Metamorphic Testing for Software Quality Assessment: A Study of Search Engines

Zhi Quan Zhou, Shaowen Xiang, and Tsong Yueh Chen

Abstract—Metamorphic testing is a testing technique that can be used to verify the functional correctness of software in the absence of an ideal oracle. This paper extends metamorphic testing into a user-oriented approach to software verification, validation, and quality assessment, and conducts large scale empirical studies with four major web search engines: Google, Bing, Chinese Bing, and Baidu. These search engines are very difficult to test and assess using conventional approaches owing to the lack of an objective and generally recognized oracle. The results are useful for both search engine developers and users, and demonstrate that our approach can effectively alleviate the oracle problem and challenges surrounding a lack of specifications when verifying, validating, and evaluating large and complex software systems.

Index Terms—Software quality, verification, validation, quality assessment, oracle problem, lack of system specification, metamorphic testing, user-oriented testing, search engine

1 Introduction

The goal of software engineering practices is to develop high quality software. It is therefore crucial to develop evaluation methods for various types of software qualities [1]. Testing is a widely used approach for evaluating software qualities and helping developers to find and remove software faults. The majority of software testing techniques assume the availability of an *oracle*, a mechanism against which testers can verify the correctness of the outcomes of test case executions [2]. In some situations, however, an oracle is not available or is available but is too expensive to be used—a situation known as the *oracle problem*, a fundamental challenge for software testing.

A *metamorphic testing* (MT) method has been developed to alleviate the oracle problem [3], [4], [5]. Unlike conventional testing methods, MT does not focus on the verification of each individual output, but instead checks the relationships among the inputs and outputs of *multiple* executions of the program under test. Such relationships are known as *metamorphic relations* (MRs), and are necessary properties of the intended program's functionality: If an MR violation is detected, then a fault is said to be revealed. MT has been used to check the functional correctness of various applications [6], [7], [8], [9], [10], [11], [12], [13], [14] and has also been applied to program proving and debugging [15], [16]. Its effectiveness has also been carefully studied [17], [18].

- *Z.Q. Zhou and S. Xiang are with the School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia. E-mail: zhiquan@uow.edu.au, sx934@uowmail.edu.au.*
- *T.Y. Chen is with the Department of Computer Science and Software Engineering, Swinburne University of Technology, Hawthorn, Victoria 3122, Australia. E-mail: tychen@swin.edu.au.*

Manuscript received 12 Feb. 2015; revised 24 Aug. 2015; accepted 5 Sept. 2015. Date of publication 9 Sept. 2015; date of current version 21 Mar. 2016. Recommended for acceptance by P. Tonella.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TSE.2015.2478001

The present research extends metamorphic testing into a quantifiable approach for software quality assessment, which includes, but is not limited to, the verification and validation of software correctness. We applied our approach to alleviate the oracle problem for the testing and quality assessment of (web) search engines. Search engines are software systems designed to search for information on the World Wide web, and are the main interface through which people discover information on the Internet; web searching is one of the most popular functionalities of the Internet, second only to email [19]. As more and more services and data are being made available on the Internet, search engines are becoming increasingly important. In today's highly competitive search market, it is imperative that search engines provide the desired result according to the queries entered. It is, however, extremely difficult to assess some key qualities of these search engines. For instance, owing to the sheer volume of data on the Internet, it is very difficult to verify or validate the correctness of the software systems or to evaluate the accuracy and completeness of the search results. Also, given the obvious subjectivity of different judges, objective assessment of search result relevance and ranking quality is very difficult.

This paper addresses the above problems using MT, and consequentially demonstrates new dimensions of the usefulness of MT. A series of empirical studies have been conducted to compare the software qualities of four major search engines, namely, Google (www.google.com), Bing (www.bing.com), Chinese Bing (Bing for Chinese users, www.bing.com.cn), and Baidu (www.baidu.com). According to NetMarketshare [20], Google, Baidu, and Bing are the three most popular search engines in the world.

The rest of this paper is organized as follows: Section 2 provides background information, introduces the basic strategies of our approach, and summarizes our contributions. Section 3 describes the MRs we identified for this study. The validity of these MRs and the design of the experiments are presented in Section 4. Section 5 analyzes

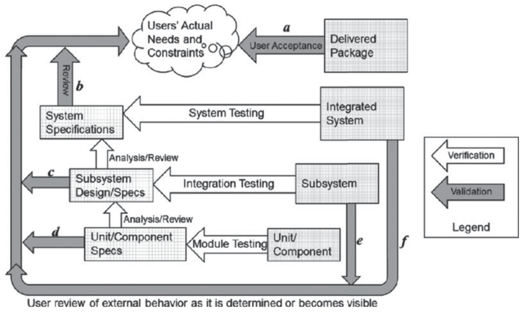


Fig. 1. Conventional software verification and validation activities (adapted from Pezzè and Young [21]).

the empirical study results. Section 6 further discusses several important software engineering issues and puts this work in context by examining some fundamental software quality models used in quality control. Section 7 summarizes the paper and provides future research directions.

2 Preliminaries

In this section, the background information and basic concepts of our approach are introduced and explained. A summary of the contributions of this research is also presented.

2.1 Search Engine Software Characteristics and User Validation Difficulties

Software testing can be performed for different purposes, with verification and validation being the core tasks. Fig. 1 sketches typical verification and validation activities in conventional software development projects [21]. The white arrows represent verification activities, which check the consistency between two descriptions (that is, the consistency of an implementation with its specification). The shaded arrows (a, b, c, d, e, and f) represent validation activities, where the software systems, designs, and specifications are tested or reviewed by actual users to assess the degree to which they meet the users' real needs. Pezzè and Young [21] further pointed out that validation activities refer primarily to the overall system specification and the final code.

Software validation for search engines is much harder than the conventional validation activities depicted in Fig. 1. This is because the vast majority of search engine users do not have access to the intermediate products (namely, system/subsystem/component specifications, designs, and implementations). In other words, the validation activities b, c, d, e, and f shown in Fig. 1 are missing—but these activities are important for the users to understand and validate the final delivered package. For search engines, therefore, software validation is limited to activity a only, where the users can only use an online search service without knowledge about its design. For users, search engines are representative of a large body of software products that come without specifications (or with incomplete specifications) [22]. The lack of knowledge about the software design and specifications details, coupled with the oracle problem discussed in Section 1, makes user validation extremely difficult, if not impossible: It is hard for the users to decide if, and to what degree, the search engines are appropriate for their specific information needs.

Search engine developers usually do not publicize detailed specifications of their systems for two main reasons: First, the design and algorithms are commercial secrets and, therefore, should not be released to the general public; and second, because of the complexity of the software systems. Owing to the unprecedented scale of the web, the unique characteristics of the web documents and web users, as well as the real-time requirements, the design of the software systems (and subsystems and modules) of search engines entails many practical considerations and tradeoffs, such as between quality and speed; quality and quantity; response time and storage; and precision and flexibility. The developer has to consider these things when designing software systems and modules, and often has to choose an algorithm/design from multiple candidates, each of which has its advantages and limitations. The developer's choice might be validated at the individual module's level, but when the modules are integrated to form systems, the interplay of the different factors involved in the considerations and tradeoffs can become so complex that it can be impossible for the developer to clearly explain to the end users. In fact, through our interactions with colleagues in industry, we found that it can often be hard even for the developers themselves to explain or to verify certain search engine behavior.

Without access to a system specification, the online help pages (referred to as *online specifications* hereafter) are the only type of specification available to the users. However, these are usually very brief, and only explain how to use the basic search operators, without actually revealing any technical design detail of the software systems. They are in no way equivalent to the "system specification" shown in Fig. 1, defined as "an adequate guide to building a product that will fulfill its goals" [21]. For instance, Bing online specification states "We design algorithms to provide the most relevant and useful results"—how the relevance or usefulness is determined is not specified. This lack of technical detail makes it difficult for users to validate the search engine for their specific needs as different users can have very different criteria of relevance and usefulness of results.

2.2 Our Approach: User-Oriented Testing

In this paper, we present a testing approach that alleviates the difficulties in search engine verification, validation, and quality assessment. Our approach is based on the following observation: A search is always performed in the context of a particular scenario, and involves certain specific functions, which are only a very small set of all functions offered by the search engine. Therefore, *the user does not need to understand the system in its entirety in order to validate the search engine*; instead, he/she only needs a testing technique that tells him/her whether or not the few functions directly involved in the search can deliver what he/she wants. When the test fails, it can either indicate a fault in the implemented software system or a deficiency in the algorithm(s) chosen by the search engine developer—for validation purposes, the user does not need to distinguish between these two cases.

Based on the above observation, we propose a user-oriented testing approach using the concept of MT. Our approach demonstrates the feasibility of MT being a unified framework for software verification, validation, and quality assessment. It is user-oriented because we propose to make use of MRs defined from the users' perspective: The users can define MRs as necessary properties of what they would expect a "good" search engine to have, to meet their specific needs, irrespective of how the search engine was designed. In other words, the MRs can be designed based on the users' expectations to reflect what they really care about, not based on the algorithms/designs chosen by the developer (which are unknown to the users anyway)—note that this view is different from that of conventional MT, where MRs are identified based on the target algorithm to be implemented [3], [4], [5]. From this perspective, our approach performs software validation.

In addition to validation, our approach can also be used to do verification—because the MRs can be designed not only based on the user's expectation, but also based on the online specification shown to the user (the search engine's online help pages). Testing the search engine against such an MR is therefore a check of consistency between the online specification and the implementation and, hence, is verification.

Our approach can also be used for software quality assessment beyond verification and validation. This is because the software qualities that we consider include, but are not limited to, functional correctness.

Our user-oriented testing approach, however, is different from previous work on end-user software engineering [23], [24], where the user, tester, and developer are the same person (known as the "end-user programmer"), such as a secretary or a scientist writing some programs to support their own work. In this situation, the user knows the details of the algorithms or formulas to be implemented and, hence, can identify MRs based on these algorithms or formulas to verify the implementation. The user also has access to the implemented source code. In the context of the present research, however, the user has no knowledge about the designs/algorithms/formulas of the software at all, and the software is treated as a sheer black box.

2.3 Metamorphic Testing (MT)

MT [4], [5], [18], [25] alleviates the oracle problem by testing against MRs, which are necessary properties of the software under test (SUT). MRs differ from other types of correctness properties in that an MR is a relationship among *multiple* executions of the target program. As a result, even if no oracle is available to verify each individual output, we can still check the multiple outputs of the SUT against the given MR. If the MR is violated for certain test cases, a failure is revealed. Consider, for instance, a program $p(G,\ a,\ b)$ purportedly calculating the length of the shortest path between nodes a and b in an undirected graph G. When G is nontrivial, the program is difficult to test because no oracle can be practically applied. Nevertheless, we can perform MT. Let $(G_1,\ a_1,\ b_1)$ and $(G_2,\ a_2,\ b_2)$ be two inputs, where G_2 is a permutation of G_1 (that is, G_2 and G_1 are isomorphic), and (a_2, b_2) in G_2 correspond to (a_1, b_1) in G_1. Then an MR can be identified as follows: $p(G_1,\ a_1,\ b_1) = p(G_2,\ a_2,\ b_2)$. A metamorphic test using this MR will run p twice, namely, a *source execution* on the *source test case* $(G_1,\ a_1,\ b_1)$ and a *follow-up execution* on the *follow-up test case* $(G_2,\ a_2,\ b_2)$. Many other MRs can also be identified, such as $p(G,\ a,\ b) = p(G,\ b,\ a)$, and so on [26].

MT was originally proposed as a *verification* technique, where an MR is a necessary property of the algorithm to be implemented. Therefore, a violation of the MR reveals a fault in the implementation. When studying MT for machine learning software, Xie et al. [8] coincidentally discovered that an MR can be a necessary property of the target software rather than the selected algorithm—in this case, even if the selected algorithm has been correctly implemented, the MR can still be violated if the algorithm has some deficiency and cannot meet the user's expectation for the target software. In this situation, MT is used for *validation*. Nevertheless, the study conducted by Xie et al. was small scale, and at the algorithm selection level (checking whether the adopted algorithm was appropriate or not). In our research, we used MT for software validation at the top level (the system/service level) and conducted very large scale empirical studies. Further, we extended MT into a *quality assessment* method.

Using MT to test search engines was first proposed by Zhou et al. [27], [28], where some logical consistency relationships among multiple responses of the search engines were employed. For instance, such a relationship, an MR, is that the number of web pages satisfying c_1 should be less than or equal to the number of web pages satisfying c_1 *or* c_2, where c_1 and c_2 are two search criteria. Zhou et al. conducted pilot studies to test the functional correctness of keyword-based search of Google, Yahoo! and Live Search [27], [28]. Imielinski and Signorini [29] essentially also used the logical consistency relationships to test "how semantic" a search engine is. They employed a (metamorphic) relation that a "truly semantic search engine" should return the same results for two semantically equivalent queries. For instance, the queries "biography of George Bush" and "find me bio of George Bush" should return the same results when submitted to a semantic search engine. It is to be noted that Zhou et al. [27], [28] only considered keyword-based search with a focus on the functional correctness, whereas Imielinski and Signorini [29] only considered semantic search with a focus on how the search engines could imitate the behavior of a human operator.

