搜索引擎测试相关论文报告

MF1933058 刘凡维

本次报告主要介绍了 3 篇搜索引擎测试相关论文

- *If you ask nicely, I will answer: Semantic Search and Today's Search Engines*
- *Metamorphic Testing for Software Quality Assessment: A Study of Search Engines*
- *On Applying Metamorphic Testing: An Empirical Study On Academic Search Engines*

If you ask nicely, I will answer: Semantic Search and Today's Search Engines

Tomasz Imielinski , Alessio Signorini

2009 IEEE International Conference on Semantic Computing

1、研究背景及研究动机

目前的搜索引擎仍然对查询的构造方式非常敏感。在某些情况下，等价但略有不同的查询形式会导致完全不同的结果。只有一个正确答案的查询或者热门查询通常被搜索引擎很好地服务，并且搜索引擎通常会在前 10 个搜索结果中返回正确答案。

今天的搜索引擎经常会为同一个查询的变体返回不同的答案。根据查询中使用的关键字，这些答案可能是对的，也可能是错的。选择正确的关键字的负担留给了用户，用户常常需要多次构造查询以获得答案。 例如：biography of George Bush、bio of George Bush、find me bio of George Bush 这三个查询可被认为是相似的查询（布什的传记）。

2、语义不变性与不变性度量

实现语义不变性是任何一个以语义为目标的搜索引擎必须具备的条件。即：在语义搜索引擎中，两个语义等价的查询应该返回相同顺序的结果。当一个查询只有一个正确答案（ORA）时，由于我们只对答案的存在感兴趣，所以这个条件的限制性可能较小：如果搜索结果页（SERP）包含一个查询的答案，那么它还应该包含所有语义上等价的重新措辞的答案。在这种较弱的等价定义下，“法国首都”和“哪个城市是法国首都”的搜索结果页都应该包含“巴黎”的答案。

接下来将介绍一些不变性度量，这些度量将在实验部分中被使用。

1. ***Entropy*** 理想的语义搜索引擎应该总是以相同的顺序返回相同的结果集，以便进行等价的查询。搜索结果页（SERP）的稳定性可能是真正理解查询语义的最有意义的标志。为了测量结果的稳定性而计算了熵。
2. ***Top-K Results Overlap*** 给出两个语义上等价的查询 r1 和 r2，这个度量的目的是计算在它们的前 K 个结果中共享的 url 的比例。理想情况下，语义搜索引擎会为两个实例返回相同的结果，使重叠稳定到 100%。

3. ***ORA Invariance*** 给定一组语义上等价的查询 q1,q2,…,qn，此测试计算在结果页中出现正确答案的比例。由于查询语义的不变性，在任何真正的语义搜索引擎中，此测试的结果应该稳定到 100%。

3、实验方法及结果

实验比较了著名的搜索引擎， Google, Yahoo!, Live, Ask.com 以及 Hakia 和 Cuil 这两个声称具有语义的新搜索引擎。

3.1 Query 构造

手动生成实验中使用的 40000 多个查询将非常耗时而且非常主观。因此实验主要依赖从 AOL 的查询日志中提取的流行模板，这些模板通过从预先分类的 Wikipedia 页面中提取的实体进行参数化。

查询模式是由对象类参数化的查询模板，例如：“个人简历”、“国家主席”或“药物副作用”都是有效的查询模式。真正的查询可以通过使用相应对象类的任何元素替换相对应的参数来生成，如“bio of George *Bush*”, “President of *Spain*” ,“side effects of *Tylenol*”.。查询模式不限于自然语言问题，可以括任何类型的查询。

