape evolve over time?

4.1 Dataset

In this study, we use the Stack Overflow data dump released in March 2015. The data ranges from 2008-07-31 to 2015-03-08 and contains 7.89 million questions that are attached with 2 or more tags, and 39948 unique tags from these questions. These questions and tags constitute the dataset for our evaluation.

4.2 RQ1: Coverage of Tags and Questions

The number of technologies in the mined TAN is affected by the minimal support t_{sup} and the minimal confidence t_{conf}. When t_{conf} is set to 0, all the frequent pairs of tags at a given minimal support will be included in the TAN, and thus the TAN will have the maximum number of technology at a given minimal support. This TAN defines the upper bound of the coverage of tags and questions at a given minimal support t_{sup}, which will be evaluated in this section.

[4] http://stackoverflow.com/tags/matplotlib/info

[5] Due to the size limitation, we only present library and concept categories in this paper.

[6] The Gephi tool sometimes may assign very similar colors to different communities.

In particular, we evaluate the general TAN mined at the 10 minimal support t_{sup} (0.0001 to 0.001 with increment 0.0001)[7]. For the following sections, we will use the general TAN mined at the minimal support 0.0007, because the resulting general TAN is complex enough to analyze the key characteristics of the mined TAN, meanwhile it can be clearly visualized in the paper (see Figure 4).

4.2.1 *Coverage of Tags*

To examine the coverage of tags, we rank all the tags in our dataset by their usage frequency in the set of Stack Overflow questions. We scan the ranked list of all the tags to find the tags that appear in the mined TAN at a given minimal support. We truncate the ranked list at the Lowest Rank Position (LRP) of the technologies in the TAN. Figure 6 presents the analysis results at the 10 minimal supports. Tags are ranked from left to right by decreasing frequency of use. A red line indicates that the tag ranked at this position is in the TAN at a given minimal support, while a pink line indicates that the tag is not in the TAN.

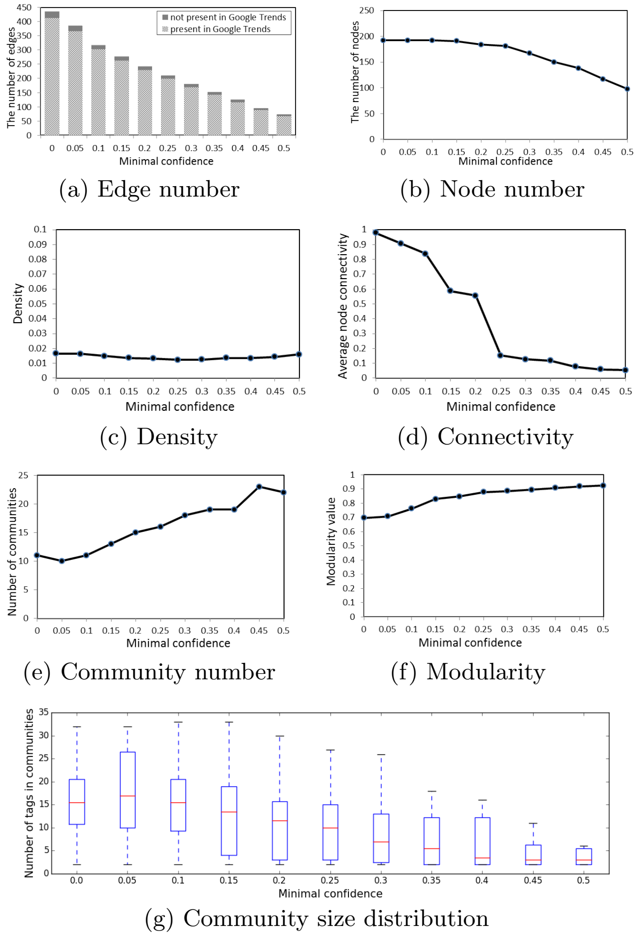
Figure 6 shows that the number of tags (#T) in the TAN and the LRP of these tags differ greatly at different minimal supports. Note that we scale the visualization of tag coverage at different minimal support to facilitate the observation of tag coverage patterns at different minimal supports. Overall, the mined TAN captures a meaningful conceptualization of important technologies than the individual tags alone. We use the general TAN mined at the minimal support 0.0001 (i.e., the bar at the bottom in Figure 6) as an example for detailed discussion.