More recently, MT techniques have also been implicitly used to detect search engine censorship [30], [31], [32]. In these studies, the tester first identified some query terms politically sensitive to the Chinese government. These query terms were then sent to the search engine under test multiple times using different languages, such as English, complex Chinese, and simplified Chinese characters. The search results for the different languages were then compared. The US-based search engine www.bing.com was found to return very different results when the queries were in simplified Chinese characters. For instance, it was reported that "If you search a term on Bing that is politically sensitive in China, in English the results are legitimate. ... Conduct the search in complex Chinese characters (the kind used in Taiwan and Hong Kong) and on the whole you still get authentic results. But conduct the search with the simplified characters used in mainland China, then you get sanitized pro-Communist results. ... this is true wherever in the world the search is conducted—including in my office in New York" [30]. Discussion of the validity of these studies is beyond the scope of this paper, but it is worth noting that Microsoft has acknowledged the presence of "error" and "bug" in Bing that caused some of these problems [30], [31], [32], [33]. The above testing method is essentially MT where the source test case is a query that contains a politically sensitive word or phrase in English (such as "Dalai Lama"), and the follow-up test case is another query having the same meaning but typed in Chinese characters. The search engine's responses for the source and follow-up test cases are then compared against predefined logical consistency relationships.

2.4 Contributions

The first contribution of the present work is the development of an approach to software verification, validation, and quality assessment to address the oracle problem and lack of specification problem. For this purpose, we made MT a quantifiable and user-oriented testing approach.

The second contribution is the application of our approach to search engine testing and quality assessment. In this study, we considered both keyword-based and semantic search features, and identified a number of useful MRs and corresponding quality metrics. We further identified different usage patterns or operational profiles.

The third contribution is the results of our large scale empirical studies, where the software qualities of four major search engines were investigated and compared under different operational profiles. Our results not only revealed search engine qualities but also provided hints that may help users to select a suitable search engine for their specific circumstances. We anticipate that the findings of this research will enable both developers and users to gain a deeper understanding of their search engines and, hence, improve the quality of searches.

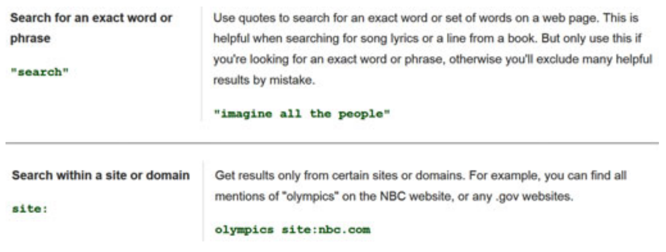
Although this study focuses on search engines, the approach can be applied to a large number of software products where there is an oracle problem and/or where there is no complete system specification (lack of specification can often be a result of poor software evolution [22] or open source software development [34]).

3 A Description of the Identified MRs

To apply MT to the automatic quality assessment of search engines, without the need for an oracle or human assessor, two groups of MRs were used: The "No Missing web Page" group assesses the search engines' capability in retrieving appropriate web pages to meet the users' needs; and the "Consistent Ranking" group assesses the ranking quality of the search results. This section provides a brief description of these MRs. Their validity and the experimental design will be discussed in Section 4.

3.1 Metamorphic Relation: MPSite

MPSite belongs to the "No Missing web Page" group of MRs, which assess the search engine's web page retrieval capability. MPSite is focused on the search engine's reliability when retrieving web pages that contain an exact word or phrase. It therefore assesses the keyword-based search feature. MPSite is described as follows:



Search for an exact word or phrase `"search"`	Use quotes to search for an exact word or set of words on a web page. This is helpful when searching for song lyrics or a line from a book. But only use this if you're looking for an exact word or phrase, otherwise you'll exclude many helpful results by mistake. `"imagine all the people"`
Search within a site or domain `site:`	Get results only from certain sites or domains. For example, you can find all mentions of "olympics" on the NBC website, or any .gov websites. `olympics site:nbc.com`

Fig. 2. Excerpts from Google help page.

Let A be a *source query* for which the search engine returns a non-empty list of results (called the *source response*), namely, $(p_1, p_2, \ldots, p_n)$, where $0 < n$ and p_i is a web page from domain d_i, $1 \le i \le n$. To enhance accuracy and validity of our approach, in MPSite we only consider situations where $0 < n \le 20$ so that we can avoid the inaccuracy associated with large result sets (such as a large list being truncated by the search engine to improve response time).

For the source response $(p_1, p_2, \ldots, p_n)$, n *follow-up queries* are constructed as follows: The ith follow-up query B_i $(1 \le i \le n)$ is constructed in such a way that B_i is identical to A except that B_i includes an additional criterion which requires that all results be retrieved from domain d_i. Let FR_i (a *follow-up response*) be the list of web pages returned by the search engine for query B_i. The metamorphic relation MPSite requires that $p_i \in FR_i$ (note that there is no requirement on the ranking of p_i in FR_i). For example, let us test Google by issuing the following source query:

"`side effect of antibiotics in babies`"

where the quotation marks are part of the query. Google returned a total of seven web pages. Without loss of generality, let us consider the top result, which is:

`http://www.dailymail.co.uk/health/article-2344802/Babies-given-antibiotics-theyre-prone-eczema-Drugs-increase-risk-40.html`

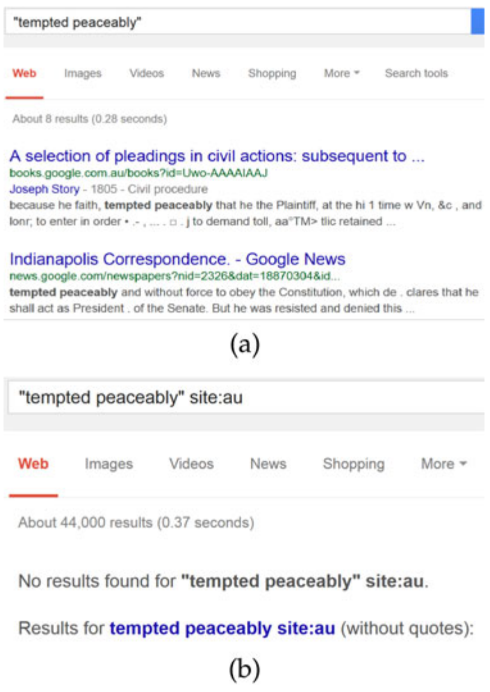
This web page is from the .uk domain.[1] The metamorphic relation MPSite enables the construction of the following follow-up query: ["`side effect of antibiotics in babies`" `site:uk`],[2] where "site:" is a Google search operator that specifies domains (see Fig. 2 (lower)). Obviously, the previously returned top result (http://www.dailymail.co.uk/...) meets this search criterion, is indexed in Google database, and therefore should still be returned by Google for this follow-up query. In this example, Google returned a total of seven web pages for the source query. Therefore, seven follow-up queries are constructed by referring to MPSite.[3]

Using MPSite, even if the assessor is unable to verify or evaluate each individual response, he/she can still

1. To be precise, the domain name of this URL is www.dailymail.co.uk, the Internet protocol is HTTP, and ".uk" is the domain suffix name. For ease of presentation, we simply say the web page "is from the .uk domain" in this paper.

2. A note on the square brackets convention: In a specification, a query can be surrounded by square brackets. When doing the query, the square brackets are not to be typed.

3. If two or more web pages are from the same domain (e.g. from the .com domain), then their follow-up queries will be the same.



"tempted peaceably"

About 8 results (0.28 seconds)

A selection of pleadings in civil actions: subsequent to ...
books.google.com.au/books?id=Uwo-AAAAIAAJ
Joseph Story - 1805 - Civil procedure
because he faith, **tempted peaceably** that he the Plaintiff, at the hi 1 time w Vn, &c , and lonr; to enter in order • .- , ... □ . j to demand toll, aa°TM> tlic retained ...

Indianapolis Correspondence. - Google News
news.google.com/newspapers?nid=2326&dat=18870304&id...
tempted peaceably and without force to obey the Constitution, which de . clares that he shall act as President . of the Senate. But he was resisted and denied this ...

"tempted peaceably" site:au

About 44,000 results (0.37 seconds)

No results found for **"tempted peaceably" site:au**.

Results for **tempted peaceably site:au** (without quotes):

Fig. 3. A Google failure detected using MPSite. The top result in (a) cannot be retrieved in (b).

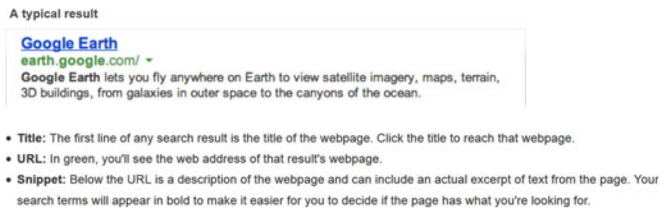
verify the logical consistency relationship among multiple responses. Here, the basic approach is to use the search engine's source response to check its follow-up response. Fig. 3 shows a failure detected using MPSite, which will be further discussed in Section 4.1.1.

All MRs identified in this paper were implemented into a testing tool and, hence, the testing and assessment process is automated.

3.2 Metamorphic Relation: MPTitle

For many search engines including those investigated in the present paper, if the words are not enclosed by double quotation marks, synonyms will be employed automatically. For instance, Google specifies that "Google employs synonyms automatically, so that it finds pages that mention, for example, childcare for the query [child care] (with a space), or California history for the query [ca history]." Synonyms are employed because the search engines attempt to return web pages that best meet users' information needs. In other words, the search engines attempt to imitate the behavior of a human operator, to which end, correct understanding of the web pages and of the user intent are key.

To test a search engine's information retrieval capability in situations where synonyms may be used for semantic search, a good strategy is to construct a test query q that best characterizes a target web page p (the words in q may or may not directly appear in p). Furthermore, p must have been indexed in the search engine's database. The search engine can be tested on q. If p is not retrieved, then the user's perception of the search quality will be poor. A research question is: *How can such q's and p's be identified in a fully automatic fashion for search engine assessment?* The metamorphic relation MPTitle is designed to meet this challenge.



A typical result

Google Earth
earth.google.com/
Google Earth lets you fly anywhere on Earth to view satellite imagery, maps, terrain, 3D buildings, from galaxies in outer space to the canyons of the ocean.

- **Title:** The first line of any search result is the title of the webpage. Click the title to reach that webpage.
- **URL:** In green, you'll see the web address of that result's webpage.
- **Snippet:** Below the URL is a description of the webpage and can include an actual excerpt of text from the page. Your search terms will appear in bold to make it easier for you to decide if the page has what you're looking for.

Fig. 4. Google's definition of page title in search results.

3.2.1 *Description of MPTitle*

MPTitle is another "No Missing web Page" group MR. Although the MPSite MR always uses quotation marks for exact searching, MPTitle involves both exact and inexact query terms. It focuses on the search engine's comprehensive capability of understanding and abstracting web pages and on understanding user intent.

The source query A is defined and constructed in exactly the same way as that of MPSite, for which the search engine returns a small list of web pages, namely, $(p_1, p_2, \ldots, p_n)$, where $0 < n \leq 20$. Let t_i be the title generated by the search engine for result p_i, $1 \leq i \leq n$. The "title" is the first line of any search result, on which users can click to reach the related web page. Fig. 4 shows Google's definition for title; the other search engines under study have similar definitions. According to Google's online specification, the generation of a title "takes into account both the content of a page as well as references to it that appear on the web. The goal of the snippet and title is to best represent and describe each result and explain how it relates to the user's query."

MPTitle requires that a follow-up query B_i, $1 \leq i \leq n$, be a conjunction of two parts: The first part is identical to A (where quotation marks are applied). The second part is the title t_i extracted from the search engine's results screen (also called *search engine results page* or SERP), with all punctuation and symbols removed to avoid ambiguity. This second part is not enclosed by double quotes so that search with synonyms is enabled. There are n follow-up queries, namely, B_1, B_2, $\ldots$, B_n. Let FR_i be the list of web pages returned by the search engine for query B_i. MPTitle requires that the web page $p_i \in FR_i$.

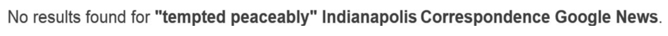
For instance, consider Fig. 3a. The source query "`tempted peaceably`" for MPSite can also serve as a source query for MPTitle. Based on the SERP, a follow-up query for MPTitle can be constructed as follows:

[“`tempted peaceably`” `A selection of pleadings in civil actions subsequent to`]

where "`tempted peaceably`" (with double quotes) is directly copied from the source query, and the rest of the query (without double quotes) is taken from the first result title (with punctuation and symbols removed). According to the online specifications, spaces among words mean "AND" rather than "OR." As the title is generated automatically by the search engine to best present the target web page, such a follow-up query meets the challenge raised earlier, namely, "how can such q's and p's be identified in a fully automatic fashion for search engine assessment?" Google did successfully return the target web page (books.google.com.au/. . .) in response to the follow-up query.

3.2.2 *An Example of Failure Detected Using MPTitle*

Based on the second result shown in Fig. 3a, another follow-up query can be constructed as [“`tempted peaceably`” `Indianapolis Correspondence Google News`]. For this query, however, Google did not find any result, as shown in the screenshot below:


No results found for **"tempted peaceably" Indianapolis Correspondence Google News**.

The above inconsistency was repeatable. Refer to the target web page (news.google.com/newspapers?. . .), which shows a newspaper article (Fig. 5). The text of this web page included the phrases “Indianapolis Correspondence” and “tempted peaceably,” and the headline (title) of this web page included “Google News.” According to Google specification, Google searches “anywhere in the document (title or not).”[4] Therefore, the above inconsistency is confirmed to be a failure.