3.2 RQ1：对query进行过度描述是否会损害返回结果的相关性

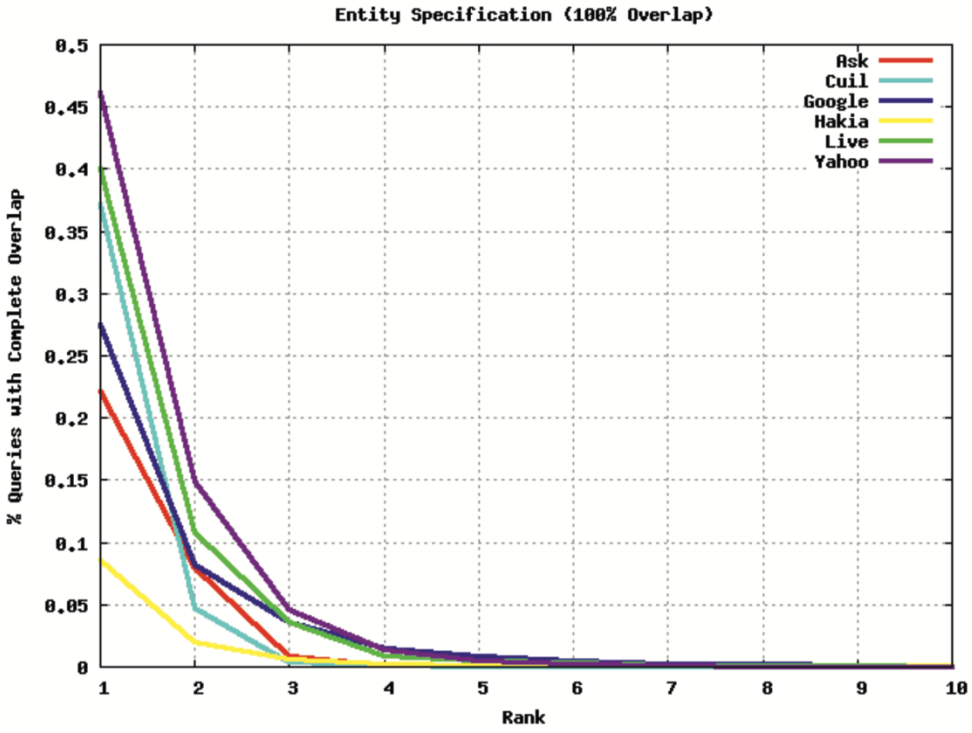
query 例子：IBM the company &IBM 、Tom Hanks the actor &Tom Hanks

实验将测试重点放在 9 个主要类别上：演员、艺术家、城市、公司、疾病、杂志、大联盟球员、政治家和总统，部分例子如下：

- Artists: Lyubov Popova, Antonio Zanchi, ...
- Companies: Iomega, Duracell, NetZero, Asus, ...
- Cities: Lakhva, Adahuesca, Fabletown, ...
- Diseases: Osteoarthritis, Diastrophic dysplasia, ...

接下来，将 query 的普通版本与扩展版本提交到 Google, Yahoo!, Live, Ask.com 以及 Hakia 和 Cuil。

下图总结了当使用类别信息过度指定查询时，在每个搜索引擎上返回的结果。



可以发现，只考虑返回第一个的结果时，排名第 1 的 Yahoo！只有 45%的查询对是相同的， LIve 结果中越有 40%，谷歌中有 27%， Ask.com 搜索结果中约有 22%。结果表明，搜索引擎大多不了解 Iomega 是一家公司，也不了解 Richard Dieberkorn 是一名艺术家。

这些多余术语的添加导致返回的 URL 中出现不同的页面集。重要的是要注意这个实验并不是为了评估这些结果之间的质量差异，而是试图指出那些事实上不应该有差异的结果是如何做到的。考虑到前三个结果，不增加冗余类别信息的查询对的比例平均小于 5%，如果考虑超过 4 个 url，则所有搜索引擎的比例几乎为零。

这表明搜索引擎对页面的标记和给定文本有很大的依赖性，不同页面和不同对象之间的敏感程度不同。

3.3 RQ2：搜索引擎对同义词使用的敏感性如何

第二个实验旨在测量搜索引擎对同义词使用的敏感性。然而，与其开发复杂的上下文感知替换算法来正确理解什么时候用另一个词替换一个词，实验决定专注于简单的数字音译。使用数字，而不是一般的同义词替换，可以简单但准确地生成大量的等价查询，以便在测试中使用。此外，选择这个测试是因为它非常简单，易于理解，并且很好地模拟了非常常见（和已知）的用户行为。

