2016 IEEE 40th Annual Computer Software and Applications Conference

Towards Correlating Search on Google and Asking on Stack Overflow

Chunyang Chen and Zhenchang Xing
School of Computer Engineering, Nanyang Technological University, Singapore
CHEN0966@e.ntu.edu.sg; ZCXING@ntu.edu.sg

***Abstract*—Search engines and Question and Answer (Q&A) sites are the two commonly used ways for developers to seek information on the web. In this paper, we ask whether the questions developers ask on Q&A sites correlate with the information developers search for using search engines. We report on our empirical study to investigate the correlations of the 185 popular technical terms developers search on Google and ask on Stack Overflow using search statistics obtained from Google Trends over a 574-weeks span and question statistics derived from Stack Overflow Data Dump over a 300-weeks span. Our study shows that technical terms searched and asked have strong correlation over time. Search and asking of newer, specific technical terms have stronger correlation, compared with older, general technical terms. We have developed a web interface for accessing our dataset and empirical results available at http://comparetrend.appspot.com/. Inspired by our empirical results, we present future directions that can harness Stack Overflow as sampled data for supporting time-aware search and semantic search.**

I. Introduction

During software development, developers frequently encounter the situation where the knowledge they possess is insufficient to complete the task. Such knowledge inadequacy leads to the information need [1], [2] for web documents providing knowledge on software engineering. Over the years, numbers of websites have been established to meet this need, such as Github (open source code), codeproject.com (technical tutorials), and Stack Overflow (technical Q&As). Developers usually seek the web information in two ways: search engines (e.g. Google) and Q&A sites (e.g. Stack Overflow)

With search engines, developers formulate their information needs in a query consisting of several keywords. But it was shown that queries presented to search engines often cannot accurately describe the searchers' information needs [1], [3], [4], because keywords are often ambiguous and multifaceted [5]. Consequently, search engines may not return relevant information that can satisfy the developers' information needs. Sometimes, even with the accurately formulated queries, successful research still cannot be achieved due to the lack of required information on the web.

With the advent of Web 2.0, Q&A sites have been launched to support social information seeking [6], [7]. Unlike search engine, Q&A sites allow the users to express their information needs in detailed questions. Most of these questions will be answered by other users who are often experts in that field. Stack Overflow is the most popular Q&A site for computer programming. Since its launch in August 2008, Stack Overflow has accumulated millions of questions and answers. These questions and answers cover many aspects of computer programming, from programming languages, platforms, tools, frameworks, APIs, to external links to other online documents such as books, tutorials and open source projects.

As questions and answers in Stack Overflow are indexable by search engines like Google, the answers provided for the previously asked questions on Stack Overflow can be retrieved as search results for satisfying the developers' information needs in response to their new searches. In this paper, we ask: are there any relationships between the questions developers ask on Stack Overflow and the information developers search on search engines?

Some researchers [8], [9], [10] investigate the use of social media information (e.g., Facebook, Youtube, Twitter) to support searching over travel information and trending events. Adar et al. [11] report that queries raised by certain web users in search engine would be reflected in their social media like blogs and posted articles. Fourney and Morris show that Q&A sites (MSDN Forum and Stack Overflow) users typically conduct online search before asking or answering a question [12]. In the context of software engineering, Parnin et al. [13] and Kavaler et al. [14] investigate the correlation of APIs used in open source projects and discussed in Stack Overflow. However, there have been no studies dedicated on the relationships between the queries developers use in search engines and the questions developers ask in Q&A sites.

To fill this gap, we carry out an empirical study using Google Trends and Stack Overflow data dump. We consider Google query keywords and Stack Overflow question tags as technical terms of computer programming, such as programming languages, platforms, tools, frameworks, and APIs. We collect 185 popular technical terms and their frequent co-occurring terms from Google Trends and Stack Overflow. These technical terms are regarded as programming knowledge developers search on Google and ask on Stack Overflow. The change of the popularity of a technical term over time is regarded as the search trend and asking trend of that technical term, which reflects the change developer's interest in the corresponding programming knowledge. Our data collection process and dataset will be described in Section II.

Using this dataset, we answer the following research questions: 1) To what extent technical terms developers search and ask overlap (Section III); 2) Do search trend and asking trend of a technical term exhibit similar trend patterns (Section IV);

0730-3157/16 $31.00 © 2016 IEEE
DOI 10.1109/COMPSAC.2016.210

3) What are the correlation and delay between search trend and asking trend of a technical term (Section V); 4) What are the relationships between search trend and asking trend of technical terms representing different versions of one programming technique (Section VI).

Our study shows that programming knowledge developers search on Google and ask on Stack Overflow correlate well, in terms of the overlap of technical terms searched and asked, and their temporal patterns and trends. Search and asking of newer, specific technical terms have larger overlap and stronger correlation, compared with that of older, general technical terms. Search and asking trends of several versions of one programming technique show the transition of the popularity of the technique between subsequent versions. Our findings suggest that Stack Overflow can be exploited as sampled data to study the temporal property and semantics of developers' information needs and online programming documents to enhance topic-based web search. We discuss our findings and present future directions for time-aware search and semantic search in Section VII.

II. Dataset

This study is based on the Google Trends [15] data for the period of Jan 04, 2004 to Jan 03, 2015 (574 weeks) and the Stack Overflow data dump [16] for the period of July 31, 2008 (when Stack Overflow was launched) to May 04, 2014 (300 weeks). This section describes our data collection process and the resulting dataset.

A. Selecting Popular Technical Terms

We collect 7.2 million questions from Stack Overflow data dump. Stack Overflow requires question askers to tag their questions with 1-5 tags. We consider these tags as technical terms of computer programming that askers consider as relevant to their questions. These technical terms usually represent programming language (e.g., Java, C#, Python), development environment (e.g., Eclipse, Visual Studio), application platform (e.g., Android, iOS), framework/library (e.g., .NET, jQuery), and application features (e.g., android-layout).

We collect 36997 distinct tags from the 7.2 million questions. As the community of Stack Overflow commits to merge synonym tags by corresponding master ones, no tag synonyms will be contained in our dataset. We rank the 36997 tags by their usage frequencies, i.e., the number of questions tagged with a given tag. The top 400 most frequently used tags cover 95.6% questions in the data dump, i.e., 95.6% questions are tagged with at least one of these 400 tags. We select these 400 tags as technical terms for further analysis.

We transform the selected 400 tags into the 400 search items to query Google Trends. Google Trends [15] provides the statistics of a search item that people use as query to search Google. One-word tag is directly transformed into a search item with one keyword. But Stack Overflow tags can concatenate several words with -, for example, *visual-studio-2010*. When querying Google Trends, if serachs keywords are concatenated with -, results will include searches containing the first keyword, but excludes other keywords. Thus, we replace - with blank space like that a tag *visual-studio-2010* is transformed into a search item with three keywords *visual studio 2010*. In this paper, tags like *visual-studio-2010* and search items like *visual studio 2010* are used interchangeably to represent the same technical terms.