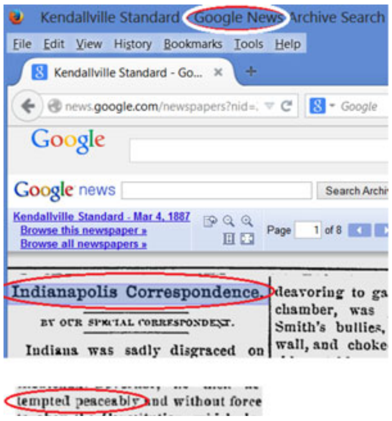

Kendallville Standard Google News Archive Search
File Edit View History Bookmarks Tools Help
Kendallville Standard - Go...
news.google.com/newspapers?nid=
Google
Google news
Search Archi
Kendallville Standard - Mar 4, 1887
Browse this newspaper »
Browse all newspapers »
Page 1 of 8
Indianapolis Correspondence.
BY OUR SPECIAL CORRESPONDENT.
Indiana was sadly disgraced on
deavoring to ga
chamber, was
Smith's bullies,
wall, and choke
tempted peaceably and without force

Fig. 5. Excerpts from the target web page: Google successfully retrieved this web page for the query [“`tempted peaceably`”] but failed to retrieve it for the query [“`tempted peaceably`” `Indianapolis Correspondence Google News`].

3.3 Metamorphic Relation: MPReverseJD

The third MR of the “No Missing web Page” group is MPReverseJD. Its design was inspired by a search engine assessment technique informally used in industry, which is based on the rationale that a good search engine should return similar results for similar queries. For instance, although a search for [`today’s movies in Redmond`] and a search for [`Redmond movies today`] (without double quotes) may return different results, the two result sets should share a large intersection if the search engine is robust to the nonessential differences between these two queries.[5] This idea was also employed by Imielinski and Signorini to test semantic search engines using semantically equivalent queries [29].

The MR MPReverseJD is designed as follows: The source query A is defined to be a query for which the search engine returns a non-empty list of up to 20 results. A is further defined to be the conjunction of up to four terms, namely: $A = A_1 \text{ AND } A_2 \underbrace{\text{AND } A_3}_{optional} \underbrace{\text{AND } A_4}_{optional}$ where A_i $(i = 1, 2, 3, 4)$ is a *name* enclosed by double quotation marks. Terms A_3 and A_4 are optional: A_3 is applied only when the conjunction of A_1 and A_2 has more than 20 results, and A_4 is applied only when the conjunction of A_1, A_2, and A_3 has more than 20 results. If the conjunction of all four terms still has more than 20 results, all these terms will be discarded and a new query will be formed. The following is an example of the source query A: [“`Vincent Van Gogh`” `AND` “`Elvis Presley`” `AND` “`Albert Einstein`” `AND` “`Plato`”]. In this example, A_1 =“`Vincent Van Gogh`,” A_2 =“`Elvis Presley`,” A_3 =“`Albert Einstein`,” and A_4 =“`Plato`.” The follow-up query B is constructed by reversing the order of A’s terms: [“`Plato`” `AND` “`Albert Einstein`” `AND` “`Elvis Presley`” `AND` “`Vincent Van Gogh`.”]

MPReverseJD states that a stable search engine should return similar results for the source query A and follow-up query B. In other words, the two result sets should have a large intersection—we refer to this kind of quality as *stability*. This requirement is reasonable especially given that the result set of A is very small (containing no more than 20 results) and that the source and follow-up queries have similar semantic meanings—this is because the queries only consist of names whose orders do not change the meaning of the queries.

To measure the similarity of the two result sets, we use the metric *Jaccard similarity coefficient* (or Jaccard coefficient for short), defined as $|X|/|Y|$, where X = source response ∩ follow-up response and Y = source response ∪ follow-up response. The source and follow-up responses refer to the source and follow-up queries’ result sets, respectively. Obviously, $0 \leq$ Jaccard coefficient ≤ 1. A larger Jaccard coefficient indicates higher similarity and, hence, better stability. Given that the vast majority of users would prefer stable search results, poor stability may result in a poor user experience. (In this paper, “user experience” refers to users’ perceived quality of the search results.)

3.4 Metamorphic Relation: SwapJD

The second group of MRs is named “Consistent Ranking.” Its first MR is SwapJD, which assesses the search engines’ ranking stability based on the concept that a stable search engine should return similar results for similar queries. SwapJD is described as follows: The source query A contains only two words (without quotation marks) and the follow-up query B is constructed by swapping the two words. A stable search engine should return similar results for A and B if these two queries have similar meanings, regardless of their word orders. The similarity can be measured by calculating the Jaccard coefficient of the top x results in the two result lists, where x can be given by the assessor. In this research, we set x to 50, as our experience suggests that most users are unlikely to browse search results beyond the top 50.[6]

3.5 Metamorphic Relation: Top1Absent

The Top1Absent MR focuses on the ranking quality of the very first result presented in the search results screen.

4. Here, the word “title” refers to the web page’s own headline, not the page title generated by the search engine and listed in the SERP.

5. The first author wishes to thank Microsoft Research, Redmond, for telling him about this practice.

6. According to Imielinski and Signorini [29], more than 65 percent of the search clicks are done on the top (first) result. We also surveyed 31 final year undergraduate students and found that, in most situations, they would not browse beyond the top 50 results.

TABLE 1
Metamorphic Relations and Experimental Design

Group	Name	Usage pattern	Result of each single metamorphic test	Result of each batch of tests	Frequency of test	Do different batches use the same test suite?	Goal
No Missing Web Page	MPSite	English	{pass, failure}	hourly ROCOF. in [0.0, 1.0].	1 batch per hour	No	To test the search engine's Web page retrieval capability, focusing on its reliability of retrieving Web pages that contain an exact word or phrase. (Double quotes are used.)
		Chinese					
	MPTitle	English	{found, not found}	hourly ROCOA. in [0.0, 1.0].	1 batch per hour	No	To test the search engine's Web page retrieval capability, focusing on its capability of abstracting Web pages and understanding user intent. (Double quotes are partially used.)
		Chinese					
	MPReverseJD	People	Jaccard coefficient. in [0.0, 1.0].	hourly avg. Jaccard coefficient. in [0.0, 1.0].	1 batch per hour	No	To test the search engine's Web page retrieval capability, focusing on its stability in terms of returning similar results for similar queries. (Double quotes are used.)
		Companies					
		Drugs					
Consistent Ranking	SwapJD	Universal	Jaccard coefficient. in [0.0, 1.0].	hourly avg. Jaccard coefficient. in [0.0, 1.0].	1 batch per hour	Yes	To test the ranking quality of the search results, focusing on the search engine's stability in terms of returning similar results for similar queries. (Double quotes are not used.)
		site:com					
		site:edu					
		site:mil					
		site:lc					
	Top1Absent	Random English words	{found, not found}	hourly ROCOA. in [0.0, 1.0].	1 batch per hour	Same source queries	To test the ranking quality of the search results, focusing on the top result. (Double quotes are used.)

This top result can be considered as the most important one among all search results. According to Imielinski and Signorini [29], more than 65 percent of search clicks are done on the first result. Top1Absent is designed by extending the idea of MPSite, as described below:

The source query A is a word randomly selected from an English dictionary (excluding common words such as "is" and "of") and is surrounded by double quotes. Let p_1 be the top result, that is, p_1 is the first listed web page returned by the search engine for query A. The follow-up query B still uses A as the query term, but restricts the search to p_1's domain only. The expected relationship is that p_1 should still appear in the search results for B.

4 MR Validity and Experiment Design

This section discusses the validity of the MRs identified in Section 3. For each MR, we also identify some basic usage patterns and evaluation metrics, and explain the design of the experiments, which are summarized in Table 1 and will be explained in the following.

4.1 MPSite

We next discuss the validity of MPSite, and then design the experiments.

4.1.1 Validity of MPSite

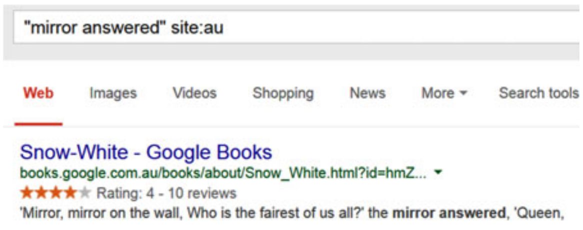
Search engines work differently from conventional types of database systems. It is therefore necessary to discuss the validity of MRs identified for search engines. While this section discusses the validity of MPSite, many of the arguments are applicable to the other MRs identified in the present research. We consider the following characteristics of the Internet and search engines:

1) The Internet is dynamic and search engine databases can be updated in real time, which may lead to inconsistencies between the source and follow-up responses. To address this issue, the following strategy is used: When a violation (inconsistency) of an MR is detected, we always immediately resend the same pair of source and follow-up queries. Only when the violation is repeatable will it be recorded.
2) Search engines are different from conventional database applications in that they often return approximate rather than exact results. For instance, results may be truncated to improve response speed, especially when the system's load is heavy or the result list is large. We address this issue using the following strategies: (i) repeat the violation-causing tests as just described in (1); (ii) conduct exact word/phrase

search (using double quotation marks) to avoid ambiguity; and (iii) avoid large result lists—in our experiments, the source response never contains more than 20 results. In situations where the initial attempt returns more than 20 results, additional words are generated and appended into the double quotation marks of the query until the number of returned web pages is less than or equal to 20. For example, consider a query term "`tempted`," for which Google returned 57,000,000 results. This result count is too large and, therefore, an additional word *peaceably* was selected to change the query term from "`tempted`" into "`tempted peaceably`." As shown in Fig. 3a, Google returned "About eight results" for "`tempted peaceably`." Since $8 < 20$, the query term "`tempted peaceably`" serves as a valid source query, and the initially attempted query term (for which the result count was 57,000,000) is discarded. In situations where the search returned zero results, the query was discarded, and a new query generated. Thus, we avoid the inaccuracy and other problems associated with large result sets.

3) In this research, all queries were constructed using correct syntax as per the online specifications of the search engines under test (for instance, see Fig. 2).
4) Because search engines may truncate queries which are too long, or contain too many words, in all our experiments, we ensured that the query length was well within limits.
5) Because search engines filter results, which may cause inaccuracies and inconsistency, we disabled all filters in our experiments. Furthermore, all cookies, history, cache, active logins, and so on, were also cleared, to avoid personalized results. Advertisements were not counted either. Search engines sometimes display a message such as: "In order to show you the most relevant results, we have omitted some entries very similar to the xxx already displayed. If you like, you can repeat the search with the omitted results included." Whenever this happened during the experiments, the option "repeat the search with the omitted results included" was always selected, which effectively disabled a filter.

Fig. 3 shows an example of Google search failure detected by our automated testing tool using MPSite. The source query is ["`tempted peaceably`"] (Fig. 3a). According to Google online specification (Fig. 2), quotation marks are used to search for an exact phrase. This "is helpful when searching for song lyrics or a line from a book." Google returned eight results, the top one from books.google.com.au (see Fig. 3a). Using MPSite, our testing tool immediately constructed a follow-up query ["`tempted peaceably`" `site:au`] and sent it to Google again. According to MPSite, the web page (books.google.com.au/...) must still be retrieved. However, this time Google reported "No results found" (see Fig. 3b. Note that the message "About 44,000 results" is not relevant as it is for a suggested search without quotes.) The source and follow-up queries were issued consecutively within a very short period of time from the same PC (this practice applied to all experiments reported in this paper).


"mirror answered" site:au

Web Images Videos Shopping News More Search tools

Snow-White - Google Books
books.google.com.au/books/about/Snow_White.html?id=hmZ...
★★★★ Rating: 4 - 10 reviews
'Mirror, mirror on the wall, Who is the fairest of us all?' the **mirror answered**, 'Queen,

Fig. 6. Excerpts from a successful Google SERP.

We also confirmed that the "missing web page" (books.google.com.au/books?id=Uwo...) was indeed valid and active, and contained the exact phrase "tempted peaceably." The detected inconsistency was repeatable, and was still present when this paper was being written. Furthermore, Google online specification for the "site" operator states that "You can specify a domain with or without a period, e.g., either as .gov or gov." Therefore, in addition to "site:au" we also repeated the test using the syntax "site:.au" and the same failure was detected.

Is it possible that the above inconsistency was caused because Google somehow considers books.google.com.au to be a website not belonging to the .au domain? To answer this question, we conducted a search using another phrase: ["`mirror answered`" `site:au`]. As shown in Fig. 6, the website books.google.com.au was successfully returned by Google (ranked fourth; results 1 to 3 are omitted in the figure).

We would like to ask a further question about the cause of the inconsistency: Is it possible that Google gave the web page (books.google.com.au/...) the highest relevance score when performing the search shown in Fig. 3a and therefore the web page was ranked first, but a very low relevance score when performing the search shown in Fig. 3b and therefore the web page was ignored by design? We first consider this question from the perspective of *verification*. According to the online specification of Google (see Fig. 2), the "site" operator "can find all mentions" of the query term on the given site or domain. The search result of Fig. 3b is obviously a failure according to this specification as Google failed to find the web page (books.google.com.au/...) that met the search criteria and was indexed in Google database.

It is to be noted, however, that the search engine failure may not necessarily be caused by a programming fault. Design or policy issues (such as the use of different databases for different queries or improper algorithms for the calculation of relevance scores) may also contribute to the cause of the failure. The top level verification of the entire system, however, does not depend on these lower level design decisions [28].

Next, consider software *validation*. The logical consistency property identified by MPSite reflects users' actual needs and expectations. It is quite natural and reasonable for the users to expect that a good search engine will respond to their queries in a logically consistent way, regardless of the intent of the system designers.

Finally, to what degree is the failure-causing phrase "`tempted peaceably`" (which was generated randomly by our testing tool) representative of users' real information needs? The validity of using random testing in this research will be discussed in detail in Section 6.2. In summary, MPSite is a valid MR.