实验从最初的一组查询是从 AOL 的查询日志中提取 1500 多个以“top number”开头的查询，并用数字（例如，“top 20 cars”到“top 20 cars”）替换数字。这类查询的例子有“前 1000 个婴儿名字”、“有史以来的前 100 个游戏”、“前 20 名摇滚热门歌曲”和“前 10 的电动汽车品牌”。

表一和表二总结了实验结果：

	Ask	Cuil	Google	Hakia	Live	Yahoo
1	0.029	0	0.057	0.125	0.025	0.025
2	0.004	0	0.018	0.040	0.007	0.004
3	0.004	0	0.004	0.018	0.004	0
4	0	0	0	0.015	0.004	0
5	0	0	0	0.007	0.004	0
6	0	0.004	0	0.007	0.004	0
7	0	0	0	0	0.004	0
8	0	0	0	0	0.004	0
9	0	0	0	0	0.004	0
10	0	0	0	0	0.004	0

Table I
FRACTION OF PAIRS WITH 100% OVERLAP AT DIFFERENT K

	Ask	Cuil	Google	Hakia	Live	Yahoo
1	0.971	1.000	0.943	0.875	0.975	0.975
2	0.957	1.000	0.883	0.740	0.949	0.950
3	0.946	1.000	0.799	0.623	0.935	0.922
4	0.925	1.000	0.689	0.542	0.913	0.900
5	0.903	1.000	0.633	0.509	0.909	0.883
6	0.878	1.000	0.604	0.484	0.891	0.872
7	0.867	1.000	0.601	0.476	0.891	0.851
8	0.853	0.996	0.590	0.462	0.884	0.833
9	0.842	0.993	0.587	0.451	0.865	0.819
10	0.832	0.986	0.576	0.436	0.855	0.819

Table II
FRACTION OF PAIRS WITH 0% OVERLAP AT DIFFERENT K

所有的被测搜索引擎在这次测试中都表现得很差。平均而言，即使只考虑返回的最顶部的 URL，也只有 3%的查询对有完全重叠。Hakia 是一个例外，它展示了对这类查询的更好的理解（尽管还远未达到预期）。显然，一个理想的语义搜索引擎应该有 100%的重叠。

3.4 RQ3：评估同一 query 的不同重述返回的答案或结果的不变性如何

这组实验的目的是评估同一查询的不同重述返回的答案或结果的不变性。实验分为两部分：ORA不变性和稳定性。

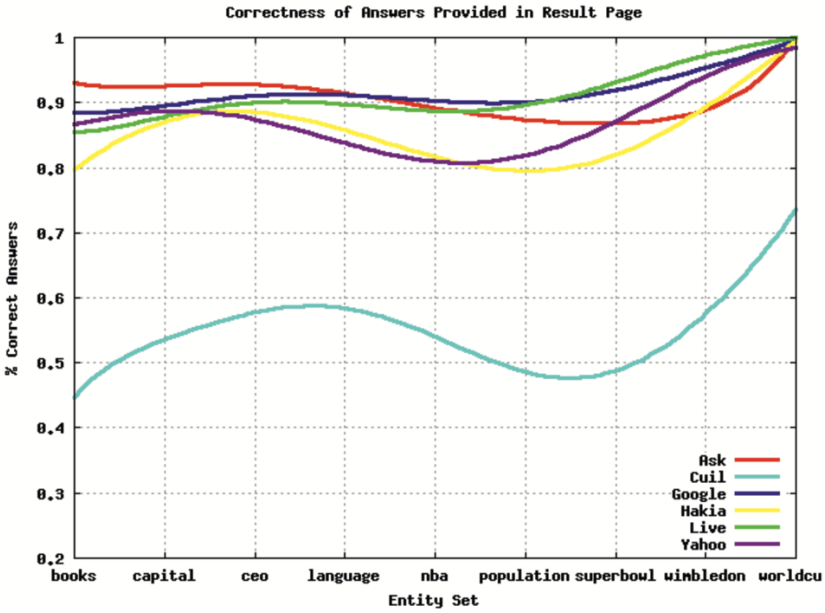
属于第一个类别的查询模式希望在结果页中找到答案。在这些实验中，我们使用了许多不同的措辞：28个代表“国家人口”，11个代表“公司首席执行官”，2个代表“书籍作者”，17个代表“超级碗年度冠军”，22个代表“温布尔登年度冠军”，17个代表“世界杯年度冠军”。这些模式已经在AOL的查询日志中获得并手动参数化。已在可公开访问的网站上手动找到正确答案。在寻找一个国家人口的准确性时，对搜索页面中的数字进行了一些标准化（例如，“750万”被转换为“7500000”），如果页面中至少有一个数字在预期值的正负10%范围内，则结果被认为是正确的。