We note that 190 of the 400 search items have a more general language usage that is not restricted to the computer programming domain. For example, *java* can mean coffee or island. *iPhone* can appear in many news or online shops. We remove such 190 search items to avoid noise in Google Trends data. But, if the search item contains several keywords including the keyword like *java*, such search item is retained. Then 210 search items are used to query Google Trends.

We enter the search item with quotation marks to Google Trends to obtain search statistics that include only queries that match the exact search item [17]. For example, given the search item *"asp.net web api"*, Google Trends returns the statistics of the exact query *"asp.net web api"* that people use to search Google. If the given query is not popular enough, Google Trends will return "no enough search volume". So we further filter out 25 out of the 210 search items, such as *mode rewrite*, *google maps api 3*, that do not have enough search volume on Google Trends, i.e., no top searches returned (as of Jan 03, 2015). Finally, we obtain 185 popular technical terms for studying the correlations between the information developers search on Google and the questions developers ask on Stack Overflow.

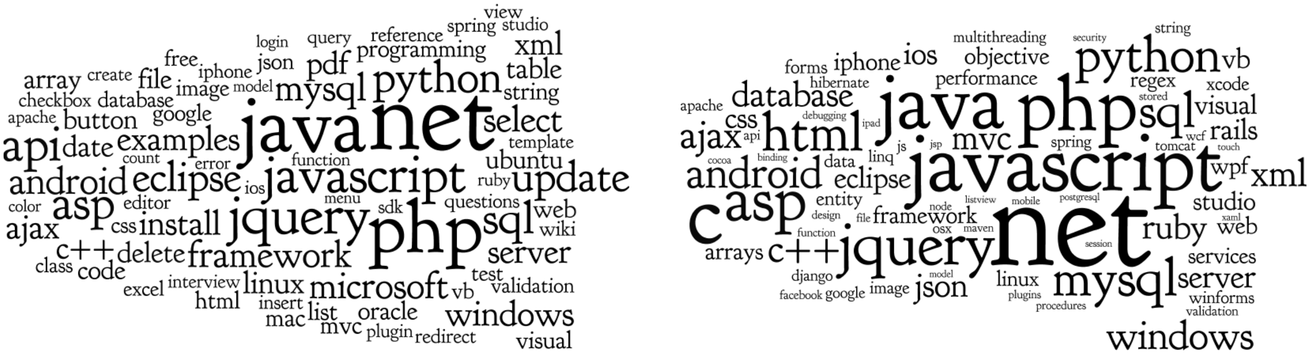
B. Collecting Frequent Search and Asking Technical Terms

For each of the 185 technical terms, we collect a set of its frequent co-occurring keywords developers use to search Google (i.e., **frequent search terms**), and a set of its frequent co-occurring tags developers use to tag questions (i.e., **frequent asking terms**). We consider these frequent search and asking terms as programming knowledge developers frequently search on Google and ask on Stack Overflow respectively.

Frequent search terms: Given a search item, Google Trends returns a list of top searches, i.e., popular search queries that are related to the search item entered [2]. We collect a set of unique keywords in the returned top searches. We refer to this set of keywords as frequent search terms related to the search item. For example, given the search item *javascript*, Google Trends returns 50 top searches (as of Jan 03, 2015). These 50 top searches contain 45 javascript-related searches such as *jquery*, *array javascript*, *javascript string*, 4 searches of other web-based technologies such as *html*, *php*, *json*, *ajax*, and 1 search of other programming language *java* . The set of keywords in these 50 top searches consists of 42 unique keywords, including *javascript* , *array, string, html, php, java*, etc. These 42 keywords constitute a set of frequent search terms related to the search item *javascript*. Note that this

[1] The words like *java* are from frequent search and asking terms, not the ambiguous search items removed.

[2] Google does not release technical details about how top searches are determined.



(a) Google

(b) StackOverflow

Fig. 1. Word cloud of the top 80 most frequent search terms and the top 80 most frequent asking terms related to the 185 technical terms [1]

set of frequent search terms contains the keyword *java* that is considered as ambiguous when selecting popular search items. However, as this *java* is extracted from the top searches related to the technical term *javascript*, we do not consider it as ambiguous when collecting frequent search terms.

Frequent asking terms: Given a tag, we collect all the questions tagged with the given tag in Stack Overflow data dump. We calculate the frequency of all the tags that co-occur with the given tag in these questions. In this study, the number of co-occurring tags of a given tag is always larger than the number of keywords in the top searches of the corresponding search item. Thus, we collect the top N frequent co-occurring tags of the tag (N be the number of unique keywords in the top searches of the corresponding search item). We regard this set of frequent co-occurring tags as frequent asking terms related to the tag. For example, the set of frequent asking terms related to the tag *javascript* contains 42 frequent co-occurring tags of *javascript*, such as *jquery, html, php, css, ajax*, etc.

Standardization of frequent search and asking terms: We note that developers often use different wording style in Google keywords and Stack Overflow tags to represent the same technical terms, for example *python2.7* and *python-2.7*. Based on our observation, we standardize frequent search and asking terms as follows: 1) split the words by char-digit switchings; and 2) split the non-digit words by delimiters such as -, . (except for technical terms like *.net*), and _. As such, both *python2.7* and *python-2.7* will be standardized as the same technical term *python 2.7*. After standardization, the set of frequent search terms and the set of frequent asking terms become comparable for studying the similarity and differences between frequent search terms and frequent asking terms related to the 185 technical terms.

C. Building Search and Asking Trends

For each of the 185 technical terms, we build a sequence of time-ordered weekly **search popularity** of the term on Google (i.e., *search trend*), and a sequence of time-ordered weekly **asking popularity** of the term on Stack Overflow (i.e., *asking trend*). Search trend and asking trend of a technical term reveal the change of developers' interest in it over time. All of them can be seen in our website: http://comparetrend.appspot.com/

Search trend: We use Google Trends to compute search trends of the 185 search items for the period Jan 04, 2004 to Jan 03, 2015 (574 weeks). Google Trends first computes a percentage of how many searches have been done for the entered search item compared to the total number of searches during a particular week. It then normalizes the weekly percentage by setting the week with the highest weekly percentage at 100, and then normalizing the percentage of other weeks to an integer index (0 to 100). This weekly integer index of the search item represents the search popularity of the corresponding technical term in a particular week.

Asking trend: We build asking trend of the 185 tags for the period from Aug 03, 2008 to May 03, 2014 (300 weeks). We first compute a percentage of how many new questions have been associated with a given tag compared to the total number of new questions asked during a particular week. The weekly percentage is then normalized by setting the week with the highest weekly percentage as 100, and normalize the percentage of other weeks to an integer index (0 to 100). This weekly integer index of the tag represents the asking popularity of the corresponding technical term in a particular week.