4.1.2 Experimental Design with MPSite

Because MPSite can automatically detect failures, it can be used for reliability assessment in the absence of an oracle. A series of experiments were designed for reliability assessment using MPSite, where two types of queries were issued, namely, English queries and Chinese queries, as we wanted to see the impact of the operational profile (language) on reliability. These two types of queries are listed in the "usage pattern" column in Table 1. A usage pattern can be considered as an operational profile because it shows how different user groups can use the search engines in different ways. The next column of Table 1 defines the result of each single metamorphic test. A single metamorphic test consists of a source query and a follow-up query, and the test result will be either "the MR is satisfied" or "the MR is violated." As explained in Section 4.1.1, a violation of MPSite implies a failure. Therefore, in Table 1, a single metamorphic test result is defined to be either a pass or a failure, depending on whether MPSite is satisfied or violated, respectively. All recorded failures are repeatable. The next column defines the result of each batch of tests. For the empirical study with MPSite, the test was run continuously for 379 hours. At the end of each hour, the hourly test result was calculated in terms of hourly ROCOF (rate of occurrence of failure), which has a value range of [0.0, 1.0]. Suppose, for instance, 1,000 pairs of valid source and follow-up responses were checked[7] from 1 a.m. to 2 a.m. of a certain day, of which 30 pairs revealed a failure, then the hourly ROCOF score for this period of time will be 0.03, and these 1,000 metamorphic tests form a batch of tests.

Query terms were formed by randomly sampling words from English and Chinese dictionaries for the English and Chinese usage patterns, respectively. Hence, different batches used different test suites. In other words, different sets of queries were issued in different hours.

4.2 MPTitle

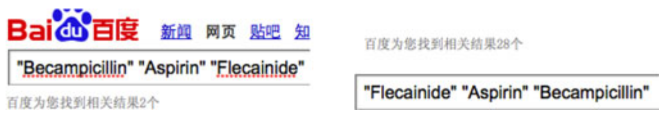
MPTitle was designed from the perspective of user validation rather than for correctness verification or failure detection. More specifically, MPTitle assesses the search engine's capability of understanding and abstracting web pages and understanding user intent.

4.2.1 Validity of MPTitle

A violation of MPTitle means that one of the following has occurred: (1) a failure, as shown in Section 3.2.2; or (2) an improper page title has been generated by the search engine; or (3) a poor synonym-based or semantic search, due to the search engine's incapability of understanding the user intent—inclusion of the page title in the follow-up query is a clear indication of the user indent but the search engine could not find it. In all three situations, the violation of MPTitle may result in poor user experience.

To avoid possible truncation of long query terms by search engines, long queries were excluded from our experiments.

7. Note, however, that this does not mean that there are 1,000 unique pairs of source and follow-up queries because when k ($k > 1$) web pages in the source response are from the same domain, there will be k identical pairs of source and follow-up queries and, hence, we only need to run them once instead of k times.



"Becampicillin" "Aspirin" "Flecainide"

百度为您找到相关结果2个

百度为您找到相关结果28个

"Flecainide" "Aspirin" "Becampicillin"

Fig. 7. Excerpts of SERPs that show inconsistency detected against MPReverseJD: Baidu found only two results for ["Becampicillin" "Aspirin" "Flecainide"] (left), but 28 results for ["Flecainide" "Aspirin" "Becampicillin"] (right).

4.2.2 Experimental Design with MPTitle

As shown in Table 1, the experimental design of MPTitle is very similar to that of MPSite, except that: (1) the result of each single metamorphic test is "found" or "not found" (not "pass" or "failure"), where "not found" indicates poor search results from the users' perspective; and (2) the result of each batch of tests is an hourly rate of occurrence of anomaly "ROCOA" (not rate of occurrence of failure, "ROCOF")—where "anomaly" refers to a violation of MPTitle. Obviously, the higher ROCOA values mean a worse perceived search quality.

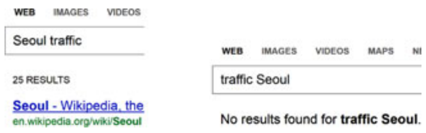
4.3 MPReverseJD

We next discuss the validity of the MPReverseJD MR, and then its experiment design.

4.3.1 Validity of MPReverseJD

A search engine may return different results for queries that differ in word order, especially when the order is important, potentially altering the meaning of the query. For instance, "Sunday school" and "school Sunday" have different meanings: The former means a class or school, generally affiliated with a church, that teaches children biblical knowledge on Sundays; whereas the latter does not have this meaning. As another example, "you eat" and "eat you" have completely different meanings. To minimize the impact of word order on query meanings, only names were used as basic query items for MPReverseJD. We further designed three usage patterns: the first pattern uses people's names only, the second pattern uses company names only, and the third pattern uses drug names only. To reduce the size of the result set, each name is enclosed by double quotes—it is each individual name, such as "`Vincent Van Gogh`" "`Elvis Presley`," rather than the entire group, such as "`Vincent Van Gogh Elvis Presley`," that is quoted. Therefore, a Web page satisfying a source query "`Vincent Van Gogh`" `AND` "`Elvis Presley`" must also satisfy its follow-up query "`Elvis Presley`" `AND` "`Vincent Van Gogh`." MPReverseJD, therefore, assesses the search engines' capability in both keyword-based search and semantic search. Keyword-based search is assessed because double quotation marks are used; semantic search is also assessed because the source and follow-up queries are semantically similar and hence should produce similar results.

As an example of a detected inconsistency, Fig. 7 (left) shows that Baidu found "two" results for the source query


WEB IMAGES VIDEOS

Seoul traffic

25 RESULTS

Seoul - Wikipedia, the
en.wikipedia.org/wiki/Seoul

WEB IMAGES VIDEOS MAPS NE

traffic Seoul

No results found for **traffic Seoul**.

Fig. 8. Failure detected against SwapJD: Bing found 25 results for [Seoul traffic] (left), but 0 results for [traffic Seoul] (right).

[“`Becampicillin`” “`Aspirin`” “`Flecainide`”], but Fig. 7 (right) shows that Baidu found “28” results for the follow-up query [“`Flecainide`” “`Aspirin`” “`Becampicillin.`”]. We reiterate that the inconsistency was repeatable and, for all search engines under study, spaces among words mean “AND.”

Of course, reversing the word order may have an impact on the word weightings calculated by the search engine and, depending on the ranking algorithm, a web page p that receives a high relevance score for the source query [“`Becampicillin`” ”`Aspirin`“ ”`Flecainide`”] may not receive the same relevance score for the follow-up query [“`Flecainide`” “`Aspirin`” “`Becampicillin`”]. However, this should only affect the ranking, not the existence, of the web page p in the SERP—the SERP is very small, so if there are more web pages satisfying the user’s search criteria then they should also be returned to meet the user’s information needs. From the perspective of user experience and validation, the users will be unhappy to receive only two results if there are actually at least 28 web pages containing all of the words. It is also quite natural and reasonable for users to expect a stable search engine to return similar results for similar queries. Besides, from the perspective of software verification, the online specifications of the search engines under study all state that *all* web pages containing the query terms will be returned. The inconsistency shown in Fig. 7, therefore, is a failure, or at least a bad case, which may result in poor user experience.

4.3.2 Experimental Design with MPReverseJD

Each usage pattern has its own name list (in English), from which query words are randomly chosen. The name list for “people” contains the names of 200 famous people (such as “Vincent Van Gogh” and “Elvis Presley”). The name list for “companies” contains the names of 200 famous companies (such as “Royal Dutch Shell” and “Volkswagen”). The name list for “drugs” contains 200 randomly selected drug names (such as “Acetazolamide” and “Vidarabine”). The result of each batch of tests is the hourly average Jaccard coefficient.

4.4 SwapJD

Similar to MPReverseJD, the design of SwapJD was also inspired by the idea that similar queries, such as [`today’s movies in Redmond`] and [`Redmond movies today`], should have similar results if the search engine is robust (insensitive) to the nonessential differences between the queries.

4.4.1 Validity of SwapJD

To discuss the validity of SwapJD, we need to first answer the following question: How can we ensure that the source and follow-up queries are semantically similar? To meet this challenge, we designed three sets of nouns. The first set contains 20 city names, namely:

S_{where} = {`Amsterdam`, `Antwerp`, `Athens`, `Atlanta`, `Barcelona`, `Beijing`, `Berlin`, `Helsinki`, `London`, `Melbourne`, `Montreal`, `Moscow`, `Oslo`, `Paris`, `Rome`, `Seoul`, `Stockholm`, `Sydney`, `Tokyo`, `Toronto`}.

The second set contains seven nouns describing time/date:

S_{when} = {`afternoon`, `evening`, `midnight`, `morning`, `today`, `tomorrow`, `yesterday`}.

The third set contains 20 nouns as follows:

S_{what} = {`airport`, `book`, `bus`, `car`, `food`, `game`, `library`, `magazine`, `movie`, `music`, `newspaper`, `Olympics`, `pollution`, `population`, `school`, `shop`, `song`, `story`, `traffic`, `weather`}.

The set of all source queries is: $S_{where} \times S_{when} \cup S_{where} \times S_{what} \cup S_{when} \times S_{what}$, where “$\times$” denotes the Cartesian product of two sets. Hence, there is a total of $20 \times 7 + 20 \times 20 + 7 \times 20 = 680$ source queries, and also 680 follow-up queries, which are repeated hourly. Because of the nature of the words in S_{where}, S_{when}, and S_{what}, swapping the two words in a query will not change the meaning.

SwapJD can detect failures and anomalies. For instance, Fig. 8 (left) shows that Bing returned 25 results for the source query [`Seoul traffic`]. It is unusual that only 25 results were found for such a popular query. But it is even more surprising to see that Bing reported “No results found” for the follow-up query [`traffic Seoul`] (Fig. 8 (right)). The Jaccard coefficient in this case is 0 and this inconsistency was repeatable at the time of the experiment.

It is to be noted that the design of SwapJD is different from that of MPReverseJD. The latter was designed to assess the search engine’s web page retrieval capability in terms of not missing any relevant web page. To detect a missing web page, MPReverseJD uses rigorous measures to enhance accuracy and exactness, namely, (1) words are enclosed by double quotes, and (2) the source response is limited to be 20 titles or less. These measures, however, are not applied in SwapJD. This is because SwapJD has a different design goal, which is to assess the search engine’s stability in *ranking* a larger number of web pages where *synonym-based search* is involved.

Next, as pointed out previously, swapping the two words in a query may impact on the weightings assigned by the search engine and, depending on the ranking algorithm, a web page that receives a high relevance score for the source query might not receive the same relevance score for the follow-up query. We should therefore not expect a perfect Jaccard coefficient of 1.0—some discrepancies between the source and follow-up responses should be allowed. It should also be noted that SwapJD was *not* designed to detect failures, although in many situations failures can indeed be detected, as shown in Fig. 8. Instead, SwapJD was designed from the perspective of understanding search engine behavior with a focus on its stability for semantically similar queries. Information obtained from the tests can be useful for developers to verify the implementation against their design goals with respect to the search engine's sensitivity to word orders, and can also be useful for users to reconstruct their queries when they are not satisfied with the initial search results (for instance, by changing the word order if the search engine is found to be order-sensitive in our study). It should also be noted that the design goal of SwapJD is different from that of Imielinski and Signorini [29], which required an ideal semantic search engine to return exactly the same results in exactly the same order for semantically equivalent queries.

4.4.2 Experimental Design with SwapJD

There are five usage patterns for SwapJD, namely, *Universal*, *site:com*, *site:edu*, *site:mil*, and *site:lc*, as shown in Table 1. An example of the Universal usage pattern is the source query [`Seoul traffic`] and its follow-up query [`traffic Seoul`] as shown in Fig. 8. It has the most basic form without any additional operator. In the site:com pattern, all results are requested from the .com domain. An example of this type is a Google search [London Olympics site:com] and its follow-up query [Olympics London site:com]. The usage patterns site:edu, site:mil, and site:lc are defined similarly.[8]

For each usage pattern and each search engine, all 680 pairs of source and follow-up queries were executed every hour. Therefore, in Table 1, the seventh column for the row "SwapJD" shows "Yes." These 680 metamorphic tests formed a "batch" of tests. They were repeated every hour because the search engines could have different behavior at different times for the same set of queries. The result of each single metamorphic test was the Jaccard coefficient of the top 50 search results. When the size of a result set was smaller than 50, all search results were considered. Hence, 680 Jaccard coefficients were collected every hour, from which the hourly means were calculated.

4.4.3 Justification for the SwapJD Usage Patterns

We next consider the *site:com*, *site:edu*, *site:mil*, and *site:lc* usage patterns. The .com domain was originally intended for commercial use, but it has become the most popular top-level domain, and is used by all types of entities. The .edu domain, in contrast, is limited to US educational institutions, which is less popular (that is, smaller) than the .com domain. The .mil domain is limited to US military, which is less popular than the .edu domain. Finally, .lc is the national domain of Saint Lucia, an island country in the eastern Caribbean Sea with a population of about 165,595 (2010). This is not a popular domain either. We selected the above four domains (in addition to the Universal pattern that covers all) because they are very different in nature and size and, hence, their test results will enable us to see whether the domain can have an impact on the search engines' performance.

8. It is to be noted that these usage patterns are different from the metamorphic relation MPSite, although all of them make use of the "site:" operator. The usage patterns of MPSite are the English and Chinese languages rather than the domains.

At first glance, one may think that a search engine will naturally perform better on smaller domains because they are easier to index and search. Why, then, should we still conduct this study? First, from the perspective of software validation, some searchers may expect that their search engines will return equally good results regardless of the domain size or type, as a basic scalability requirement. Whether or not this requirement is met is yet to be validated.

Another type of stakeholder—businesses who are concerned about the visibility of their websites to searchers—do not want their websites' domain to negatively impact on the search engine results (for example, see [35]). It is therefore meaningful to investigate whether or not the search engine can perform equally well for different domains.