在实验的第二部分中，实验目的在于测量由于对原始查询进行小修改而返回的结果之间的差异。例如，从AOL的查询日志中提取406个格式为“how to cook”的查询，手动检查它们的一致性，然后参数化“cook”、“make”和“prepare”之间的动词。同样，从维基百科上获得了100本有史以来最有影响力的书的列表，并使用该列表手动参数化了“谁是书的作者”和“谁写了书”的查询。最后，

从WebMD中获得了一个常见疾病列表，并从AOL的查询日志中收集了400多个包含“症状”或“符号”的等价查询，这些查询是使用这些疾病参数化的。

在这个测试中，对于提交给搜索引擎的每个查询都会在结果页面的任何地方查找答案。如果找到了答案，就被认为是积极的结果，否则，就是消极的结果。最后，对于每个ORA模式，实验测量了每个给定搜索引擎提供正确答案的改写短语的比例。

在这个不变性测试中，所有的搜索引擎（除了 Cuil）都表现得出奇的好。大多数搜索引擎似乎能够在 SERP 中为大约 90%的测试重述提供正确答案。



返回结果显示了即使搜索引擎对query缺乏真正的理解，使用流行主题（如世界杯或超级杯）的query也可以通过简单的关键字匹配来找到答案。

下图Figure 4显示了在不同年份的“超级碗年度冠军”的等价查询中返回正确答案的重述部分。

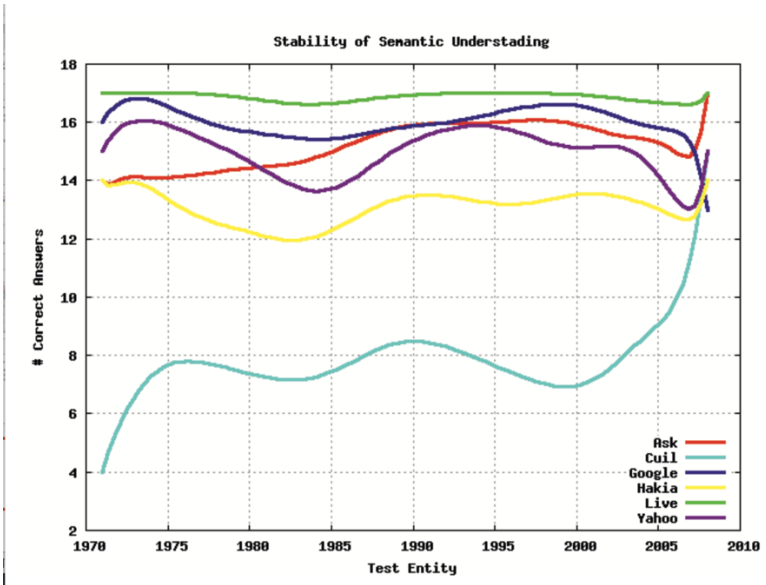


Figure 4. Stability of Semantic Understanding for Super Bowl dataset

如前所述，在一个真正的语义搜索引擎中，每个重述提供的答案的正确性最终不应取决于所选参数，假设引擎的数据库中有这样的知识。最后的实验结果表明，Live是最稳定的搜索引擎，Cuil是最不稳定的搜索引擎，对方案中使用的参数具有很高的敏感性。鉴于先前的实验证实Live不是语义的，结果的高稳定性再次归因于主题的流行以及主题周围内容和标记的冗余。

4、总结

实验提取并生成了多个查询模式的不同重述，测试了简单的同义词替换，并测量了添加冗余类别词时查询过度规范的影响。AOL 的查询日志和基于 Wikipedia 的实体列表已经被用来构造我们实验中使用的查询。本文总结的实验结果表明：