The general TAN mined at the minimal support 0.0001 contains 1439 tags (#T). These 1439 tags account for 53% of the top 2714 most frequently used tags (LRP). 66% of these 1439 tags fall into the top 40% range of the 2714 most frequently used tags. However, about 13% of the top 40% of the 2714 most frequently used tags do not appear in the general TAN, as indicated by the pink lines within the top 40%. These tags are usually some common programming concepts such as *formatting*, *automation*, *documentation* and *numbers*. Although these common tags are frequently used as a whole, their co-occurrences with other tags are often not frequent enough because they are correlated with many technologies. As such, these common tags do not appear in the general TAN. Note that these common tags may still appear in the technology-specific TAN if their co-occurrences with a given technology is frequent enough.

34% of the 1439 tags in the general TAN scatter in the lower 60% range of the 2714 most frequently used tags. These tags usually represent features of some specific techniques, such as *django-queryset*, *android-custom-view* and *jquery-ui-draggable*. Although these tags are less frequently used than many other frequently used tags, their co-occurrences with some specific technologies (e.g., *django*, *android*, *jquery*) are often frequent. As such, these tags appear in the TAN.

4.2.2 *Coverage of Questions*

If the N tags of a question appear in the TAN, we say that the question is covered by N technologies in the TAN. Figure 7 shows the percentage of the questions in our dataset that are covered by 1-5 technologies at the 10 minimal supports. Note that the percentage is computed in an exclusive manner. That is, the questions that are covered by N technologies do not include those that are covered by $N-1$ technologies.



(a) Edge number

(b) Node number

(c) Density

(d) Connectivity

(e) Community number

(f) Modularity

(g) Community size distribution

Figure 8: The impact of minimal confidence on the size and modularity of the general TAN

We can see that although the general TAN covers only a small portion of all the tags in our dataset, it still covers a large portion of all the questions. As the minimal support increases, the coverage of questions by 3 or more technologies decreases significantly from 40% at 0.0001 to 10% at 0.001. The coverage of questions by 2 tags remains about 30% at different minimal supports. The coverage of questions by only 1 tag increases from 22% at 0.0001 to 41% at 0.001. The overall coverage of questions decreases from 96% at 0.0001 to 82% at 0.001. This suggests that most of the questions that can be covered by 3 or more technologies at lower minimal support can still be covered by the TAN at higher minimal support, but at high minimal support these questions can only be covered by 1 or 2 most frequently used tags that are used to tag large numbers of questions.

4.3 RQ2: Size and Modularity of Technology Communities

Given a minimal support, the number of technology associations in the mined TAN is affected by the minimal confidence t_{conf}. The number of technology associations consequently affects the size and modularity of technology

[7] Due to the data sparsity, too large support value results in rather small and sparse TAN. After experiments with various values, we choose this range for our evaluation.

communities in the mined TAN. Next, we analyze the impact of the minimal confidence t_{conf} on the size and modularity of the general TAN mined at the minimal support 0.0007 and the 11 different minimal confidences t_{conf} (0 to 0.5 with increment 0.05).

As shown in Figure 8(a) and Figure 8(b), the number of edges (i.e., technology associations) keeps decreasing, as the minimal confidence increases. In contrast, the number of nodes (i.e., technologies) remains unchanged until the minimal confidence increases to certain extent (0.2 in this case). After that, the number of nodes decreases roughly at the same pace as the number of edges decreases. This suggests that the increase of minimal confidence has more impact on the structure of the mined TAN than the nodes of the TAN.

We compute the density of the mined TAN (i.e., the number of edges in the TAN divided by the number of edges in a complete graph of the same number of nodes), and the average of local node connectivity (i.e., minimum number of nodes that must be removed to disconnect two nodes) of all pairs of nodes in the TAN. As shown in Figure 8(c) and Figure 8(d), the density of the knowledge graph remains low and relatively stable as the minimal confidence increases. However, the node connectivity drops sharply as the minimal confidence increases.

As shown in Figure 8(e) and Figure 8(f), the decrease of node connectivity in turn results in the increase of the number of technology communities and the modularity of technology communities in the TAN. Figure 8(g) shows the box plot of the number of tags in the detected technology communities at different minimal confidences. We can observe a trade-off between the size and modularity of technology communities. At low minimal confidence, the knowledge graph has more weak technology associations, which often results in small numbers of large communities with low modularity.