III. Technical Terms Searched and Asked

To understand the overlap and differences of programming knowledge developers search and ask, we compare the similarity between frequent search terms and frequent asking terms related to the 185 technical terms.

A. Overall Comparison

We merge the 185 sets of frequent search terms into an overall set that keeps 835 unique frequent search terms which appear in at least two sets. Similarly, we merge the 185 sets of frequent asking terms into an overall set of frequent ask terms. And 595 unique frequent asking terms are preserved which appear at least in two sets. 386 technical terms (i.e., 46% of 835 frequent search terms and 65% of 595 frequent asking terms) are the same in the two overall sets of frequent terms. 449 terms appear only in the overall set of frequent search terms, while 209 terms appear only in the overall set

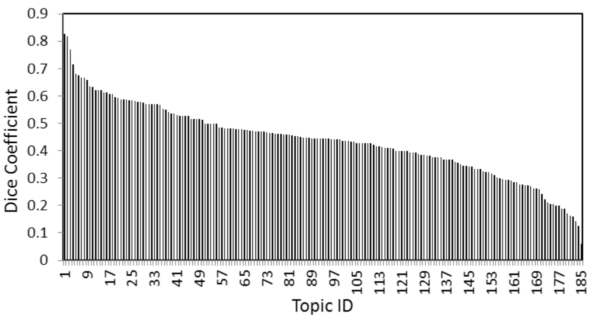


Fig. 2. The Dice coefficient of the set of frequent search terms and the set of frequent asking terms (Ranked from high to low)

of frequent asking terms. This suggests that technical terms which developers frequently search on Google are broader than that frequently asked on Stack Overflow.

Fig. 1 shows the top 80 most frequent technical terms in the overall set of frequent search terms and frequent asking terms. We can see that most technical terms people frequently search on Google and frequently ask on Stack Overflow overlaps. The overlapping terms include popular programming languages, frameworks, and platforms. Furthermore, we can observe that the word relative frequency (as indicated by the font size) of the same technical term in the two word clouds is similar. This suggests that developers' interest in these technical terms is similar on Google and on Stack Overflow.

We can also observe the differences between frequent search terms and frequent asking terms in Fig. 1. Frequent search terms often include general terms, such as *api*, *code*, *download*, *examples*, *programming*, *tutorial*, *reference*, and *wiki*, which reflect the general interests of developers searching for programming knowledge. In fact, the terms *download* and *tutorial* appear in 156 sets and 103 sets of frequent search terms respectively, which are much more frequent than other frequent search terms. We do not include *download* and *tutorial* in the word cloud of frequent search terms, because they significantly distort the comparison of relative word frequencies of the two word clouds. In contrast, frequent asking terms include more specific technical terms such as *forms*, *hibernate*, *django* and *linq* .

B. Comparison of Individual Programming Topics

For each of the 185 technical terms, we compute the Dice coefficient [3] of the set of frequent search terms S and the set of frequent asking terms A related to the technical term, i.e.,

$$Dice(S,A) = \frac{2 \times |S \bigcap A|}{|S| + |A|}$$

As shown in Fig. 2, the Dice coefficient of frequent search terms and frequent asking terms of 18 (10%) technical terms is above 0.6, that of 105 (57%) technical terms is between 0.4 and 0.6, that of 54 (29%) technical terms is between 0.4 and 0.2, and that of 8 (4%) technical terms is below 0.2.

We find that the higher the Dice coefficient, the more specific the technical terms tends to be, and the smaller the set of frequent search terms and the set of frequent asking terms. For example, the set of frequent search terms and frequent asking terms related to *asp.net-mvc-2* contains only 6 terms (e.g., c#, jquery, mvc) and the Dice coefficient of the two sets is 0.67. The set of frequent search terms and frequent asking terms related to *api.net web api* contains only 11 terms and the Dice coefficient is 0.73. In contrast, the lower the Dice coefficient, the more general the technical terms tends to be, and the larger the set of frequent search terms and the set of frequent asking terms. For example, the set of frequent search terms and frequent asking terms related to *python* contains more than 40 terms and the Dice coefficient is below 0.3.

Summary: Most of technical terms that developers search on Google and ask on Stack Overflow overlaps. Developers search Google for broader technical terms, while they ask questions regarding more specific technical terms on Stack Overflow. Frequent search terms and frequent asking terms related to a general technical term (e.g., *python*) can be diverse, while those related to a specific technical term (e.g., *asp.net-mvc-2*) tend to be more focused.

IV. Patterns of Search and Asking Trends

Next, we investigate what patterns search trend and asking trend of the 185 technical terms exhibit and how many technical terms exhibit the same search and asking trend patterns. In this section we focus on qualitative analysis of long-term trend patterns displayed in search trends and asking trends, not the weekly fluctuation in search trends and asking trends. In the next section, we perform quantitative analysis of correlations and delays of search trends and asking trends.

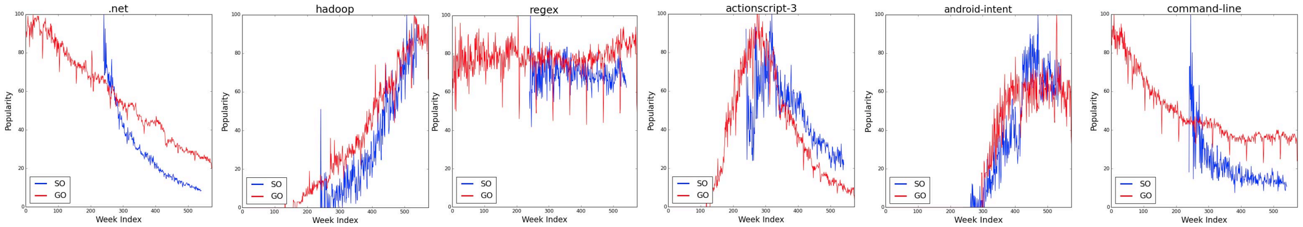
A. Observing Trend Patterns

Search (or aksing) trend of a technical term is a time-series of observation of search (or asking) popularity of the term for the period of time we study. Inspired by Kulkarni's work [18], we manually examine the search trend and asking trend of the 185 technical terms to identify long-term spike(s), seasonality and progression patterns of search trends and asking trends.

A *spike* occurs when search (or asking) popularity of a technical term increases for a period of time and then decreases afterwards. We find that search trend and asking trend of all the 185 technical terms have only zero or one spike: 124 search trends and 129 asking trends have 0 spike (e.g., Fig. 3(a), 3(b), 3(c), 3(e), 3(f)), while 61 search trends and 56 asking trends have 1 spike (e.g., Fig. 3(d)). *Seasonality* reveals a repetitive pattern of spikes for a period of time. As search trends and asking trends have only zero or one spike, we do not observe seasonality in search trends and asking trends.