1. 一般查询的结果不变性仍然很差。如今的搜索引擎对查询的措辞非常敏感。它们大多是基于关键字的，与模拟人工查询理解相去甚远。
2. 只有一个正确答案（ORA）的查询似乎被搜索引擎服务得很好，搜索引擎设法在结果页面中返回正确答案，但是对查询的形式却出人意料地漠不关心。

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

Zhi Quan Zhou, Shaowen Xiang, and Tsong Yueh Chen
IEEE TRANSACTIONS ON SOFTWARE ENGINEERING 2016

1、研究背景及研究动机

大多数软件测试技术都假设 oracle 是可用的，测试人员可以根据 oracle 来验证测试用例执行结果的正确性。但是，在某些情况下，oracle 不可用或不可用，但使用成本太高，这种情况称为 oracle 问题，是软件测试的一个基本挑战。为了缓解 oracle 问题，开发了一种蜕变测试（Metamorphic Testing）方法。

搜索引擎的软件验证比传统验证活动要困难得多。用户只能在不了解其设计的情况下使用在线搜索服务。对于用户来说，搜索引擎代表了大量没有规范（或规范不完整）的软件产品。由于缺乏软件设计和规范细节方面的知识以及 oracle 问题，用户很难决定搜索引擎是否以及在多大程度上适合他们的特定信息需求。

2、蜕变测试与蜕变关系

蜕变测试通过蜕变关系（MR）进行测试来缓解 oracle 问题，蜕变关系是被测软件（SUT）的必要属性。MR 不同于其他类型的正确性属性，因为 MR 是目标程序的多个执行之间的关系。因此，即使没有 oracle 可用于验证每个单独的输出，我们仍然可以对照给定的 MR 检查 SUT 的多个输出。如果在某些测试用例中违反了蜕变关系，则会显示失败。

用户可以将 MR 定义为他们期望的“好”搜索引擎所具有的必要属性。蜕变关系在本文中被手动确定。为了将 MT 应用于搜索引擎的自动质量评估，使用了两组 MR：

- “不缺少网页”组，评估搜索引擎检索适当网页以满足用户需求的能力：MPSite、MPTitle、MPReverseJD
- “一致排名”组，评估搜索结果的排名质量：SwapJD、Top1Absent

3、实验设计与结果

3.1 MPSite

MPSite 主要关注搜索引擎在检索包含确切单词或短语的网页时的可靠性。因此，它评估基于关键字的搜索特性。

由于 MPSite 可以自动检测故障，因此可以在没有 oracle 的情况下用于可靠性评估。设计了一系列使用 MPSite 进行可靠性评估的实验，其中发布了两种类型的查询，即英文查询和中文查询，我们希望看到操作配置文件（语言）对可靠性的影响。

一次蜕变测试由一个源查询和一个后续查询组成，测试结果将是“MR 满足”或“MR 被违反”。违反 MPSite 意味着失败。对于 MPSite 的实证研究，测试持续运行 379 小时。在每小时结束时，每小时的测试结果根据小时 ROCOF（故障发生率）计算，其值范围为[0,1]。例如，假设从某一天的上午 1:00 到凌晨 2:00 检查了 1000 对有效源和后续响应，其中 30 对显示失败，那么这段时间的每小时 ROCOF 分数将为 0.03，并且这 1000 次蜕变测试形成一批测试。