The increase of the minimal confidence can remove the weaker associations from the TAN with higher modularity. As a result, the knowledge graph becomes less connective. However, a very high confidence risks throwing away meaningful technology associations, leading to excessive partition of the TAN into many small, disconnected communities, which is often not desirable. Therefore, to produce a good balance and trade-off between the number of edges and nodes in the general TAN, the minimal confidence should be between 0.15 and 0.25.

4.4 RQ3: Semantic Distance of Technology Associations

In this section, we examine whether technology associations are meaningful by evaluating the semantic distance between the two correlated technologies in the mined TAN using the "Google distance" approach [8, 10]. Google distance is a crowd-scale method to measure the semantic distance between a set of words by analyzing search engine data. The assumption is that the co-occurrence of a set of words in the same queries is a good indicator of the semantic distance between the words. In this work, we use Google Trends [11] to evaluate the semantic distance of technology associations in the mined general TAN.

Given a technology association (i.e., an edge $< t_1, t_2 >$) in the TAN, we generate a set of search terms to query Google Trends. For example, to check whether the two technologies *php* and *facebook* are really correlated, we query the Google Trends with the search terms "php facebook". Google Trends provide the trend statistics for popular queries. For example, "php facebook" is a popular query because Facebook is built using PHP and it supports PHP APIs. If a set of search terms is not popular enough, Google Trends will provide no trend statistics.

As shown in Figure 8(a), there are a small percentage of technology associations (less than 10% at all the minimal confidences) in the TAN, which are not present in Google Trends. Lower minimal confidence values do not significantly result in more noisy technology associations. Furthermore, even the technology associations are not present in Google Trends, it does not necessarily indicate wrong associations. Take the minimal confidence 0.15 as an example. The TAN has 15 technology associations that are not present in Google Trends. 7 out of these 15 associations involves tags with specific version number such as $< doctrine2, symfony2 >$ which is not commonly searched in Google. In contrast, "doctrine symfony2" is a popular query. The other 8 associations are the results of different wording styles in Google and Stack Overflow. For example, Stack Overflow users frequently tag questions with both *knockout.js* and *javascript*, while Google users search "knockoutjs" directly without "javascript".

4.5 RQ4: Network Structure

In this section, we analyze the network structure of the general TAN and the technology-specific TANs[8] mined at the minimal support 0.0007 and the minimal confidence 0.15. We show that the general TAN and the technology-specific TANs constitute a complex, non-hierarchical, and multi-faceted technology landscape.