A search (or asking) trend may progress in a sequence of stages. Each progression stage has a distinct trend direction for a period of time (i.e., *up*, *flat*, and *down*). We find that search trend and asking trend of all the 185 technical terms have at most two progression stages. We identify 6 *progression patterns*: Down, Up, Flat, UpDown, UpFlat

[3] We also compute other similarity measurements like Jaccard coefficient, but find no fundamental difference from Dice coefficient.



(a) Down (b) Up (c) Flat (d) UpDown (e) UpFlat (f) DownFlat

Fig. 3. Examples of progression patterns of search and asking trends.

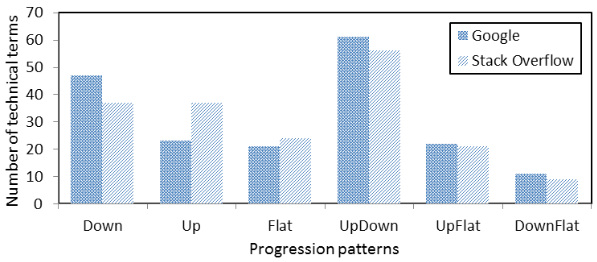


Fig. 4. Distribution of progression patterns in Google and Stack Overflow

and DownFlat. Fig. 3 present examples of these progression patterns. The red line and blue line represent the search trend in Google (GO) and the asking trend in Stack Overflow (SO).

B. Analyzing Progression Patterns

Fig. 4 shows the number of technical terms whose search (or asking) trend exhibits a particular progression pattern. We can see that UpDown is the most common progression pattern in two datasets. UpDown pattern accounts for 61 technical terms in Google and 56 in Stack Overflow. Most of them are specific versions of a programming technique such as *action-script-3* in Fig. 3(d). The popularity of *action-script-3* increases, but then decrease as a newer version *action-script-4* is released. Other technical terms having UpDown pattern are often techniques (e.g., *svn*) whose popularity increases in the past but then decreases due to the rising of competing techniques (e.g., *git*).

Flat, UpFlat and DownFlat patterns together account for 54 technical terms in Google and 54 in Stack Overflow. Technical terms with Flat pattern are often related to common programming concepts, such as regular expression *regex* in Fig. 3(c). Popularity of such common concepts is usually constant over time. UpFlat and DownFlat patterns show that popularity of some technical terms has gone through an increase or decrease stage, and then has become stable, such as *android-intent* in Fig. 3(e) and *command-line* in Fig. 3(f).

Down pattern accounts for 47 technical terms in Google and 37 in Stack Overflow. Technical terms with Down pattern are often with long history and have been losing popularity due to the rising of competing techniques, such as *.net* in Fig. 3(a). Up pattern accounts for 23 technical terms in Google and 37 in Stack Overflow. These technical terms often represent hot programming tools or frameworks that are gaining popularity, such as *hadoop* in Fig. 3(b).

In general, search trend and asking trend of 144 (78%) technical terms exhibit the same progression pattern. 21 technical terms exhibit partially matched progression patterns, for example, *asp.net-mvc* (UpFlat search trend versus DownFlat asking trend), *svn* in Fig. 5(c) (UpDown search trend versus Down asking trend), and *xml-parsing* (Down search trend versus UpDown asking trend). 20 terms exhibit progression patterns with different directions. For example, 13 technical terms exhibit Down pattern in Google, but Flat (e.g., *jdbc* in Fig. 5(b)) and Up patterns (e.g., *javascript* in Fig. 5(a)) in Stack Overflow. As shown in the next section, search trend and asking trend of these 41 technical terms often have low or negative correlations. We will further elaborate the reasons for unmatched progression patterns together with quantitative analysis of trend correlations in the Section V-B.

Summary: Search trends and asking trends exhibit no seasonality and six types of progression patterns. About 1/3 search and asking trends exhibit UpDown pattern, about 1/3 search and asking trends exhibit Flat, UpFlat and DownFlat patterns, and the rest 1/3 search and asking trends exhibit Down or Up patterns. 78% of technical terms exhibit the same progression pattern in Google and in Stack Overflow, 16% of technical terms exhibit partially matched progression patterns, and the rest 16% exhibit different trend directions.

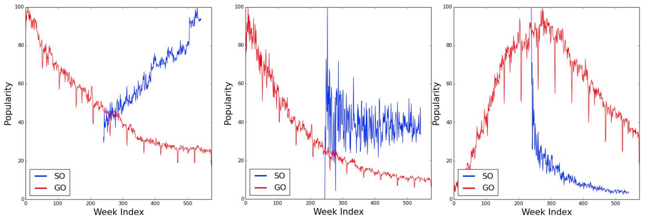
V. Alignment of Search and Asking Trends

To better understand the change of developers' interest in Google and in Stack Overflow, we quantitatively study the correlation and delay of the search and asking trends of the 185 technical terms.

A. Method

The time span of search trend is 574 weeks, while that of asking trend is 300 weeks. We use cross-correlation method proposed in [11] to find the maximum Pearson correlation coefficient between the search trend and asking trend of a technical term (see Algorithm 1). We set the first week of asking trend (i.e., Aug 03, 2008) as 0, and thus the last week of asking trend (i.e., May 03, 2014) is 299, the first week of search trend (i.e., Jan 04, 2004) is -239, and the last week of search trend (i.e., Jan 03, 2015) is 334.

Given the search trend T_s and asking trend T_a of a technical term, the Algorithm 1 moves the asking trend backward in time up to 239 ($0-239$) weeks and forward in time up to 35 ($334-299$) weeks (i.e., w in $-239:35$). For each w, it first gets a segment of 300-weeks (i.e., $T_a.len$) search trend starting at the



(a) javascript (b) jdbc (c) svn

Fig. 5. Examples of unmatched progression patterns

week w, and then compute the Pearson correlation coefficient between the search trend segment T_{s-seg} and the asking trend. The algorithm returns the maximum correlation between the two trends, and the corresponding week w maximizing the correlation as the delay between the two trends. Positive delay value ($1 \leq w \leq 35$) means that asking trend precedes search trend. Negative delay value ($-239 \leq w \leq -1$) means search trend precedes search trend.

```
Input: Search trend $T_s$ and Asking trend $T_a$ of a
       technical term
Output: maxCorrelation, delay
maxCorrelation = $-1$ ;
for $w$ in $-239 : 35$ do
    $T_{s-seg} \leftarrow T_s.getSegment(w, w + T_a.len)$ ;
    $r \leftarrow$ pearson($T_{s-seg}$, $T_a$) ;
    if $corr > maxCorrelation$ then
        maxCorrelation $\leftarrow r$ ;
        delay $\leftarrow w$;
    end
end
```

Algorithm 1: Cross correlation of search and asking trends

B. Correlation of Search and Asking Trends

Fig. 6 depicts the 185 technical terms (dots) in a scatter plot. The vertical axis shows the correlation of the search trend and asking trend of a technical term. The horizontal axis shows the first week in which the technical term appears in Google Trends, i.e., the first week with non-zero search popularity. The smaller the first week is, the older the technical term is.