Furthermore, without evidence, it cannot be assumed that a search engine will perform better on smaller domains. In fact, some search or ranking algorithms work better for large scale problems than for small scale problems [36]. It has also been reported that traditional search engines such as Google tend to interpret queries globally and hence may not work well for small domains [37]. When searching the web, there is a phenomenon known as users' *domain bias*, where users tend to click on "top domains" in a SERP [38]. In this study, we investigate whether a type of domain bias exists from the perspective of search results quality, rather than user clicks.

4.5 Top1Absent

The validity of Top1Absent is obvious. Consider, for instance, a source query ["`stanford`"]. Google returned 29,600,000 results, and the website `www.stanford.edu` was ranked first. As this website is from the .edu domain, the follow-up query will be constructed as ["`stanford`" `site:edu`]. For this query, the website `www.stanford.edu` should still be retrieved; otherwise there must be something wrong in the search engine. Here, the requirement is quite relaxed as the website is only required to be "retrieved" rather than to be "ranked first." So, as long as the URL `www.stanford.edu` is included in the search results for the follow-up query (regardless of its ranking), Top1Absent will be satisfied.

A practical difficulty in verifying Top1Absent is that the search engines under study will not return the complete list of search results when the list is very large. For example, the result count may show that "1,000,000 results were found," but the search engine may actually only provide the URLs of the top 100 results because it is generally believed that most users would only look at the top few results. As explained earlier, we also believe that most users are unlikely to browse beyond the top 50 results. In other words, even if the target website is retrieved, it is likely to be missed out by the user if its ranking is too low. We decided, therefore, to look at the top 50 results for the follow-up query: If the first ranked result for the source query A does not appear among the top 50 results for the follow-up query B then an anomaly ("not found") will be recorded.

In the experiments, we randomly sampled 500 different words from an English dictionary (excluding common words such as "of") to serve as source queries (with double quotes applied). These 500 source queries and their dynamically constructed follow-up queries formed one batch of tests, which were performed on an hourly basis for every search engine under test. As explained earlier, the search engines can have different behavior at different times for the same queries, and therefore testing with the same set of source queries at different times is meaningful. The result of a batch of tests is the ROCOA (rate of occurrence of anomaly) of the current batch, and has a value in the range of [0.0, 1.0].

5 Results of Experiments

This section will present the empirical study results for each MR identified in Section 3. Both statistical and practical significance of the results will be analyzed.

5.1 Experiments with MPSite

We first identify the independent and dependent variables of the experiments, and then perform visual and statistical analyses of the results.

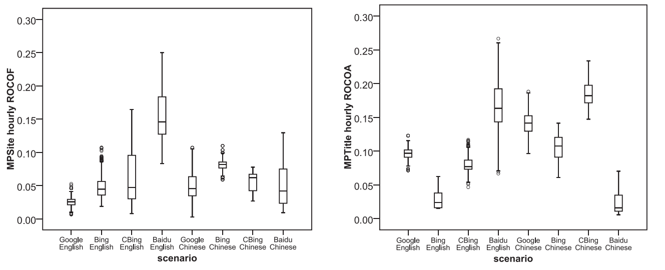
5.1.1 Independent and Dependent Variables

The independent variables of the experiments with MPSite include (1) usage pattern ("English queries" or "Chinese queries") and (2) search engine ("Google," "Bing," "Chinese Bing," or "Baidu"). Therefore, there are a total of $2 \times 4 = 8$ combinations (scenarios) of independent variable values. The dependent variables are the ROCOF scores collected hourly for each of the eight scenarios throughout 379 hours of experiments. All eight scenarios were tested in the same hours (that is, at the same time).

For Google, about 1,000 pairs of source and follow-up responses were checked per hour for each language, and hence a total of approximately 379,000 different pairs were checked across the 379 hours for each Google usage pattern. For the other six scenarios (that is, Bing English, Bing Chinese, Chinese Bing English, Chinese Bing Chinese, Baidu English, and Baidu Chinese), in each scenario about 3,000 pairs of source and follow-up responses were checked per hour and, hence, approximately $3{,}000 \times 379 = 1{,}137{,}000$ different pairs were checked across the 379 hours. (Google's test amount was lower because, during the experimental period, Google processed fewer queries per hour than did the other engines.) As a result, the empirical study with MPSite checked a total of approximately $379{,}000 \times 2 + 1{,}137{,}000 \times 6 = 7{,}580{,}000$ pairs of source and follow-up responses.

5.1.2 Comparison of MPSite ROCOF Scores

Fig. 9a shows the box plots of the distributions of the MPSite hourly ROCOF scores (the y-axis) for the eight scenarios (the x-axis), where "CBing" is short for Chinese Bing. A box plot shows the median, interquartile range, outliers and extreme values of the dataset. The top and bottom bars represent the maximum and minimum values, respectively (excluding outliers). The top and bottom of the box denote the third quartile (25 percent of data are greater than this value) and the first quartile (75 percent of data are greater than this value), respectively. The horizontal line inside the box represents the median. Circles and asterisks represent outliers. Using the software package SPSS, outliers are identified as cases that are more than $1.5 \times d$ below the lower or above the upper hinge of the box, where d is the box length.



(a) MPSite hourly ROCOF. (b) MPTitle hourly ROCOA.

Fig. 9. Distributions of MPSite and MPTitle results.

As explained in Section 3.1, MPSite is focused on the search engines' reliability for retrieving web pages containing an exact word or phrase. Lower ROCOF values indicate higher reliability of the search service. A visual analysis of Fig. 9a shows that all hourly ROCOF scores of all scenarios are above 0. This means that none of the search engines was perfect: Each search engine under each usage pattern produced failures in each and every hour. An interesting observation is that different search engines had very different performance. Relatively speaking, the most reliable service was Google's English search, whereas the least reliable service was Baidu's English search, whose hourly ROCOF reached as high as around 0.25 in the worst case. Baidu's Chinese search, however, strongly outperformed its English search, whereas Google's and Bing's English search outperformed their corresponding Chinese search. These findings seem to agree with the fact that Baidu is a China-based search engine (and hence better designed for, or more trained in, Chinese queries than English queries), whereas Google and Bing are US-based engines with more English queries.

A one-way ANOVA (analysis of variance) confirmed that there is a statistically significant[9] and practically significant difference among the means of the eight scenarios ($p < 0.0005$, $\eta_p^2 = 0.658$). *Partial eta squared* (η_p^2) is a measure of the *effect size* in ANOVA [39]. There are several effect size measures/estimates in ANOVA, such as omega squared, epsilon squared, and eta squared. But they tend to differ only slightly, especially with large samples as in the present study [40]. In one-way ANOVA, the values of partial eta squared and eta squared are the same, for which small, medium and large effects are generally considered to be 0.01, 0.06, and 0.14, respectively [39].

We further performed *post hoc multiple comparisons* to compare the means. Because equal population variances are not assumed, we used the Games-Howell procedure for the multiple comparisons as this procedure generally offers the best performance in this situation [41]. Games-Howell is also accurate when sample sizes are unequal [41]. It has

9. As a convention [39], where there is no ambiguity, "statistical significance" or "statistically significant" without further specification in the present paper can be respectively interpreted as "statistical significance" or "statistically significant" at no more than the .05 level.

TABLE 2
Details of MPReverseJD Experiments

Search engine	Usage pattern	tcp/h	OH	Ttcp
Google	People	500	150	75,000
	Companies	500	150	75,000
	Drugs	500	150	75,000
Bing	People	1,000	452	452,000
	Companies	1,000	452	452,000
	Drugs	1,000	452	452,000
CBing	People	1,000	205	205,000
	Companies	1,000	205	205,000
	Drugs	1,000	205	205,000
Baidu	People	1,000	185	185,000
	Companies	1,000	185	185,000
	Drugs	1,000	185	185,000

tcp/h: number of test case pairs per hour (approximate); OH: number of observed hours; Ttcp: total number of test case pairs.

been found that, in most situations, the query language had a statistically significant impact on the search engines' performance and that the performance of different search engines for the same language was also different with a statistical significance.

In the multiple comparisons, we also measured the practical significance (effect size) using Cohen's d [39]: d is the absolute value of the difference of the two means divided by the square root of the mean of the two variances. While η_p^2 estimates the effect size in one-way ANOVA, Cohen's d estimates the effect size when comparing two means. Cohen suggested the following benchmarks to interpret d: medium effect size ($d = 0.5$) is "an effect of a size likely to be apparent to the naked eye of a careful observer," small effect size ($d = 0.2$) is noticeably "smaller yet *not trivial*," and large effect size ($d = 0.8$) is "the same distance above medium as small is below it" [42]. Therefore, we considered a d value of 0.20 or above to be significant (nontrivial). In our statistical analyses, all statistically significant differences are also found to be practically significant, and most of the d values are large (above 0.8).

5.2 Experiments with MPTitle

The experimental procedure of MPTitle is very similar to that of MPSite, and the two sets of experiments were also conducted during the same period of time, except that MPTitle experiments had 380 (not 379) hours of observations. Similar to MPSite, about 1,000 pairs of source and follow-up responses were checked per hour for Google English and Google Chinese, and about 3,000 pairs were checked per hour for the other six scenarios.

Fig. 9b shows the box plots of the distributions of hourly ROCOA scores for all eight scenarios. A visual analysis shows that none of the search engines was perfect as their hourly ROCOA scores are all above 0: They all produced failures or anomalies in each and every hour. The query language has a strong impact on the performance: Google English and Bing English continued to outperform Google Chinese and Bing Chinese, respectively. CBing English also outperformed CBing Chinese, indicating that CBing did not favor Chinese search even though it was designed for Chinese users. On the other hand, Baidu Chinese continued to outperform Baidu English. Baidu also continued to rank as the best Chinese engine but the worst English one.

Statistical analysis results are consistent with the visual analysis: A one-way ANOVA shows that there is a statistically and practically significant difference among the means of the eight scenarios ($p < 0.0005$, $\eta_p^2 = 0.890$). Further multiple comparisons and effect size analysis show that the query language had a statistically and practically significant impact on every search engine's performance and that different search engines also performed differently (with a statistical and practical significance) under each language. Bing and Baidu were the best English and Chinese engines, respectively.

5.3 Experiments with MPReverseJD

MPReverseJD is focused on the retrieval stability, in terms of returning similar results for similar queries.

5.3.1 *Independent and Dependent Variables*

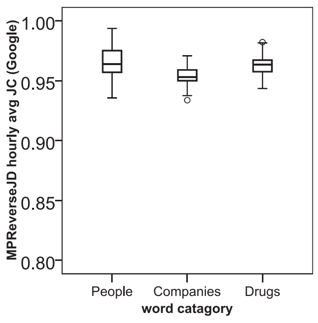
The independent variables include (1) usage pattern, which is "people," "companies," or "drugs," and (2) search engine, which is "Google," "Bing," "Chinese Bing," or "Baidu." Therefore, there are $3 \times 4 = 12$ scenarios. The dependent variables are the average Jaccard coefficients calculated hourly for each of the 12 scenarios. More details of the experiments are given in Table 2. Because of resource limitations, there are some differences in the number of hours and number of test case pairs between different search engines, as shown in Table 2. Nevertheless, for the same search engine, the three sets of experiments for the three usage patterns were conducted at exactly the same hours.

5.3.2 *Comparison of MPReverseJD Jaccard Coefficients*

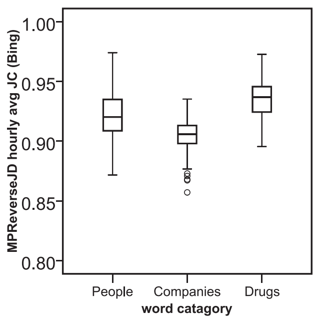
Fig. 10 shows the box plots of the distributions of the MPReverseJD hourly mean Jaccard coefficients for all 12 scenarios. A higher coefficient implies better stability in web page retrieval. A visual analysis of Fig. 10 shows that all hourly scores of all search engines are below 1. This means that none of the search engines was perfect in any hour. Among all four engines, Google was the most stable whereas Baidu was the least stable. The usage pattern (word category) appears to have an impact on the search engines' performance.

A one-way ANOVA confirmed that there is a statistically and practically significant difference among the means of the 12 scenarios ($p < 0.0005$, $\eta_p^2 = 0.759$). Post hoc multiple comparisons and effect size analysis show that the usage pattern (that is, the type of the query words) had a statistically and practically significant impact on the web page retrieval stability of all search engines. Furthermore, there are statistically and practically significant differences among the search engines' performance under every usage pattern, where Google and Baidu were found to be the most and least stable search engines, respectively.

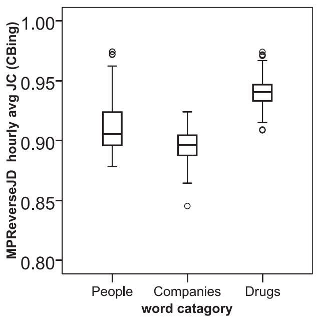
A further question is: What is the difference among the three types of words that makes the stability of their search results significantly different? Every test case used for MPReverseJD consists of names. For a search engine to process such a query, *named entity recognition* techniques



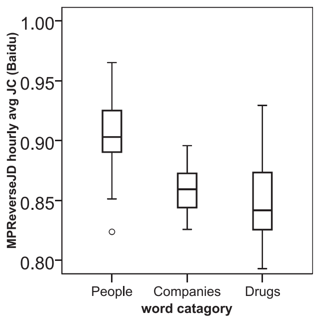
(a) Google



(b) Bing



(c) CBing



(d) Baidu

Fig. 10. Distributions of MPReverseJD results. JC: Jaccard coefficient.

are probably used [43]. It was reported that the levels of difficulty in recognizing these three types of named entities (people, companies, and drugs) are different [44]. The frequencies of occurrence of these three types of names in user queries are very different, too [43], which might have caused the search engines' machine learning software to be trained differently. Furthermore, company names normally have good commercial and advertising values, and drug names also have such values to a certain degree. Personal names are more neutral. The above factors, when combined, may have influenced the quality of search for different types of names.