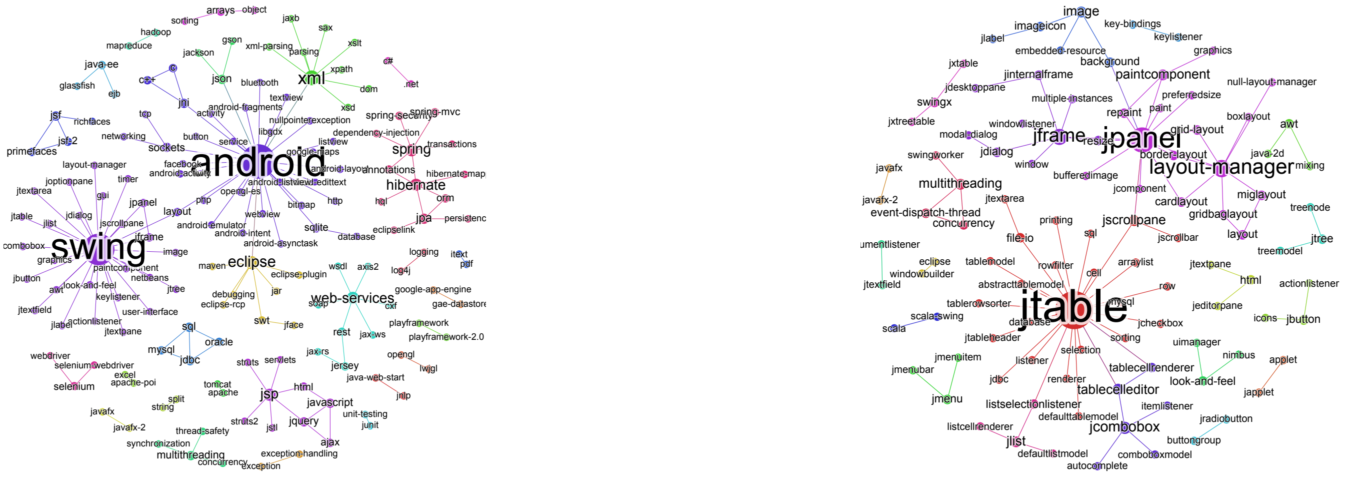
4.5.1 The General TAN

The general TAN shown in Figure 4 contains 191 technologies and 258 technology associations, which forms 13 technology communities. It provides a general overview of major software-related technologies, such as concepts (e.g., multithreading, generics, regex), languages, platforms, tools and frameworks, and the associations between the technologies. In the following discussion, we use the technology with highest degree centrality in a technology community to refer to the community. To avoid ambiguity, we refer to the community with the all-capitalized name, such as *JAVA*.

In Figure 4, we can see that 7 communities follows primarily star structure, such as *C#*, *JAVA* and *ANDROID*. These communities define a technology folksonomy [13], with a high degree core technology surrounded by some technologies that only link to the core technology. In contrast, 3 communities have complex network structure, i.e., *IOS*, *JAVASCRIPT*, *SQL*. These 3 communities have several high degree technology, such as *ios, iphone, objective-c* in the *IOS* community. Furthermore, technologies in these communities link to not only the high-degree technologies but also each other. They form a network of relevant technologies for Apple application development, web development, and database management, respectively.

7 communities form a connected component in the center of Figure 4. These 7 communities contain 142 (74%) technologies that are relevant to *C#*, *JAVA*, *ANDROID*, *PYTHON*, *JAVASCRIPT*, *PHP*, and *SQL*. These communities are linked by some betweenness technologies that have high betweenness

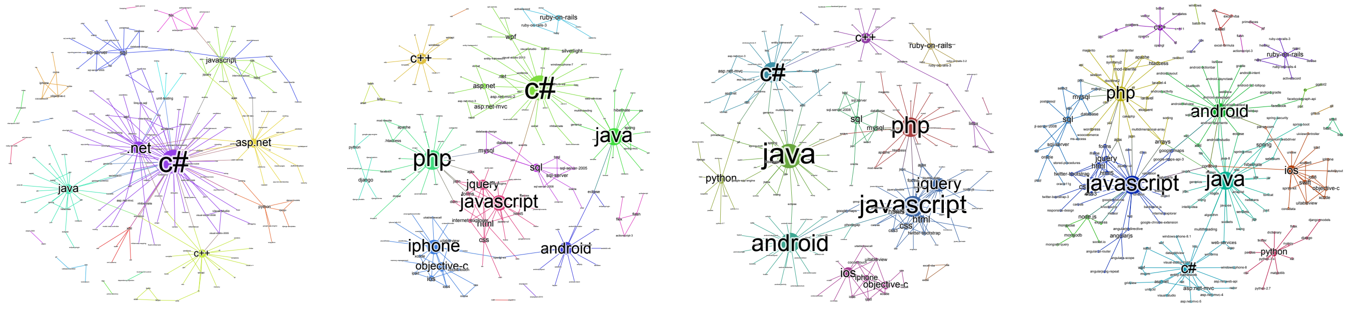
[8] The node size represents degree centrality in Figure 4, Figure 9(a), and Figure 9(b).



(a) Java

(b) Swing

Figure 9: *java*-specific TAN and *java-swing*-specific TAN



(a) 2009-01

(b) 2011-01

(c) 2013-01

(d) 2015-01

Figure 10: The evolution of the general TANs

centrality, for example, *servelets* and *eclipse* between *JAVA* and *ANDROID* community, and *JAVASCRIPT* community, and *html5*, *css* and *json* between *JAVASCRIPT* and *PHP* community. These betweeness tags reflect the relations betweenness relevant techniques in practice. For example, *eclipse* is the popular IDE for Java development and is the standard Android development environment. *html5*, *css* and *json* are the key technologies for web applications.

6 communities are isolated, among which 3 communities (*IOS*, *C++*, and *RUBY-ON-RAILS*) are medium-size, while the other 3 communities are very small (*EXCEL-VBA*, *GIT*, and *FLASH* at the top-right corner of Figure 4). They are not connected with other communities because relationship among the inner components are much stronger than the outer ones.