As Evans [19] suggests that Pearson correlation $r > 0.6$ indicates strong correlation between the two variables, 125 (68%) technical terms have strongly correlated search and asking trend ($p < 0.05$). Many of these 125 technical terms represent specific programming techniques or frameworks, such as *.net*, *hadoop*, *regex*, *action-script-3*, *android-intent*, and *command-line* (see Fig. 3). In addition, 26 technical terms that appear after May 17, 2009 have the strongest correlation $r > 0.9$.

49 technical terms have moderately or weakly correlated search and asking trend (i.e., $0 < r < 0.6$). 38 (78%) of these 49 technical terms appear in the first week (Jan 04, 2004) of the Google Trends data, i.e., those dots at the horizontal tick 0. These technical terms represent programming techniques that already exist before the earliest search statistics Google Trends provides, such as *jdbc*, *gcc* and *xpath* . As such, search trends of these technical terms do not capture their entire life span in Google. This data incompleteness may result in unmatched progression patterns and weaker correlations of search and asking trends. In contrast, except *mvvm* and *django-models*, all other 47 technical terms that appear after Stack Overflow was launched (Aug 03, 2008) have strongly correlated search trend and asking trend (i.e., $r > 0.6$), because both Google Trends and Stack Overflow can provide complete search and asking statistics for these technical terms.

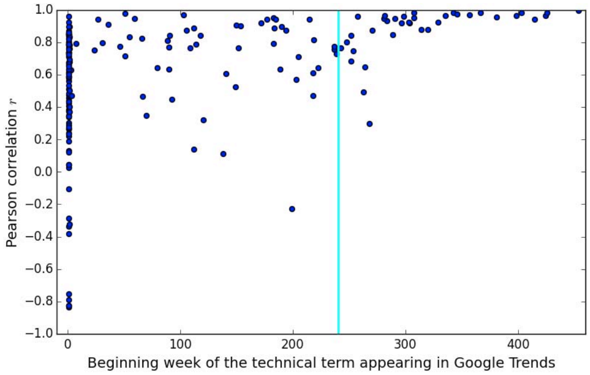


Fig. 6. Correlations of search and asking trends. The green line is the 239th week when Stack Overflow was launched. 125 (68%) technical terms have strongly correlated (i.e., $r > 0.6$) search and asking trends. 26 technical terms appear after May 17, 2009 (the 281th week) have the strongest correlations $r > 0.9$.

11 (6%) technical terms have negatively correlated search trend and asking trend, i.e., $r < 0$. 10 of these 11 technical terms already exist before the earliest search statistics Google Trends provides, including *arraylist*, *html*, *.htaccess*, *iframe*, *javascript*, *mysql*, *php*, *postgresql*, *servelets*, *xml-parsing*. These 10 technical terms exhibit the opposite progression patterns (DOWN versus UP) in Google and Stack Overflow, and thus have negative correlations. Another reason for negative correlations could be the difference of user behavior in Google and Stack Overflow. For example, developers may search a specific javascript technique such *jquery*, *angularjs* without the keyword *javascript*. However, they frequently tag questions with both *javascript* and specific javascript techniques in Stack Overflow. This could explain why *javascript* is gaining asking popularity, but search trend of *javascript* is going down.

C. Delay between Search and Asking Trends

Fig. 7 depicts the 125 (68%) technical terms that have strong correlation (i.e., $r > 0.6$) between search and asking trend. The vertical axis shows the delay (the number of weeks) between the search trend and asking trend of a technical term. Same as Fig. 6, the horizontal axis shows the first week in which the technical term appears in Google Trends.

We consider the search and asking trend of 14 technical terms as synchronous (i.e., within ±1 week). These 14 technical terms are shown as blue dots in Fig 7. 6 of them appear in Google Trends after Stack Overflow was launched. These 6 topics are all new programming frameworks or platforms released in recent years, such as *ember.js*, *symfony2*, and *visual-*

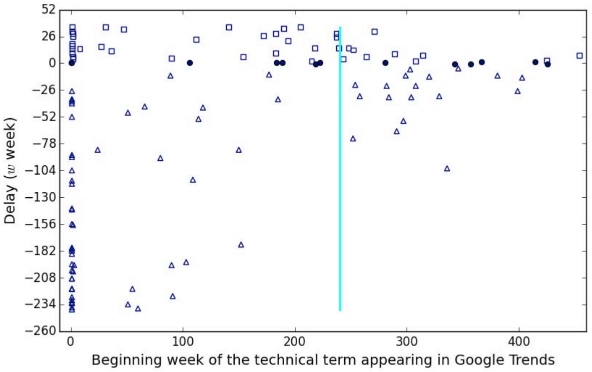


Fig. 7. Delay between search and asking trends. The green line is the 239th week when Stack Overflow was launched. The newer the technical terms, the smaller the delay between their search and asking trends.

studio-2012. The other 5 technical terms like *sql-server-2008*, *numpy* appear within 2.5 years (about 130 weeks) before Stack Overflow was launched. The rest 3 technical terms represent long-history techniques, *dll*, *xml*, and *xcode*. The popularity of these long-history techniques has been decreasing on the same pace in Google and Stack Overflow.

Search trend of 71 technical terms precedes at least 1 week before their asking trend, i.e., $w < -1$. Among these 71 technical terms, 17 terms have delay in between 1 and 34 weeks ($-34 \leq w < -1$) and 14 of these 17 technical terms appear after Stack Overflow was launched. The rest 54 terms have delay more than 34 weeks ($w < -34$). Among the 54 long-delayed technical terms, 50 technical terms appear in Google Trends before Stack Overflow was launched, and 21 appear in Google Trends from the first week of search trend. High trend correlation and long delay of these technical terms suggest that developers' interest in these terms went through similar trajectories in Google and in Stack Overflow. Because Stack Overflow was launched later, asking trend has to be moved backward in time to match search trend.

Asking trend of 40 technical terms precedes at least 1 week before their search trend, i.e., $w > 1$. Among these 40 technical terms, 10 appear after Stack Overflow was launched. 7 of these 10 technical terms have very short delay, i.e., $w < 10$. The rest 30 technical terms appear before Stack Overflow was launched. 21 of these 30 technical terms represent specific programming techniques (some even with version number) with DOWN, UPDOWN and DOWNFLAT progression patterns. Developers' interest in these techniques have been decreasing over time both in Google and in Stack Overflow, as new, competing techniques emerge. It seems that developers' interest decreases faster in Stack Overflow than in Google search. This could be because Google search is used by a much broader set of users, some of which may not be so sensitive to emerging new techniques, for example students.

Summary: 125 (68%) of the 185 technical terms we study have strongly correlated ($r > 0.6$) search trend and asking trend. Two main reasons may cause weak or negative correlations: the incompleteness of the life span captured if the techniques already exist before the launch of Google Trends and Stack Overflow, and the user behavior difference in Google and Stack Overflow. Overall, search trend and asking trend of newer technical terms have stronger correlation and shorter delay, compared with that of older technical terms.