查询词是从英汉词典中随机抽取单词，分别针对英汉两种语言的使用模式而形成的。因此，不同的批次使用不同的测试套件。换句话说，在不同的时间发出不同的查询集。

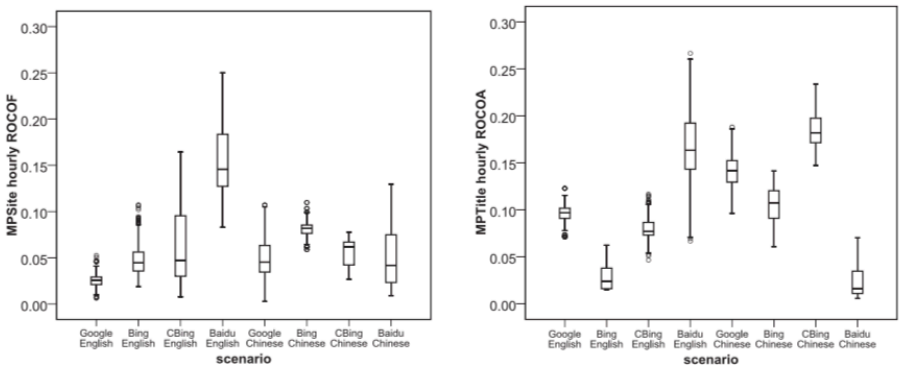
如前文所述，MPSite 关注的是搜索引擎在检索包含确切单词或短语的网页时的可靠性。ROCOF 值越低，表明搜索服务的可靠性越高。Figure9 (a)表明，所有场景的所有每小时 ROCOF 分数都在 0 以上。这意味着没有一个搜索引擎是完美的：每个使用模式下的搜索引擎每小时都会产生故障。一个有趣的观察是，不同的搜索引擎有非常不同的性能。相对而言，最可靠的服务是谷歌的英语搜索，而最不可靠的服务是百度的英语搜索，最糟糕的情况下，百度的每小时 ROCOF 高达 0.25 左右。然而，百度的中文搜索明显优于英文搜索，而谷歌和 Bing 的英文搜索则优于相应的中文搜索。这些发现似乎与以下事实一致：百度是一个基于中国的搜索引擎（因此可以认为百度主要是为中文查询而设计的，或是在中文查询方面比英文查询方面更训练有素），而谷歌和 Bing 是一个基于美国的引擎，有更多的英文查询。

3.2 MPTitle

MPTitle 关注搜索引擎理解和抽象网页以及理解用户意图的综合能力。MPTitle 的实验过程与 MPSite 非常相似，两组实验也在同一时间段内进行，只是 MPTitle 实验有 380（不是 379）小时的观察时间。与 MPSite 类似，Google English

和 Google Chinese 每小时检查约 1000 对源代码和后续响应，其他 6 个场景每小时检查约 3000 对。

Figure9(b) 给出了所有 8 种情况下每小时 ROCOA 得分分布的箱型图。结果表明，没有一个搜索引擎是完美的，因为他们每小时的 ROCOA 得分都在 0 以上：他们每小时都会产生失败或异常。查询语言对性能有很强的影响：Google English 和 Bing English 的性能继续优于 Google Chinese 和 Bing Chinese。CBing 英语的表现也优于 CBing 中文，这表明 CBing 不擅长中文搜索，尽管它是为中国用户设计的。另一方面，百度中文的表现继续优于百度英文。百度也继续被评为最好的中文引擎，但最差的英文引擎。



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

Fig. 9. Distributions of MPSite and MPTitle results.

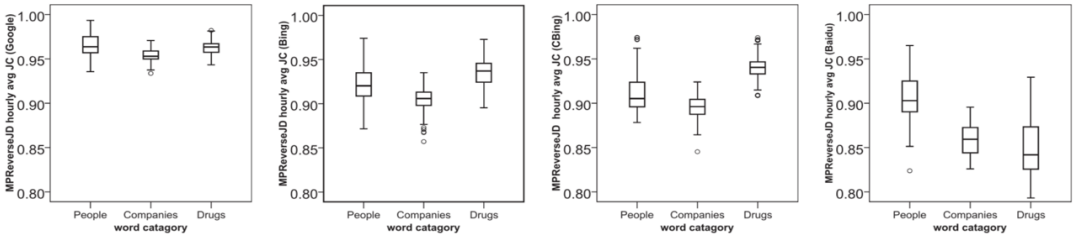
3.3 MPReverseJD

搜索引擎可能会返回不同的查询结果，这些查询的词序不同，特别是在顺序很重要的情况下，这可能会改变查询的含义。例如，“Sunday School”和“School Sunday”有不同的含义：前者是指通常隶属于教会的一个班级或学校，在星期日教给儿童圣经知识；而后者则没有这种含义。为了最小化单词顺序对查询含义的影响，MPReverseJD 只使用名称作为基本查询项。我们进一步设计了三种使用模式：第一种模式只使用人名，第二种模式只使用公司名，第三种模式只使用药品名。