4.5.2 Technology-Specific TANs

We use *java*-specific TAN (Figure 9(a)) and *java-swing*-specific TAN (Figure 9(b)) to illustrate technology-specific TANs. We can see that more fine-grained technologies are captured in the TAN when we "zoom-in" a specific technology. A technology associated with *java* in the general TAN, such as *web-services*, *spring*, *hibernate*, and *swing*, becomes a technology community in the *java*-specific TAN. Furthermore, some technologies that are not present in the general TAN are captured in the *java*-specific TAN, such as *primefaces*.

Most technology communities in the *java*-specific TAN are isolated from each other. This is consistent with the star structure of the *JAVA* community in the general TAN. However, some un-correlated technologies in the general TAN become correlated in the *java*-specific TAN as technology associations become frequent enough from a specific technology perspective. For example, *spring-mvc* and *spring* are linked to *java* but not linked to each other in the general TAN, while *spring-mvc* is an entity in the community *SPRING* in the *java*-specific TAN.

Similar observations can be made when comparing *java-swing*-specific TAN with *java*-specific TAN. A distinct difference is that the *SWING* community in the *java*-specific TAN is a star structure, but as more associations are captured in the *java-swing*-specific TAN, several major communities that represent the key Java Swing classes (e.g., *jframe*, *jpanel*, and *jtable*) form a large connected component.

It is important to note that the general TAN and technology-specific TANs are non-hierarchical and multi-faceted. For example, the *java*-specific TAN has *ANDROID* community and *JSP* community. The *JSP* community consists of *javascript* and *jquery*. However, the *ANDROID*, *JAVASCRIPT* and *JQUERY* are the communities at the same level as the *JAVA* community in the general TAN. Furthermore, neither *javascript* nor *jquery* has direct association with *java* or *jsp* in the general TAN. But as *javascript*, *jquery* and *jsp* are always cooperatively adopted for web development, *javascript* and *jquery* appear in *JSP* community of *java*-specific TAN.

4.6 RQ5: Network Evolution

We mine the general TAN from the questions available

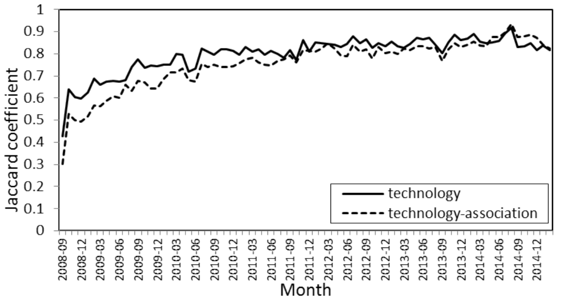


Figure 11: Similarity of Consecutive General TANs.

at the end of every month for the period August 2008 to February 2015. We visualize and compare the resulting 79 general TANs. Figure 10 shows four of these 79 general TANs mined at January of the year 2009, 2011, 2013 and 2015. Overall, several major programming languages are present all the times in the general TAN, i.e., *C#*, *JAVA* and *C++*, while some technologies gradually disappear, e.g., *ASP.NET*. Web development (*PHP*, *JAVASCRIPT*) and mobile development (*IOS*, *ANDROID*) technologies has been growing rapidly. Technologies in the general TAN and the structure of the general TAN change fast in the first few months of Stack Overflow, and then gradually become stable in late 2011. From then on, the number of technology communities in the general TAN and the structure of these communities remain stable with only small changes over time.

To confirm our qualitative observation, we compute the Jaccard coefficient of the general TANs mined from the two consecutive months. Let $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ be the two TANs, we compute both technology Jaccard coefficient $\frac{|V_1 \bigcap V_2|}{|V_1 \bigcup V_2|}$, and technology-association Jaccard coefficient $\frac{|E_1 \bigcap E_2|}{|E_1 \bigcup E_2|}$. Figure 11 shows that both technology Jaccard coefficient and technology-association Jaccard coefficient increase rapidly from the September 2008 to March 2009, and then increase slowly till November 2011. From then on, the Jaccard coeficient becomes stable and remain around 0.9 with only small fluctuations. This result is consistent with our qualitative observation.

5. THE TECHLAND WEBSITE

We develop a `TechLand` website[9] that displays a technology page for a given technology in the mined technology landscape. The technology page shows the technology description extracted from the TagWiki, the mined TAN, and other related information extracted from the Stack Overflow. The user can search the technologies in the mined technology landscape or navigate from one technology page to another in the graphical TAN. The website also allows the user to compare the TAN of several technologies side by side.