TABLE I
GENERAL AND SPECIFIC TECHNICAL TERMS

General	Specific
.net	.net-4.0
actionscript	actionscript-3
c#	c#-4.0
css	css-3
entity-framework	entity-framework-4
html	html5.
iis	iis-7
jsf	jsf-2
sqlite	sqlite3
asp.net-mvc	asp.net-mvc-2, asp.net-mvc-3, asp.net-mvc-4
python	python-2.7, python-3.x
sql-server	sql-server-2005, sql-server-2008
visual-studio	visual-studio-2008,visual-studio-2010,visual-studio-2012

VI. TRENDS OF GENERAL & SPECIFIC TECHNICAL TERMS

We identify 13 general technical terms (e.g., *.net, html, sql-server*) and their corresponding 19 specific technical terms (e.g., *.net-4.0, html5, sql-server-2005 and sql-server-2008*) in the 185 technical terms (see Table I). We comparatively study the correlations of search and asking trend of these general and specific technical terms.

A. Comparison of Trend Correlations

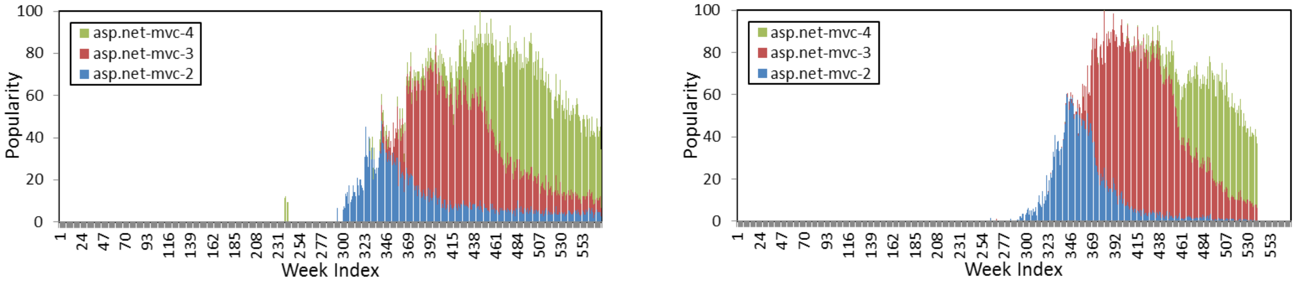
Fig. 9 shows the distribution of search and asking trend correlations of the 13 general technical terms and that of the 19 specific technical terms. Overall, the search and asking trend of specific technical terms have stronger correlations than that of general technical terms.

All the 19 specific technical terms have strongly correlated search and asking trend, i.e., $r > 0.6$. *sqlite3* and *iis-7* have the relative low Pearson correlation 0.6 and 0.64 respectively. Although *sqlite3* is a relative specific term compared with *sqlite*, it has many versions such as sqlite3.0.1, sqlite3.2.8 and sqlite3.8.8 from 2004 till now. Developers searching or asking *sqlite3* may be interested in many different versions. This may result in the relative low correlation of the search and asking trend of *sqlite3*. And *iis-7* has the similar situation.

9 (69%) general technical terms have Pearson correlation $r > 0.6$. *.net* has the highest Pearson correlation $r = 0.92$. As shown in Fig. 3(a), search and asking trend of *.net* have the similar DOWN trajectory. *html* has the negative Pearson correlation $r = -0.8$. As discussed in Section V-B, due to the incomplete life span of search statistics that Google Trends provide for long-history techniques like *html*, these techniques often exhibit unmatched progression patterns and have weakly or negatively correlated search and asking trend.

B. Trends of Related Specific Technical Terms

Four general technical terms (i.e., *asp.net-mvc*, *sql-server*, *python* and *visual-studio*) have two or more specific technical terms. For each general technical terms (like *python*), we aggregate the search trend of its corresponding specifics ones (such as *python-2.7* and *python-3.x*) into an aggregated search



(a) Google (b) Stack Overflow

Fig. 8. Replacement process of asp.net-mvc version 2, 3 and 4 in 574 weeks

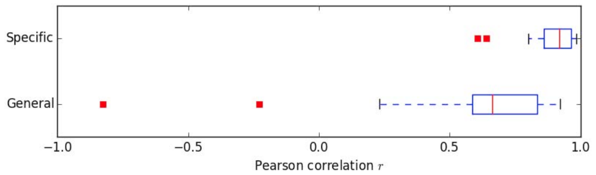


Fig. 9. Distribution of trend correlations of general and specific terms

trend by normalizing the sum of weekly percentage of these specific technical terms into an integer index (0 to 100). Similarly, we obtain an aggregated asking trend of this set of specific technical terms. We find that there is only weak correlation between the aggregated search (or asking) trend and that of the corresponding general technical term. This may be because there are many other versions (*python 2.3* and *python 2.6*) that are not included in our data set.

Fig. 8 visualizes the aggregated search trend and the aggregated asking trend of *asp.net-mvc-2*, *asp.net-mvc-3* and *asp.net-mvc-4* in stack graph for the period of 574 weeks. We can see that the search popularity of *asp.net-mvc-2* and *asp.net-mvc-3* in Google was increasing then decreasing, and the popularity of *asp.net-mvc-4* in Google was increasing and become stable. The asking popularity of *asp.net-mvc-2*, *asp.net-mvc-3* and *asp.net-mvc-4* in Stack Overflow bears the same relations. We also observe the same trend relations for the other three sets of specific technical terms. This phenomenon reveals a replacement process in which the popularity of one specific term has been gradually shifted to the newer related term both in Google and in Stack Overflow.

We use the cross-correlation method (see Section V-A) to find the maximum Pearson correlation coefficient between the aggregated search trend and the aggregated asking trend of a set of specific technical terms. The results shows that the aggregated search trend and the aggregated asking trend are strongly correlated: *asp.net-mvc* set (0.96), *sql-server* set (0.88), *python* set (0.87), and *visual-studio* set (0.95). All the correlations are statistically significant at the 0.05 level.

Summary: Search and asking trend of specific technical terms have stronger correlations than that of the corresponding general technical term. Search and asking trend of several versions of a programming technique reveal the transition of the popularity of the technique between subsequent versions. The aggregated search trend and the aggregated asking trend of a set of specific technical terms are strongly correlated, but they cannot represent the search trend and asking trend of the corresponding general technical term.

VII. Stack Overflow as Sampled Data

Our study shows that the information developers search on Google and the questions developers ask on Stack Overflow have strong correlation. This result provides the evidence that Stack Overflow can be used as an important information source for satisfying developers' information needs in web search. In addition to be yet another information source in the wealth of online programming documents, we believe that Stack Overflow data can be exploited as sampled data to study the temporal property and semantics of developers' information needs and online programming documents to enhance topic-based search queries and search results.