5.4 Experiments with SwapJD

SwapJD has five usage patterns. We first consider the basic form, the "Universal" usage pattern, and then consider the others.

5.4.1 Results of Experiments Using the "Universal" Usage Pattern

For the Universal usage pattern, the 680 pairs of source and follow-up queries were executed every hour for a period of 548 hours for each search engine under study. For each search engine, therefore, a total of 548 hourly average Jaccard coefficients were collected. The distributions of these results are shown in Fig. 11, from which it can be seen that, although the same 680 pairs of tests were performed every hour, the performance of all four search engines varied a lot over time. All hourly mean Jaccard coefficients are below 1, that is, the stability quality is not perfect for any of the search engines.

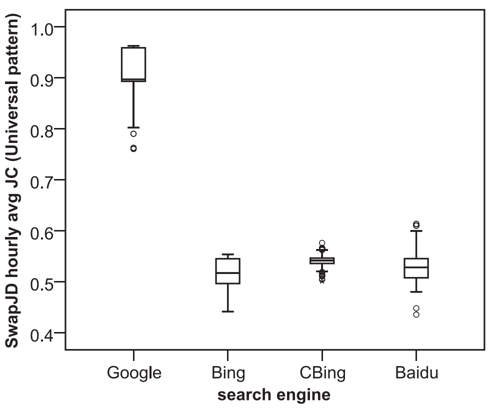


Fig. 11. Distributions of SwapJD results under the Universal usage pattern. JC: Jaccard coefficient.

Fig. 11 shows that Google was superior to all other three engines in ranking stability. The differences between Google's and the other engines' means have large effect sizes (Cohen's d values are all greater than 10). To enhance usability (more specifically, to enhance the usefulness of the software systems [45]), users should pay attention to the word order when searching with Bing, CBing, or Baidu as these engines are much more sensitive to this than Google. Users of these three engines may also consider changing the word order to query again when they are not satisfied with the initial search result.

5.4.2 Results of Experiments Using the site:com, site:edu, site:mil, and site:lc Usage Patterns

We next investigate the impact of usage patterns (that is, different search domains) on the quality of ranking. The experiments for different search engines were conducted at about the same time; however, owing to resource limitations, the durations of the experiments for different engines are not identical, which are 103, 148, 131, and 100 hours for Google, Bing, CBing, and Baidu, respectively. The difference in durations, however, does not affect the analyses of the experimental results. For the same search engine, the four sets of experiments for the four usage patterns were conducted at exactly the same time.

The results are shown in Fig. 12, which includes a total of 16 scenarios. In most situations, the hourly mean Jaccard coefficients are below 1. For all four search engines, the .lc domain outperformed the .mil domain; the .mil domain outperformed the .edu domain; and the .edu domain outperformed the .com domain. Statistical and effect size analyses confirmed that all of these differences (except Baidu .lc vs Baidu .mil) are both statistically and practically significant with most Cohen's d values above 0.8, and the one-way ANOVA returning significant results with $p < 0.0005$ and $\eta_p^2 = 0.814$.

The experimental results suggest that MT is useful for assessing software *scalability*: All four search engines are found to have performance degradation when searching large domains. Intuitively, the .com domain is larger than the .edu domain, which is in turn larger than the .mil and .lc domains. To confirm this intuition, we analyzed the result counts and found that, for all four search engines, the .com domain returned the largest number of results, followed by .edu, then .mil. The .lc domain was the smallest. It is therefore evident that all four search engines have better

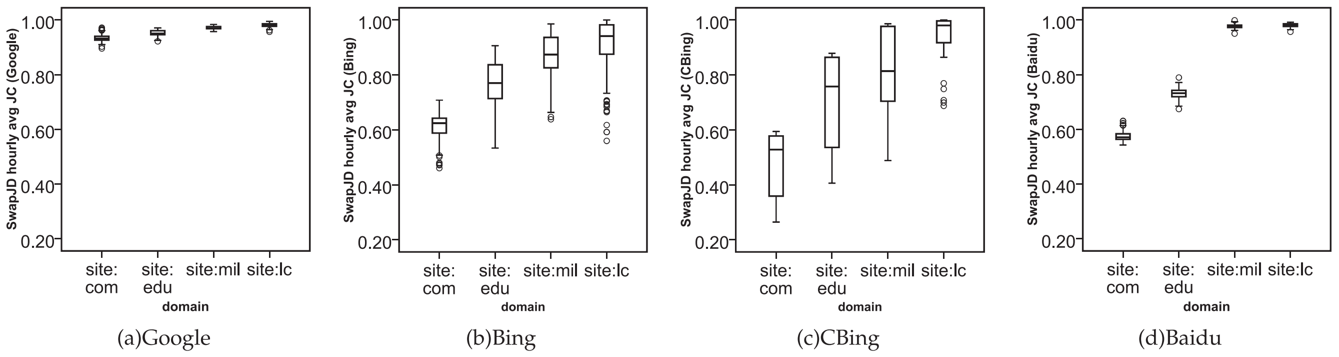


Fig. 12. Distributions of SwapJD results for different domains. JC: Jaccard coefficient.

performance in ranking stability when the domain size is small. In other words, scalability should be the reason for the search engines' performance difference across different domains. In addition to the lower number of web pages, smaller domains are also likely to be less diverse in many ways (such as in concentration and locality, web document types, user search patterns and the web usage culture [37]), which may make them easier to be searched.

In summary, SwapJD results show that Google was superior to the other engines in both ranking stability and scalability.

5.5 Experiments with Top1Absent

For Top1Absent, all four search engines were tested at the same time for a total of 353 hours. Distributions of the hourly ROCOA scores are shown in Fig. 13. Although the same source queries (500 English words) were used every hour, the performance of the search engines varied a lot over time. In the best case, all four search engines achieved an hourly ROCOA of 0. In the worst case, Google, Bing, CBing, and Baidu had an hourly ROCOA of 3.6, 14.4, 10.6, and 11.0 percent, respectively. A high ROCOA score means a high rate of anomalies (failures or bad cases). A one-way ANOVA returned significant results ($p < 0.0005, \eta_p^2 = 0.184$). Further multiple comparisons and effect size analyses show that Google and Bing yielded the best and worst search results, respectively, with both statistical and practical significance.

6 Discussions

In this section, we discuss several important software engineering issues, namely, MR identification, test case selection, causes of failures, software quality models, and implications of the empirical study results.

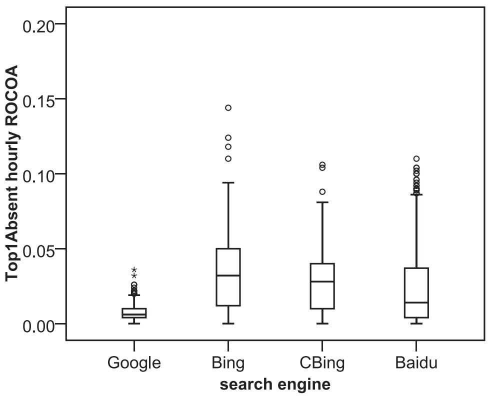


Fig. 13. Distributions of Top1Absent results (ROCOA).

6.1 Identification of Metamorphic Relations

The identification of MRs requires knowledge of the problem domain, and is therefore a manual process. In this study, we manually identified five MRs by referring to the online specifications of the search engines. These five MRs were selected because they address two types of crucial qualities (page retrieval and ranking) that are difficult to assess using conventional methods owing to the oracle problem. It is however not the intention of this study to identify an exhaustive list of MRs; instead, we studied the effectiveness of using MT as a software quality assessment method with the four search engines and five MRs. The study shows that our method is effective. By referring to the online specifications, it would not be difficult to identify additional MRs to cover other features and search operators which are increasingly being supported by modern search engines.

As a general rule, we should consider two situations when attempting to identify a comprehensive list of MRs.[10] In the first one, the follow-up input is only related to the source input without referring to the source output. For instance, MPReverseJD and SwapJD are both this kind of MR, where the follow-up query is constructed by reordering the source query's terms. In the other situation, the follow-up input is related to both the source input and the source output. Examples of this type include MPSite, MPTitle, and Top1Absent, where the follow-up query is constructed by not only referring to the source query's terms but also referring to the domains or titles of the search results included in the source response. The above observation may provide hints for the development of methods and tools to assist software testers with the identification of useful MRs.

6.2 Selection of Test Cases

The effectiveness of MT not only depends on the MR but also on the source test cases. Obviously, if a source test case leads to an incorrect output (though unrevealed because of the oracle problem), it would be likelier to trigger a violation of the MR. Therefore, in order to study the effectiveness of

10. For ease of presentation, consider those MRs that involve one source execution (which accepts a source input and generates a source output) and one follow-up execution (which accepts a follow-up input and generates a follow-up output).

MRs for software quality assessment, we restricted ourselves to use randomly sampled words as source test cases in our experiments. This practice is also in line with the reliability estimation principle that requires unbiased samples for valid statistics [21].

A question then arises: To what degree do the test results based on random words reflect the search engines' actual performance for human-generated query terms? To answer this question, we need to consider both verification and validation. For software verification purposes, random testing is quite cost-effective for fault detection when compared with other techniques [46], [47]. It is simple in concept and easy to implement, and the random test cases can often reveal unexpected failures. The large number of failures and anomalies detected in the present research provide further evidence of the fault-detection capability of random testing.

For the purpose of validation, random sampling of test cases following a usage-profile-based probability distribution is often preferred [47]. There are, however, practical reasons why we did not adopt this approach in the present study. First of all, the usage profiles (such as the user logs) of the search engines under test were not available to us. Second, even if we were able to collect such a usage profile, given that we test four different search engines together, it would be unfair to test one search engine using another search engine's usage profile. Needless to mention, for the same search engine, its users also have different usage profiles. Third, user interests change quickly over time; hence yesterday's popular query terms may no longer be popular today. Our random sampling strategy, which does not refer to the system logs, is a simple solution to the above problems and provides a common ground for fair comparison of different systems.

A further question is: Are the randomly generated query terms really useful/meaningful from the users' perspective? Or are they just meaningless random strings? To answer this question, we should first note that different users have different information needs. Even a random string that is useless/meaningless for the majority of users can be very useful for a searcher looking for a particular product model number or zip code [28]. Indeed, it is hard to say which queries are not important. It is therefore reasonable for the users to expect that the search engine works for all queries regardless of whether they are "popular" or not.

Furthermore, although this research adopted a random sampling strategy for test case generation, the query terms thus generated are by no means meaningless random strings. For MPSite and MPTitle, a source query is a phrase enclosed by double quotation marks—such a phrase is meaningful because, first, it consists of valid English or Chinese words and, second, the design of the experiments guarantees that the source response always contains one or more results, which means that the phrase is part of some web documents and is hence meaningful. According to Google specification (Fig. 2), users are likely to issue this type of query when searching for song lyrics or a line from a book/article. The follow-up query of MPTitle further includes the page title generated by the search engine, such as `[Indianapolis Correspondence Google News]` as shown in Section 3.2.2, which is meaningful and mimics human-generated input.

For MPReverseJD, three types of names (that is, named entities) were used as test cases, which are meaningful, too. According to an analysis conducted by Guo et al. [48], about 71 percent of real-world queries contained named entities. People search and company search are common types of searches; drug search is a specific type of search that may be issued by patients and medical personnel.

For SwapJD, the following three word lists were used to generate test cases: S_{where}, S_{when}, and S_{what}. These three lists were selected to ensure that (1) any pairwise combination is meaningful and falls within the common types of searches that users perform, and (2) swapping the word order in the query will not change the meaning of the query. The generation of these three sets of words involved manual inspection of randomly sampled words to ensure the satisfaction of the above two requirements.

Finally, for Top1Absent, each source query contains one word randomly sampled from an English dictionary, excluding common words such as "of." Every English word is meaningful; so is the query.

It should be noted, however, that reliability estimation (as part of validation) is always usage-profile dependent. Different user groups may have different usage profiles as they use the search engine in different ways. Scientific researchers, for instance, may frequently look for research papers by issuing queries containing scientific terminologies; whereas prospective students may frequently search for course information on the .edu domain. Developers, on the other hand, may have a usage profile based on the system's user logs. As a result, different stakeholders may get different reliability estimates for the same search engine. As an independent tester, we used a random sampling strategy in the present research. For verification purposes, our strategy has been very effective in failure detection. For validation purposes, our strategy has also been very effective in revealing the search engines' behavioral differences under different operational profiles. In Section 6.3, we will further show that our approach complements the developer's user-log-based approaches to software quality assurance.

6.3 Causes of Failures

A search engine is a web application. Therefore, a search engine failure can be considered as a *web failure*, defined as "the inability of a web application to correctly function or deliver information or documents required by the users" [49]. There are typically two causes for web failures: (1) problems in the web application software (such as "not meeting functionality and the users' needs"); and (2) failures in the supporting infrastructure (such as hardware or network failures) [49].

The failures detected in our experiments are violations of MRs (inconsistencies / logical errors in the behavior of the search engines) and are repeatable, rather than system availability or network problems caused by a hardware failure. The failure-causing queries can be considered to be a kind of "difficult queries," for which the quality of search results is poor [50]. It has been reported that there are mainly three causes for the poor quality of search results: (1) factors relating to the retrieval method (the algorithm or its implementation); (2) properties of the data to be retrieved; or (3) linguistic features of the queries that make it difficult for the search engine to understand, such as ambiguity [50]. As software applications consist of both code and data, causes (1) and (2) are both software issues that reflect the capability or quality of the software that implements the search engine. Cause (3) is not relevant to this research because, first, our experiments were carefully designed to avoid ambiguity. Second, cause (3) may explain why a user is not satisfied with the search results (a subjective criterion) but cannot explain why an MR is violated (an objective criterion). Third, we found that the failure-causing queries for different search engines were different, and this phenomenon "is probably due to system dependent factors such as the specific retrieval methodology and implementation details" [50].