We release our website to the public and post this news on several programming-related websites (e.g., http://stackapps.com/questions/6569). According to the Google Analytics[10], more than 1,000 users from 64 countries visit our site (Figure 12) from Sept 4th 2015 to Jan 13th 2016. These users on average browse 4.24 pages in each session for 6 minutes and they browse 6,674 pages in total (including the homepage). The usage statistics show initial evidence of the public interests in technology landscape services.

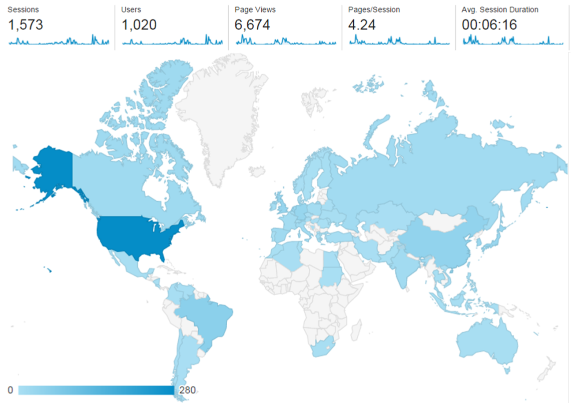


Figure 12: The Google Analytics of our website

To investigate the user navigation pattern in 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 these 290 users, about 50 users returned days or weeks later to use our website again.

We observe some interesting exploration history in the web logs of these 290 users. For example, the user 162 first visited the *nlp* page and then double-clicked the *machine-learning* node in the *nlp* TAN. 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* TAN. Then, the user further double-clicked the *neural-network* node in the *machine-learning* TAN to navigate to the *neural-network* page. The *neural-networdk* node is very obvious in the *machine-learning* TAN, as neural network is one of the popular machine learning techniques. From the *neural-networdk* page, the user navigated to the *theano* page. Theano is an efficient numerical computation library for Python. Such exploration history indicates that an overview of technology landscape could provide information scents and guided navigation for the users to explore the technology landscape and find the desired information.

When designing the website, we expect that users would first search and view some technology pages and then compare the 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 TAN of these tools side by side[11]. This usage pattern indicates the need for comparing similar technologies when exploring the technology landscape for certain tasks.

Overall, the field deployment did not lead to as much usage as we had hope. However, the usage data of our website, albeit very limited, are promising, given that we posted only brief announcements, performed no training, and many users likely just visited to satisfy their curiosity. The initial results demonstrate both the needs and the interests in some technology landscape services that our approach supports.

[9] https://graphofknowledge.appspot.com/

[10] As most search engine robots do not activate Javascript, robot traffic is not counted in Google Analytics [12]

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

Table 1: 3 types of questions in user study

Type	Question
overview	What are the best overviews for cloud technology
	A good resource for an overview of web technologies
	Technical architecture diagram for an iPhone app
concept	What are the top 3 main concepts in WPF
	What are the core concepts in functional programming
	What are best tools/concepts/things to be a better java programmer
library	What are some good OpenID libraries
	What are your favorite JavaScript libraries/scripts to create tooltips
	What languages and libraries should I use to work with Gmail?

6. USER STUDY

Finally, we report a user study of our website to evaluate the usefulness of our tool.

6.1 Experiment design

According to our observation of the website visiting data, our site could be helpful to answer three types of technology-landscape questions, i.e., overview-related, concept-related and library-recommendation questions. Thus, we use several keywords such as *overview, concept* and *libraries* to search the questions in Stack Overflow and randomly sample 9 questions (3 questions for each type) for this user study. Table 1 summarizes these 9 questions. For each question, we take all its tags as the query to generate the corresponding TANs as our answer to the question.

We recruit 7 PhD students from our school who have at least 4-year experience in programming to participant in this study. We ask the participants to compare the information of the TANs in our website with the original textual answers to these questions. Then they are asked to mark three metrics for each question on 5-point likert scale (1 being the worst and 5 being the best), i.e., accuracy, coverage and satisfaction, after inspecting the information in our TAN and reading the original question answers.