A. *Time-Aware Search*

Developers' search for online information often involves implicit temporal intent, for example, searching for information related to a specific version of a framework which is popular for a particular period of time. On the other hand, online programming documents, such as Stack Overflow questions and answers, are also time-sensitive. For example, questions and answers on "asp.net mvc" framework for the period of November 2010 to October 2012 in Stack Overflow are more likely related to *asp.net-mvc-3*, while those after October 2012 are more likely related to *asp.net-mvc-4* (see Fig. 8).

This suggests that time may play an important role in retrieving and ranking relevant online programming documents. However, existing search engines mainly rely on topic similarity, considering version number only as a topic, but do not exploit temporal information that the queries and online documents imply. Recently, researchers have demonstrated the positive effect of incorporating the temporal property of words into microblog search [20], [21], [22]. We believe that the implicit temporal information of online programming documents should also be considered in conjunction with the topic similarity to derive the final document ranking.

We can analyze search and asking statistics to determine if a technique and its related queries and online documents are time sensitive [23]. For example, the technique with Flat search and asking trend (e.g., *regex* in Fig. 3(c)) would be unlikely time-sensitive. In contrast, the technique with UPDOWN search and asking trend (e.g., *action-script-3* in Fig. 3(d), *asp.net-mvc-2/3/4* in Fig. 8) would likely be time-sensitive. We can build temporal profile of a set of documents returned in response to a time-sensitive query [24], [18]. Different from traditional approaches that consider the relevance of each document in isolation, temporal profile allows the design of time-sensitive ranking algorithms that determine the relevance of a document for a query based on the relevance of other documents with similar content that were published around the same time frame [21].

B. Semantic Search

Developers' web search often involves a complex information seeking process [25], in which they have to browse, filter, and digest many online documents around programming knowledge they need. Search engines consider online programming resources only as web pages and links. There lack of effective knowledge representation and organization to assist developers in retrieving, aggregating, and exploring online documents around the knowledge they need.

Over the recent years, semantic search is gaining momentum with the proliferation of several large-scale knowledge graphs [26], [27], such as DBPedia, YAGO, Google's Knowledge Graph, Microsoft's Satori. A knowledge graph is a graphical knowledge representation that captures the types, properties, and relationships of the entities for a particular domain. Techniques have been developed to derive knowledge graph from structured or unstructured documents [28], [29], [30].

Considering the richness and high-quality of the crowd-sourced knowledge in Stack Overflow, we argue that Stack Overflow is an important source for mining knowledge graph for computer programming. As what developers search on Google and what they ask on Stack Overflow have strong correlation, we envision that the knowledge graph mined from Stack Overflow would allow us to add "semantics" to developers' web search to enhance their search experience.

For example, the knowledge graph may allow us to transform a keyword query "eclipse editor example illegalargumentexception" into a structured query for "*example* `resolve` *EditPart.openEditor* `throwing` *IllegalArgumentException*" over the online documents indexed by the knowledge graph. The knowledge graph may also be exploited to support serendipitous search [31], [32]. For example, based on the knowledge graph, search engine may return a result *d3.js* (a fast growing Javascript visualization library) to the developer who searches for "java visualization". If the developer is not constrained to the Java tools, he will find *d3.js* serendipitous, i.e., unexpected, yet useful. Last but not least, based on the categories and relations of technical terms in the knowledge graph, a multifaceted, vertical search user interface [33], [34] can be developed to assist developers' exploratory search [35], [36] in the complex information seeking process. Exploratory search would be very useful when the developer does not know which keywords to use or when he is not looking for a single answer.

VIII. RELATED WORK

There has been a large amount of research on developer's behavior on Stack Overflow. Allamanis and Sutton [37] use Latent Dirichlet Allocation (LDA) to discover question concepts and types in Stack Overflow. Similarly, Linares-Vásquez et al. [38] exploit LDA to analyze topics related to mobile development. Barua et al. [39] use LDA to mine the topics in Stack Overflow discussions, and study the relationships of the mined topics and the popularity of these topics over time. Parnin et al. [13] show that crowd documentation in Stack Overflow can achieve a high coverage of API elements. Our study complements existing work by providing a new perspective of technical terms developers ask in Stack Overflow in relation to what developers search in Google.

Google Trends data has been widely used in many applications. It is used by Bauckhage et al. [40] to model the temporal dynamics of Internet memes. Google Trends is also adopted to quantify trading behaviour in financial markets [41] and to monitor disease outbreak in real time [42]. Rech [43] uses Google Trends data to analyze media attention, search interests, and relations of software engineering technologies and tools. Trends in social media has also been studied. Zhang and Li [44] analyze the trend in Yahoo! Answers to discover hot topics. Giummolè et al. [9] and Kairam et al. [10] analyze the trending events in Twitter to improve trend-sensitive search. Achananuparp et al. [45] develop a web interface for computing trends of software-related tweets. These works analyze trends in one system, while our study correlates two different information sources, i.e., search trend in Google and asking trend in Stack Overflow.

Adar et al. [11] show that there are correlations between user behaviors on multiple web-based systems using search queries, blog posts and news articles. Xiang and Gretzel [8] investigate the role of Facebook and Youtube in searching online travel information. Kavaler et al. [14] study the relationships between the complexity of API elements, traditional software documentation and Stack Overflow Q&As. Linares-Vásquez et al. [46] investigate what types of API changes trigger more discussions in Stack Overflow. Different from these works, our study focuses on the relationships of the information searched on Google and the questions asked on Stack Overflow.

IX. CONCLUSION AND FUTURE WORK

We have demonstrated that search on Google and asking on Stack Overflow has strong correlation, in terms of technical terms searched and asked as well as their corresponding temporal patterns and trends. Search and asking of newer, specific technical terms have larger content overlap and stronger correlation, compared with older, general programming topics. This suggests that with time going by, as new techniques emerge and replace old ones, the information developers search on Google could become more and more correlated with the questions developers ask on Stack Overflow. We discuss time-aware search and semantic search that can exploit the temporal property and semantics of Stack Overflow data to enhance topic-based search engines.

Acknowledgment

The authors would like to thank the comments of the reviewers. This work was partially supported by MOE AcRF Tier1 Grant M4011165.020.

References

[1] R. S. Taylor, "The process of asking questions," *American documentation*, vol. 13, no. 4, pp. 391–396, 1962.

[2] J. W. Perry, "Defining the query spectrum-the basis for developing and evaluating information-retrieval methods," *Engineering Writing and Speech, IEEE Transactions on*, vol. 6, no. 1, pp. 20–27, 1963.

[3] N. J. Belkin, "Anomalous states of knowledge as a basis for information-retrieval," *Canadian Journal of Information Science-Revue Canadienne Des Sciences De L Information*, vol. 5, no. MAY, pp. 133–143, 1980.

[4] R. N. Oddy, "Information retrieval through man-machine dialogue," *Journal of documentation*, vol. 33, no. 1, pp. 1–14, 1977.