We discussed our method and some of the detected failures with colleagues at Baidu Inc, Microsoft Research, and Google Inc. Colleagues at Baidu indicated that the detected failures are "bad cases" that revealed "logic errors" in the software. They further pointed out that logic errors are difficult to detect as they do not crash the system while generating the incorrect outputs, and that they can exist in the implementation or design of the software. We were also told that log file analysis is the main approach used by the company to detect potential bad cases. Analyzing system log files, however, cannot reveal problems not recorded in the log files. For example, it cannot identify a failure-causing query if this query is never issued by any user during the period of analysis. Log file analysis may also generate both false positives and false negatives. Colleagues at Baidu indicated that our testing approach, which generates random test cases rather than looking at what is recorded in the system log files, would complement their log file analysis approach.

Colleagues at Microsoft Research indicated that the assessment of search engines is indeed challenging. They recognized our approach as a useful software testing and evaluation method for search engines, and commented that a closely related technique based on the same principle had been used informally within Microsoft to assess search engine software robustness (as discussed in Section 3.3). They were able to repeat and confirm some of our reported software failures.

Colleagues at Google indicated that the reported cases could have been caused by software related factors and that further investigation was under way.

In summary, our approach can be used for verification, validation, and quality assessment of the software systems of the search engines. Without more details of the design and algorithms used (which are commercial secrets), it is hard for a user or an independent tester to tell whether a detected failure is caused by a fault in the implemented code or a flaw in the design or algorithm. But, from the users' perspective, it does not matter, because users are only concerned about the quality of the final operational software, regardless of whether the fault is in the code or in the design. To this end, our approach enables the users to perform validation and quality assessment of the software. Developers, on the other hand, know the details of the algorithms adopted, and therefore can identify MRs for these algorithms, verify the implementation using these MRs, and find the root cause of detected failures.

6.4 Software Quality Models

In this paper, we have extended MT into a software quality assessment method and conducted a study using search engines. We now set our work in context by considering its relationship with software quality models that constitute the foundation of software product quality control [51]. More specifically, we consider the well-known software quality model standard ISO/IEC 25010 [45] as this standard will have an effect on existing quality models in practice [51].[11]

ISO/IEC 25010 defines the quality of a system as the degree to which the system satisfies the stated and implied needs of its various stakeholders. The quality models can, for example, "be used by developers, acquirers, quality assurance and control staff and independent evaluators, particularly those responsible for specifying and evaluating software product quality." A wide range of stakeholders have been considered, such as software developers, owners, users, and so on [45]. The standard contains a *product quality model* and a *quality in use model*, both of which have a hierarchical structure that breaks the concept of software product quality into smaller and more manageable parts named *characteristics*, which can, in turn, consist of *subcharacteristics*. If a characteristic or subcharacteristic cannot be directly measured, a collection of measurable quality-related *properties* need to be identified, with associated quality measures. The MRs and their associated evaluation metrics presented in this paper can therefore be considered to be such quality-related properties and quality measures, respectively.

The product quality model breaks product quality into eight characteristics, each of which consists of a set of subcharacteristics, as shown in Table 3 (left and upper-right corner). The *quality in use model* is shown in the lower-right corner of Table 3, and consists of five characteristics (some of which consist of subcharacteristics). Quality in use is the degree to which a product can be used by specific users to meet their needs to achieve specific goals in specific contexts of use. The quality (sub)characteristics related to the present research are starred.

Consider the *product quality model* shown in Table 3. All three MRs of the "No Missing web Page" group are related to the quality subcharacteristic *functional correctness*, which refers to the "degree to which a product or system provides the correct results with the needed degree of precision" [45], because they (namely, MPSite, MPTitle, and MPReverseJD) can be used to detect "missing web pages" that should have been retrieved according to the search engine's online specification. Furthermore, because the tests can be performed under different operational profiles and the test results are quantified on an hourly basis, all three MRs are related to *maturity* of *reliability*, where reliability refers to the "degree to which a system, product or component performs specified functions under specified conditions for a specified period of time," and maturity refers to the "degree to which a system, product or component meets needs for reliability under normal operation."

The MR SwapJD has revealed scalability problems of search engines and, therefore, is related to *capacity* of

11. The IEEE is planning to adopt ISO/IEC 25010:2011 [52], [53].

TABLE 3
ISO/IEC 25010 Product Quality Model and Quality in Use Model, Where Quality (Sub)Characteristics Related to the Present Research Are Starred

Product Quality Model
Functional suitability
Functional completeness
* Functional correctness
Functional appropriateness
Performance efficiency
Time behaviour
Resource utilization
* Capacity
Compatibility
Co-existence
Interoperability
Usability
Appropriateness recognizability
Learnability
* Operability
* User error protection
User interface aesthetics
Accessibility
Reliability
* Maturity
Availability
Fault tolerance
Recoverability
Security
Confidentiality
Integrity
Non-repudiation
Accountability
Authenticity

Product Quality Model (continued)
Maintainability
Modularity
Reusability
Analysability
Modifiability
Testability
Portability
Adaptability
Installability
Replaceability

Quality in Use Model
* **Effectiveness**
Efficiency
Satisfaction
Usefulness
Trust
Pleasure
Comfort
Freedom from risk
Economic risk mitigation
Health and safety risk mitigation
Environmental risk mitigation
Context coverage
* Context completeness
Flexibility

performance efficiency, where capacity refers to the "degree to which the maximum limits of a product or system parameter meet requirements," and performance efficiency refers to "performance relative to the amount of resources used under stated conditions."

All five MRs make use of some search operators, such as the "site:" operator, the double quotation marks, as well as the AND operator (implied by spaces among words). Therefore, all five MRs are related to *operability* under *usability*, where operability refers to the "degree to which a product or system has attributes that make it easy to operate and control," and usability refers to the "degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."

Arguably, MPReverseJD and SwapJD are also related to *user error protection* (under usability), where user error protection refers to the "degree to which a system protects users against making errors." This is because, in many situations, the users are unaware of what word order would be the best for them to type their query. For instance, most users would not care about the word order difference between the queries [`Seoul traffic`] and [`traffic Seoul`] (Fig. 8), or the difference between the queries [`"Becampicillin" "Aspirin" "Flecainide"`] and [`"Flecainide" "Aspirin" "Becampicillin"`] (Fig. 7). It is therefore important for the search engine to return accurate and complete results even if the user did not issue the query by following the "optimal order."

Next, consider the *quality in use model* shown in Table 3 (lower-right corner). The quality in use of a system characterizes the impact that the product has on its stakeholders. All five MRs studied in this paper are related to *effectiveness*, which refers to the "accuracy and completeness with which users achieve specified goals." This is because all five MRs are closely related to the accuracy and completeness of the search results.

All of the MRs except Top1Absent are also related to *context completeness* (under *context coverage*), where context completeness refers to the "degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in all the specified contexts of use." In our testing framework, the "contexts of use" are the different operational profiles such as different languages, different types of query terms, and different types of domains to search.

Arguably, all the MRs studied in the present research can also be related to *satisfaction*, defined as the "degree to which user needs are satisfied when a product or system is used in a specified context of use." In particular, the MRs can be related to the subcharacteristics *usefulness* and *trust*, whose respective definitions are the "degree to which a user is satisfied with their perceived achievement of pragmatic goals" and the "degree to which a user or [another] stakeholder has confidence that a product or system will behave as intended" [45]. This is because the users will not be satisfied with (or trust) the search engine if they find that it often fails to retrieve the expected relevant results (as per the MRs). However, we did not star the above (sub) characteristics in Table 3, because user satisfaction is best assessed by actual users interacting with the system, but the focus of this research was automatic quality assessment *without* a human oracle. Nevertheless, as all the investigated MRs were identified from the users' perspective, it is reasonable to believe that these MRs can at least serve as a (low-cost) proxy for user satisfaction (usefulness and trust).

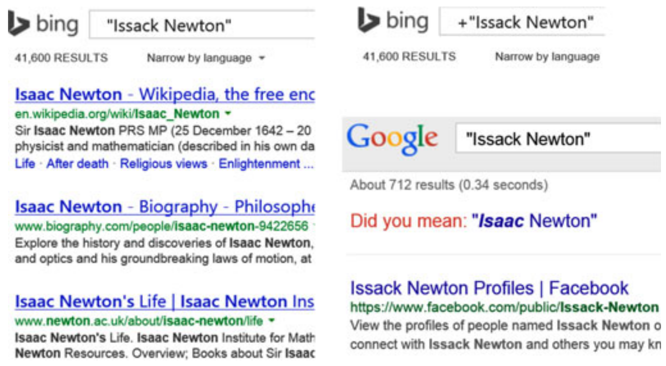
TABLE 4
List of the Best and Worst Performers

MR and usage pattern	The best performer	The worst performer
MPSite (English)	Google	Baidu
MPSite (Chinese)	Google and Baidu	Bing
MPTitle (English)	Bing	Baidu
MPTitle (Chinese)	Baidu	CBing
MPReverseJD (people)	Google	CBing and Baidu
MPReverseJD (companies)	Google	Baidu
MPReverseJD (drugs)	Google	Baidu
SwapJD (Universal)	Google	Bing and Baidu
Top1Absent	Google	Bing

6.5 Implications of the Empirical Study Results

Section 5 shows considerable variation in the performance of the different search engines for the MRs. The best and worst performers found in the empirical studies are listed in Table 4, which shows that there is no single best or worst search engine. A reason for this is that the different search engines can perform better or worse depending on the MRs and operational profiles used. It is evident, however, that Google is the best performer in most situations. Furthermore, Google is never the worst performer.

All five MRs in this paper were identified based on the online specifications of the search engines. Therefore, the experimental results suggest that, among all four search



41,600 RESULTS

Isaac Newton - Wikipedia, the free enc
en.wikipedia.org/wiki/Isaac_Newton
Sir Isaac Newton PRS MP (25 December 1642 – 20
physicist and mathematician (described in his own da
Life · After death · Religious views · Enlightenment ...

Isaac Newton - Biography - Philosoph
www.biography.com/people/isaac-newton-9422656
Explore the history and discoveries of Isaac Newton,
and optics and his groundbreaking laws of motion, at

Isaac Newton's Life | Isaac Newton Ins
www.newton.ac.uk/about/isaac-newton/life
Isaac Newton's Life. Isaac Newton Institute for Math
Newton Resources. Overview; Books about Sir Isaac

41,600 RESULTS

About 712 results (0.34 seconds)

Did you mean: "*Isaac* Newton"

Issack Newton Profiles | Facebook
https://www.facebook.com/public/Issack-Newton
View the profiles of people named Issack Newton o
connect with Issack Newton and others you may kn

Fig. 14. An exact-phrase search in Bing and Google.

engines, Google has been the most rigorously designed and implemented software system according to the online specifications. In other words, the behavior of Google is more predictable from the users' perspective. To further investigate this phenomenon, we conducted additional studies by searching for exact phrases, of which an example is shown in Fig. 14.

Fig. 14 (left) is an excerpt of Bing's SERP for the query [“`Issack Newton`”]. The double quotes in the query mean “finds the exact words” according to Bing's online specification. Bing returned “41,600 results.” We inspected the top 10 search results, and found that nine of them contained “Isaac Newton” rather than “Issack Newton.” Furthermore, Bing did not inform the user that his/her search criterion had been modified or give the user any spelling suggestion. Fig. 14 (upper-right corner) shows that a plus sign (+) was added to the query. According to Bing's online specification, the “+” operator is to further guarantee that the search must be exact and that none of the words in the query term shall be ignored. The search results, however, remained the same. Fig. 14 (lower-right corner) shows an excerpt of Google's SERP: Google returned “about 712 results,” much fewer than Bing's “41,600 results.” At the top of the SERP, Google prompted the user to check spelling by asking: `Did you mean: "Isaac Newton.`” Further, prior to the user's confirmation, Google faithfully searched for “Issack Newton” rather than “Isaac Newton”: All of the top 10 search results contained the exact phrase.

The above example shows a software design or implementation difference between Bing and Google: Bing appears to incorporate some “intelligence” in the search by including additional results that it believes would interest the user. We have pointed out to colleagues at Microsoft Research that, with this kind of design, *users of Bing have no way to perform an exact-phrase search*—this is a crucial deficiency in the software's *functional completeness*, which refers to the degree to which the set of functions covers all the specified tasks and user objectives [45]. We further found that the behavior of CBing and Baidu was also like that of Bing. In comparison, Google more rigorously followed its online specifications to faithfully search for the exact phrase.

Although the above problem was not a cause for the failures and anomalies reported in Section 5, the underlying principle, namely, how rigorously the software system was designed and implemented according to its specification, may be a reason for the good performance of Google, as reported in Section 5. Further discussions about the design and implementation differences among the search engines are beyond the scope of this paper. Although the precise causes for the detected failures or anomalies might not have been identified (due to our limited knowledge of the search engines' design and implementation), our empirical study results do provide useful hints for the search engine developers, allowing them to verify the software behavior against their design goals.

7 Summary and Future Work

Metamorphic testing was initially proposed as a verification technique, where metamorphic relations were identified by referring to the target algorithm to be implemented [3], [5]. In this paper, we have demonstrated the feasibility of MT being a unified framework for software verification, validation, and quality assessment. We conducted a study on search engines, where we identified MRs from the users' perspective without referring to the target algorithms or system specifications. More generally, this approach allows users to recognize whether or not a system is appropriate for their specific needs in the absence of complete software documentation, which is often the case with web services, poorly evolved software, and open source software.