6.2 Results

Figure 13 summarizes the average scores of the three metrics that participants give for the original question answers and our TechLand answers. The higher mark means that participant are more satisfied. Overall, our TechLand answers have higher scores than the original question answers. For accuracy, the information in our TAN and the technology categorization are slightly better (about 10%) than the original answers. However, for coverage, our TAN can provide a more comprehensive and integrated technology overview than the original unorganized answers, each of which usually covers only a small part of the whole technology landscape. Therefore, the score for the coverage of our TAN is 29.6% higher than that for Stack Overflow answers. Our TAN also results in the better overall satisfaction. These results, albeit limited, demonstrate the usefulness of our TAN and the website for answering the technology landscape questions.

In addition to the quantitative investigation, we also collect some feedbacks from participants about our website and the Stack Overflow answers. Participants suggest that our TAN and the website can provide a clear, structured overview of related technologies which can guide users to explore and learn new technologies. This can effectively complements the textual, unstructured question answers, whose natural-language discussions and external links provide more contextual information but very hard to follow and aggregate due to the informal nature of the discussions.

7. RELATED WORK

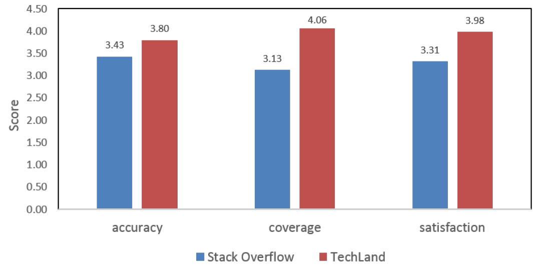


Figure 13: Average score after user study

Tagging supports the categorization of information using user-defined, open-ended vocabularies, as opposed to predefined, fixed taxonomies. It is used by many social computing systems, in which users tag objects such as web sites (e.g., Delicious), photos (e.g., Flickr), research papers (e.g., Connotea), software projects (e.g., Freecode, Maven), and questions (e.g., Stack Overflow, Quora). Furthermore, tagging has also been integrated in software development process. Storey et al. [25] develop the TagSEA tool that uses the ideas of social tagging to support collaboration in asynchronous software development. Treude and Storey [27] show that tagging of work items in IBM Jazz can improve team-based software development practices. Tags in these systems are often presented in a ranked list or tag cloud [28] which show only the usage frequency of tags, but not the relations among tags.

Many studies have shown that structured knowledge can emerge from social tagging systems [14, 17, 19]. Hierarchical clustering techniques have been applied to induce taxonomies from collaborative tagging systems [15, 24], and from software project hosting site Freecode [29]. Schmitz analyzes association rule mining results to infer a subsumption based model from Flickr tags [23]. Sanderson and Croft [22] analyze the co-occurrence of words to derive concept hierarchies from text. Different from these taxonomy mining techniques, our technology associative network is a complex, non-hierarchical, and multi-faceted network.

Tian et al. [26] construct a software-specific related words by computing the word co-occurrence weights in a corpus of Stack Overflow questions. Yang and Tan [30] infer semantically related words from software source code. The goal of these two works is to build software-specific dictionary to determine associative meanings between software-specific technical terms. However, such software-specific dictionaries cannot present developers an overview of technology landscape.

To understand the use of specific technologies and the trends of the technologies, we demonstrate the correlation of Stack Overflow and Google Trends [7]. Barua et al. [3] uses topic model technique LDA to discover the main topics present in developer discussions in Stack Overflow. Their results match our analysis of the evolution of the technology landscape (see Section 4.6). Their analysis involves manual classification of topic-related questions, while our analysis is based on the TAN automatically mined from questions.

Although software engineering community has a long history of studying graphical software models [9, 21], the concept of knowledge graph, the relevant mining techniques, the application of knowledge graph in software engineering context have not been widely adopted in this community. Our work attempts to mine TAN from Stack Overflow question tags. We believe more attention along this line of research is needed to improve the developers' life on the Internet.

8. CONCLUSION

In this paper, we present a data mining technique for mining technology landscape from the "by-product" (i.e., tags) of the Q&A practices in Stack Overflow. Our evaluation shows that the mined technology landscape can provide an aggregated view of a wide range of technologies, the complex relationships among the technologies, and the trend of the technologies, which reflect the practices of a large community of developers. We also introduce our website for accessing the mined technology landscape. The website usage data, albeit limited, provides initial evidence of the interests in and the usefulness of the mined technology landscape. In the future, we will continuously collect web usage data by Google Analytics, and also collect more fine-grained user interaction data such as cursor hover and right click in the TAN. Such interaction data will help us improve the website design and usability to make the information in the mined technology landscape more easily accessible. We will also enrich the technology landscape with more entities and richer semantics, and develop more knowledge-graph based applications.