[5] A. Anagnostopoulos, A. Z. Broder, and D. Carmel, "Sampling search-engine results," *World Wide Web*, vol. 9, no. 4, pp. 397–429, 2006.

[6] M. R. Morris, J. Teevan, and K. Panovich, "What do people ask their social networks, and why?: a survey study of status message q&a behavior," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2010, pp. 1739–1748.

[7] R. Gazan, "Social q&a," *Journal of the American Society for Information Science and Technology*, vol. 62, no. 12, pp. 2301–2312, 2011.

[8] Z. Xiang and U. Gretzel, "Role of social media in online travel information search," *Tourism management*, vol. 31, no. 2, pp. 179–188, 2010.

[9] F. Giummolè, S. Orlando, and G. Tolomei, "Trending topics on twitter improve the prediction of google hot queries," in *Social Computing (SocialCom), 2013 International Conference on*. IEEE, 2013, pp. 39–44.

[10] S. R. Kairam, M. R. Morris, J. Teevan, D. J. Liebling, and S. T. Dumais, "Towards supporting search over trending events with social media." in *ICWSM*, 2013.

[11] E. Adar, D. S. Weld, B. N. Bershad, and S. S. Gribble, "Why we search: visualizing and predicting user behavior," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 161–170.

[12] A. Fourney and M. R. Morris, "Enhancing technical q&a forums with citehistory." in *ICWSM*. Citeseer, 2013.

[13] C. Parnin, C. Treude, L. Grammel, and M.-A. Storey, "Crowd documentation: Exploring the coverage and the dynamics of api discussions on stack overflow," *Georgia Institute of Technology, Tech. Rep*, 2012.

[14] D. Kavaler, D. Posnett, C. Gibler, H. Chen, P. Devanbu, and V. Filkov, "Using and asking: Apis used in the android market and asked about in stackoverflow," in *Social Informatics*. Springer, 2013, pp. 405–418.

[15] "Google trend," https://www.google.com.sg/trends/, accessed: 2015-01.

[16] "Stack overflow data dump," https://archive.org/details/stackexchange, accessed: 2014-09.

[17] "Google trend rules of search terms," https://support.google.com/trends/answer/4359582?hl=en, accessed: 2015-01.

[18] A. Kulkarni, J. Teevan, K. M. Svore, and S. T. Dumais, "Understanding temporal query dynamics," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 167–176.

[19] J. D. Evans, *Straightforward statistics for the behavioral sciences*. Brooks/Cole, 1996.

[20] T. Miyanishi, K. Seki, and K. Uehara, "Time-aware latent concept expansion for microblog search," in *Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[21] W. Dakka, L. Gravano, and P. G. Ipeirotis, "Answering general time-sensitive queries," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 2, pp. 220–235, 2012.

[22] J. Lin and M. Efron, "Temporal relevance profiles for tweet search," in *SIGIR Workshop on Time-aware Information Access*. Citeseer, 2013.

[23] D. Metzler, R. Jones, F. Peng, and R. Zhang, "Improving search relevance for implicitly temporal queries," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2009, pp. 700–701.

[24] R. Jones and F. Diaz, "Temporal profiles of queries," *ACM Transactions on Information Systems (TOIS)*, vol. 25, no. 3, p. 14, 2007.

[25] H. Li, Z. Xing, X. Peng, and W. Zhao, "What help do developers seek, when and how?" in *Reverse Engineering (WCRE), 2013 20th Working Conference on*. IEEE, 2013, pp. 142–151.

[26] A. Bordes and E. Gabrilovich, "Constructing and mining web-scale knowledge graphs: Kdd 2014 tutorial," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014.

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

[28] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 601–610.

[29] A. Uyar, F. M. Aliyu, and G. Gorman, "Evaluating search features of google knowledge graph and bing satori: entity types, list searches and query interfaces," *Online Information Review*, vol. 39, no. 2, 2015.

[30] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J.-R. Wen, "Statsnowball: a statistical approach to extracting entity relationships," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 101–110.

[31] 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 Conference on information & knowledge management*. ACM, 2013, pp. 109–118.

[32] E. G. Toms, "Serendipitous information retrieval." in *DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries*. Zurich, 2000.

[33] Z. Nie, J.-R. Wen, and W.-Y. Ma, "Object-level vertical search." in *CIDR*, 2007, pp. 235–246.

[34] P. Papadakos, S. Kopidaki, N. Armenatzoglou, and Y. Tzitzikas, "Exploratory web searching with dynamic taxonomies and results clustering," in *Research and Advanced Technology for Digital Libraries*. Springer, 2009, pp. 106–118.

[35] G. Marchionini, "Exploratory search: from finding to understanding," *Communications of the ACM*, vol. 49, no. 4, pp. 41–46, 2006.

[36] R. W. White and R. A. Roth, "Exploratory search: Beyond the query-response paradigm," *Synthesis Lectures on Information Concepts, Retrieval, and Services*, vol. 1, no. 1, pp. 1–98, 2009.

[37] M. Allamanis and C. Sutton, "Why, when, and what: analyzing stack overflow questions by topic, type, and code," in *Proceedings of the 10th Working Conference on Mining Software Repositories*. IEEE Press, 2013, pp. 53–56.

[38] M. Linares-Vásquez, B. Dit, and D. Poshyvanyk, "An exploratory analysis of mobile development issues using stack overflow," in *Proceedings of the 10th Working Conference on Mining Software Repositories*. IEEE Press, 2013, pp. 93–96.

[39] 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.

[40] C. Bauckhage, K. Kersting, and F. Hadiji, "Mathematical models of fads explain the temporal dynamics of internet memes." in *ICWSM*, 2013.

[41] T. Preis, H. S. Moat, and H. E. Stanley, "Quantifying trading behavior in financial markets using google trends," *Scientific reports*, vol. 3, 2013.

[42] H. A. Carneiro and E. Mylonakis, "Google trends: a web-based tool for real-time surveillance of disease outbreaks," *Clinical infectious diseases*, vol. 49, no. 10, pp. 1557–1564, 2009.

[43] J. Rech, "Discovering trends in software engineering with google trend," *ACM SIGSOFT Software Engineering Notes*, vol. 32, no. 2, pp. 1–2, 2007.

[44] Z. Zhang and Q. Li, "Questionholic: Hot topic discovery and trend analysis in community question answering systems," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6848–6855, 2011.

[45] P. Achananuparp, I. N. Lubis, Y. Tian, D. Lo, and E.-P. Lim, "Observatory of trends in software related microblogs," in *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering*. ACM, 2012, pp. 334–337.

[46] M. Linares-Vásquez, G. Bavota, M. Di Penta, R. Oliveto, and D. Poshyvanyk, "How do api changes trigger stack overflow discussions? a study on the android sdk," in *Proceedings of the 22nd International Conference on Program Comprehension*. ACM, 2014, pp. 83–94.