We have applied our approach to assess several key software qualities of search engines under different operational profiles in the absence of an objective and generally recognized oracle. All ANOVA analyses returned statistically significant results with large effect size (η_p^2) values. Most multiple-comparison results also had a statistical and practical significance (with large Cohen's d values in most situations), indicating that our approach is effective. We have also discussed the investigated software qualities in the framework of the software quality model standard ISO/IEC 25010.

The empirical results demonstrate that our approach is useful for both developers and users. First, our approach can effectively detect various kinds of failures. Second, we found that the operational profiles have a significant impact on the quality of search. For a given search engine, its quality of search can be significantly different for different query languages, different types of query words, and different domains being searched. This finding provides a hint for the developers to identify the strength and weakness of their systems, and is also useful for the users to choose a suitable search engine or to better construct their queries. The ability to automatically detect failures and anomalies using MRs can also provide hints for the construction of run-time self-correction mechanisms, which will be a future research topic.

Similar to the use of a set of mutants to assess the effectiveness of a test suite in the context of mutation analysis, a set of properly selected MRs can potentially be used to assess certain software quality characteristics. Further study on this topic will be a future research area.

Acknowledgments

The authors are grateful to Microsoft Research for supporting a pilot study of this work under an academic research collaboration scheme. They wish to thank Baidu Inc. and

Google Inc. for discussions on their work. They would also like to thank Dave Towey, Antony Tang, and Lei Wang for their comments. This research was supported in part by a linkage grant of the Australian Research Council (Project ID: LP100200208).

REFERENCES

[1] C. Ghezzi, M. Jazayeri, and D. Mandrioli, *Fundamentals of Software Engineering*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.

[2] R. M. Hierons, "Oracles for distributed testing," *IEEE Trans. Softw. Eng.*, vol. 38, no. 3, pp. 629–641, May/Jun. 2012.

[3] T. Y. Chen, S. C. Cheung, and S. M. Yiu, "Metamorphic testing: A new approach for generating next test cases," Dept. Comput. Sci., Hong Kong Univ. Sci. Technol., Hong Kong, Tech. Rep. HKUST-CS98-01, 1998.

[4] T. Y. Chen, T. H. Tse, and Z. Q. Zhou, "Semi-proving: An integrated method based on global symbolic evaluation and metamorphic testing," in *Proc. ACM SIGSOFT Int. Symp. Softw. Testing Anal.*, 2002, pp. 191–195.

[5] T. Y. Chen, T. H. Tse, and Z. Q. Zhou, "Fault-based testing without the need of oracles," *Inf. Softw. Technol.*, vol. 45, no. 1, pp. 1–9, 2003.

[6] T. Y. Chen, J. W. K. Ho, H. Liu, and X. Xie, "An innovative approach for testing bioinformatics programs using metamorphic testing," *BMC Bioinformat.*, vol. 10, no. 24, p. 24, 2009.

[7] C. Murphy, K. Shen, and G. E. Kaiser, "Automatic system testing of programs without test oracles," in *Proc. 18th ACM SIGSOFT Int. Symp. Softw. Testing Anal.*, 2009, pp. 189–200.

[8] X. Xie, J. W. K. Ho, C. Murphy, G. Kaiser, B. Xu, and T. Y. Chen, "Testing and validating machine learning classifiers by metamorphic testing," *J. Syst. Softw.*, vol. 84, pp. 544–558, 2011.

[9] J. Zhang, J. Chen, D. Hao, Y. Xiong, B. Xie, L. Zhang, and H. Mei, "Search-based inference of polynomial metamorphic relations," in *Proc. 29th IEEE/ACM Int. Conf. Autom. Softw. Eng.*, 2014, pp. 701–712.

[10] V. Le, M. Afshari, and Z. Su, "Compiler validation via equivalence modulo inputs," in *Proc. 35th ACM SIGPLAN Conf. Program. Lang. Des. Implementation*, 2014, pp. 216–226.

[11] J. Regehr. (2014, Jun. 20). Finding compiler bugs by removing dead code [Online]. Available: http://blog.regehr.org/archives/1161

[12] A. Núñez and R. M. Hierons, "A methodology for validating cloud models using metamorphic testing," *Ann. Telecommun.*, vol. 70, no. 3-4, pp. 127–135, 2015.

[13] S. Segura, A. Durán, A. B. Sánchez, D. L. Berre, E. Lonca, and A. Ruiz-Cortés, "Automated metamorphic testing of variability analysis tools," *Softw. Testing, Verification Rel.*, vol. 25, pp. 138–163, 2015.

[14] M. Lindvall, D. Ganesan, R. Árdal, and R. E. Wiegand, "Metamorphic model-based testing applied on NASA DAT—an experience report," in *Proc. 37th Int. Conf. Softw. Eng.*, 2015, pp. 129–138.

[15] T. Y. Chen, T. H. Tse, and Z. Q. Zhou, "Semi-proving: An integrated method for program proving, testing, and debugging," *IEEE Trans. Softw. Eng.*, vol. 37, no. 1, pp. 109–125, Jan./Feb. 2011.

[16] X. Xie, W. E. Wong, T. Y. Chen, and B. Xu, "Metamorphic slice: An application in spectrum-based fault localization," *Inf. Softw. Technol.*, vol. 55, no. 5, pp. 866–879, 2013.

[17] Y. Cao, Z. Q. Zhou, and T. Y. Chen, "On the correlation between the effectiveness of metamorphic relations and dissimilarities of test case executions," in *Proc. 13th Int. Conf. Quality Softw.*, 2013, pp. 153–162.

[18] H. Liu, F.-C. Kuo, D. Towey, and T. Y. Chen, "How effectively does metamorphic testing alleviate the oracle problem?" *IEEE Trans. Softw. Eng.*, vol. 40, no. 1, pp. 4–22, Jan. 2014.

[19] I. Soboroff, "Dynamic test collections: Measuring search effectiveness on the live Web," in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2006, pp. 276–283.

[20] NetMarketshare. (2015, Mar.). Market share reports—desktop search engine market share [Online]. Available: https://www.netmarketshare.com

[21] M. Pezzè and M. Young, *Software Testing and Analysis: Process, Principles, and Techniques*. New York, NY, USA: Wiley, 2008.

[22] D. Lo, J. Li, L. Wong, and S.-C. Khoo, "Mining iterative generators and representative rules for software specification discovery," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 2, pp. 282–296, Feb. 2011.

[23] T. Y. Chen, F.-C. Kuo, and Z. Q. Zhou, "An effective testing method for end-user programmers," in *Proc. Workshop End-User Softw. Eng.*, 2005, pp. 21–25.

[24] A. J. Ko, R. Abraham, L. Beckwith, A. Blackwell, M. Burnett, M. Erwig, C. Scaffidi, J. Lawrance, H. Lieberman, B. Myers, M. B. Rosson, G. Rothermel, M. Shaw, and S. Wiedenbeck, "The state of the art in end-user software engineering," *ACM Comput. Surveys*, vol. 43, no. 3, pp. 21:1–21:44, 2011.

[25] T. Y. Chen, F.-C. Kuo, D. Towey, and Z. Q. Zhou, "Metamorphic testing: Applications and integration with other methods," in *Proc. 12th Int. Conf. Quality Softw.*, 2012, pp. 285–288.

[26] T. Y. Chen, D. H. Huang, T. H. Tse, and Z. Q. Zhou, "Case studies on the selection of useful relations in metamorphic testing," in *Proc. 4th Ibero-Am. Symp. Softw. Eng. Knowl. Eng.*, 2004, pp. 569–583.

[27] Z. Q. Zhou, T. H. Tse, F.-C. Kuo, and T. Y. Chen, "Automated functional testing of web search engines in the absence of an oracle," Dept. Comput. Sci., The Univ of Hong Kong, Hong Kong, Tech. Rep. TR-2007-06, 2007.

[28] Z. Q. Zhou, S. Zhang, M. Hagenbuchner, T. H. Tse, F.-C. Kuo, and T. Y. Chen, "Automated functional testing of online search services," *Softw. Testing, Verification Rel.*, vol. 22, no. 4, pp. 221–243, 2012.

[29] T. Imielinski and A. Signorini, "If you ask nicely, I will answer: Semantic search and today's search engines," in *Proc. IEEE Int. Conf. Semantic Comput.*, 2009, pp. 184–191.

[30] N. Kristof, "Boycott Microsoft Bing," *The New York Times*, Nov. 20, 2009.

[31] S. Denyer, "Microsoft denies censoring results in the 'uncensored' version of Chinese-language Bing," *The Washington Post*, Feb. 12, 2014.

[32] P. Carsten, "Microsoft denies global censorship of China-related searches," Reuters, Feb. 12, 2014.

[33] A. Sohn. (2009, Nov. 20). Committed to comprehensive results [Online]. Available: http://bit.ly/6CD49e

[34] S. Morasca, D. Taibi, and D. Tosi, " T-DOC: A tool for the automatic generation of testing documentation for OSS products," in *Open Source Software: New Horizons* (series. IFIP Advances in Information and Communication Technology), P. Ågerfalk, C. Boldyreff, J. M. González-Barahona, G. R. Madey, and J. Noll, Eds. Berlin, Germany: Springer-Verlag, 2010, vol. 319, pp. 200–213.

[35] Quora. Will .es domain hurt my search results [Online]. Available: http://www.quora.com/Will-es-domain-hurt-my-search-results, 2012.

[36] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *J. Mach. Learn. Res.*, vol. 4, pp. 933–969, 2003.

[37] M. A. Qader, S. Rahman, and M. Hasan, "Structured web search in small domain," in *Proc. 16th Int. Conf. Comput. Inf. Technol.*, 2013, pp. 426–431.

[38] S. Ieong, N. Mishra, E. Sadikov, and L. Zhang, "Domain bias in web search," in *Proc. 5th ACM Int. Conf. Web Search Data Mining*, 2012, pp. 413–422.

[39] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. Hillsdale, MI, USA: Hillsdale, 1988.

[40] T. R. Levine and C. R. Hullett, "Eta squared, partial eta squared, and misreporting of effect size in communication research," *Human Commun. Res.*, vol. 28, no. 4, pp. 612–625, 2002.

[41] A. Field, *Discovering Statistics Using IBM SPSS Statistics*, 4th ed. Newbury Park, CA, USA: Sage, 2013.

[42] J. Cohen, "Statistical power analysis," *Current Directions Psychol. Sci.*, vol. 1, no. 3, pp. 98–101, 1992.

[43] A. Alasiry, M. Levene, and A. Poulovassilis, "Extraction and evaluation of candidate named entities in search engine queries," in *Proc. 13th Int. Conf. Web Inf. Syst. Eng.*, 2012, pp. 483–496.

[44] J. R. Smarr, "Categorization by character-level models: Exploiting the sound symbolism of proper names," Master's thesis, The Symbolic Systems Program, Stanford Univ., Stanford, CA, USA, 2003

[45] *"Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—System and Software Quality Models,"* ISO/IEC 25010:2011, 2011.

[46] T. Y. Chen and R. Merkel, "An upper bound on software testing effectiveness," *ACM Trans. Softw. Eng. Meth.*, vol. 17, no. 3, pp. 16:1–16:27, 2008.

[47] A. Arcuri, M. Z. Iqbal, and L. Briand, "Random testing: Theoretical results and practical implications," *IEEE Trans. Softw Eng.*, vol. 38, no. 2, pp. 258–277, Mar./Apr. 2012.

[48] J. Guo, G. Xu, X. Cheng, and H. Li, "Named entity recognition in query," in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2009, pp. 267–274.

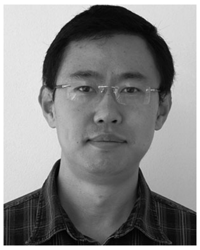
[49] G. Rossi, O. Pastor, D. Schwabe, and L. Olsina, *Web Engineering: Modelling and Implementing Web Applications*. New York, NY, USA: Springer-Verlag, 2008.

[50] D. Carmel and E. Yom-Tov, *Estimating the Query Difficulty for Information Retrieval*. San Mateo, CA, USA: Morgan, 2010.

[51] S. Wagner, *Software Product Quality Control*. New York, NY USA: Springer-Verlag, 2013.

[52] *IEEE Standards Association, IEEE Standard—Adoption of ISO/IEC 15026-2:2011, Systems and Software Engineering—Systems and Software Assurance—Part 2: Assurance Case, IEEE Std 15026-2^{TM}2011*, 2011.

[53] IEEE Project Number: P25010. *Standard—Adoption of ISO/IEC 25010-2010, Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation, (SQuaRE)—System and Software Quality Models*, 2011.



Zhi Quan Zhou received the BSc degree in computer science from Peking University, China, and the PhD degree in software engineering from The University of Hong Kong. He is currently a senior lecturer in software engineering at the University of Wollongong, Australia. His research interests include software testing, security testing, debugging, software maintenance, symbolic analysis, and quality assessment of Web search engines.

Shaowen Xiang received the BEng degree in computer science and technology from the Northeast Forestry University, China, and the master of computer science degree (in software engineering) from the University of Wollongong, Australia. His research interests include software testing and quality assessment of Web search engines. He is currently a software test analyst at Itree Pty Ltd, Australia.

Tsong Yueh Chen received the BSc and MPhil degrees from The University of Hong Kong, MSc degree and DIC from the Imperial College of London University, and the PhD degree from The University of Melbourne. He is a professor of software engineering at the Swinburne University of Technology, Australia. Prior to joining Swinburne, he taught at The University of Hong Kong and The University of Melbourne. His current research interests include software testing, fault localization, and program repair.