Acknowledgment

The authors would like to thank Lei Han for the help with the UI design of our website. This work was partially supported by MOE AcRF Tier1 Grant M4011165.020.

9. REFERENCES

[1] R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. In *VLDB*, volume 1215, pages 487–499, 1994.

[2] L. Bao, J. Li, Z. Xing, X. Wang, X. Xia, and B. Zhou. Extracting and analyzing time-series hci data from screen-captured task videos. *Empirical Software Engineering*, pages 1–41, 2016.

[3] 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*, 19(3):619–654, 2014.

[4] M. Bastian, S. Heymann, M. Jacomy, et al. Gephi: an open source software for exploring and manipulating networks. *ICWSM*, 8:361–362, 2009.

[5] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[6] C. Chen, S. Gao, and Z. Xing. Mining analogical libraries in q&a discussions -incorporating relational and categorical knowledge into word embedding. In *The 23rd SANER*, pages 338–348. IEEE, 2016.

[7] C. Chen and Z. Xing. Towards correlating search on google and asking on stack overflow. In *The 40th COMPSAC*, pages 83–92. IEEE, 2016.

[8] R. L. Cilibrasi and P. M. Vitanyi. The google similarity distance. *TKDE*, 19(3):370–383, 2007.

[9] J. Ferrante, K. J. Ottenstein, and J. D. Warren. The program dependence graph and its use in optimization. *ACM TOPLAS*, 9(3):319–349, 1987.

[10] R. Gligorov, W. ten Kate, Z. Aleksovski, and F. Van Harmelen. Using google distance to weight approximate ontology matches. In *WWW*, pages 767–776. ACM, 2007.

[11] Google trends. https://www.google.com.sg/trends/.

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

[13] T. Gruber. Ontology of folksonomy: A mash-up of apples and oranges. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 3(1):1–11, 2007.

[14] H. Halpin, V. Robu, and H. Shepherd. The complex dynamics of collaborative tagging. In *WWW*, pages 211–220. ACM, 2007.

[15] D. Helic, M. Strohmaier, C. Trattner, M. Muhr, and K. Lerman. Pragmatic evaluation of folksonomies. In *WWW*, pages 417–426. ACM, 2011.

[16] M. Jacomy, S. Heymann, T. Venturini, and M. Bastian. Forceatlas2, a continuous graph layout algorithm for handy network visualization. *Medialab center of research*, 560, 2011.

[17] G. Macgregor and E. McCulloch. Collaborative tagging as a knowledge organisation and resource discovery tool. *Library review*, 55(5):291–300, 2006.

[18] 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 *28th ICSM*, pages 25–34. IEEE, 2012.

[19] V. Robu, H. Halpin, and H. Shepherd. Emergence of consensus and shared vocabularies in collaborative tagging systems. *ACM TWEB*, 3(4):14, 2009.

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

[21] J. Rumbaugh, I. Jacobson, and G. Booch. *Unified Modeling Language Reference Manual, The*. Pearson Higher Education, 2004.

[22] M. Sanderson and B. Croft. Deriving concept hierarchies from text. In *SIGIR*, pages 206–213. ACM, 1999.

[23] P. Schmitz. Inducing ontology from flickr tags. In *Collaborative Web Tagging Workshop at WWW*, volume 50, 2006.

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

[25] M.-A. Storey, L.-T. Cheng, I. Bull, and P. Rigby. Shared waypoints and social tagging to support collaboration in software development. In *Proceedings of the 20th CSCW*, pages 195–198. ACM, 2006.

[26] Y. Tian, D. Lo, and J. Lawall. Automated construction of a software-specific word similarity database. In *CSMR-WCRE*, pages 44–53. IEEE, 2014.

[27] C. Treude and M. Storey. How tagging helps bridge the gap between social and technical aspects in software development. In *ICSE*, pages 12–22. IEEE, 2009.

[28] F. B. Viegas, M. Wattenberg, and J. Feinberg. Participatory visualization with wordle. *IEEE TVCG*, 15(6):1137–1144, 2009.

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

[30] J. Yang and L. Tan. Inferring semantically related words from software context. In *MSR*, pages 161–170. IEEE, 2012.