MPReverseJD 评估了搜索引擎在基于关键字的搜索和语义搜索方面的能力。基于关键字的搜索被评估是因为使用了双引号；语义搜索也被评估是因为源查询和后续查询在语义上相似，因此应该产生相似的结果。

该实验的每个使用模式都有独立的名称列表，从中随机选择查询词。“人物”名单上有 200 位名人的名字。“公司”名单上有 200 家著名公司的名字。“药物”的名称列表包含 200 个随机选择的药物名称。每批测试的结果是每小时平均的 Jaccard 系数。

下图展示了所有 12 个方案的 MPReverseJD 小时平均 Jaccard 系数的分布的箱形图。较高的系数意味着网页检索的稳定性更好。对下图的可视分析显示，所有搜索引擎的所有小时得分都低于 1。这意味着没有一个搜索引擎在任何时候都是完美的。在这四个引擎中，谷歌最稳定，而百度最不稳定。使用模式（单词类别）似乎对搜索引擎的性能有影响。



(a) Google (b) Bing (c) CBing (d) Baidu

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

3.4 swapJD

与 MPReverseJD 类似，SwapJD 的设计还受到这样一个想法的启发，即如果搜索引擎对查询之间的非本质差异很健壮（不敏感），类似的查询应该有相似的结果。SwapJD 描述如下：源查询 A 只包含两个单词，后续查询 B 是通过交换这两个单词来构造的。如果这两个查询有相似的含义，不管它们的词序如何，一个稳定的搜索引擎应该为 A 和 B 返回相似的结果。

实验设计了三组名词：

第一组包含 20 个城市名称

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

第二组包含 7 个时间/日期

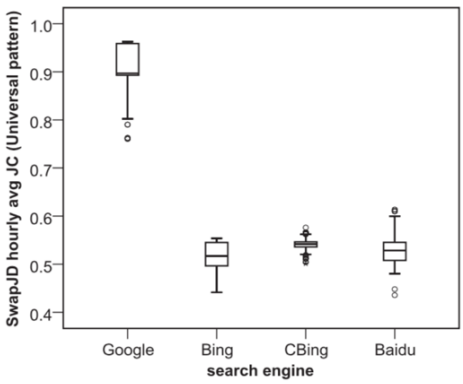
- Swhen = {afternoon, evening, midnight, morning, today, tomorrow, yesterday}

第三组包含 20 个名词

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

接下来构造以下形式的 queryA：”Swhere Swhen” “Swhen Swhat” “Swhere Swhat”（共 680 种组合方式）。将两个名词颠倒生成后续查询 B

结果的分布如下图所示，从图中可以看出，尽管每小时执行相同的 680 对测



试，但所有四个搜索引擎的性能随时间变化很大。所有的时均 Jaccard 系数都小于 1，也就是说，对于任何一个搜索引擎来说，稳定性质量都不是完美的。

同时，结果表示谷歌在排名稳定性上优于其他三个引擎。为了提高可用性（更具体地说，为了增强软件系统的可用性），用户在使用 Bing、CBing 或百度进行搜索时应注意词序，因为这些引擎对此比 Google 敏感得多。当用户对最初的搜索结果不满意时，这三个引擎的用户也可以考虑更改词序以再次进行查询。

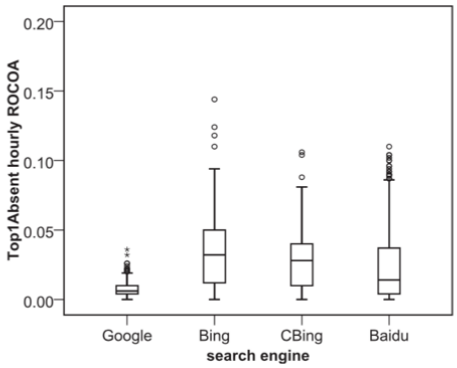
3.5 Top1Absent

Top1Absent 关注的是搜索结果显示的第一个结果的排名质量，这个排名第一的结果可以被认为是所有搜索结果中最重要的一个

Top1Absent 描述如下：源查询 A 是从英语词典中随机选择的单词，p1 是搜索引擎返回查询 A 的第一个网页。后续查询 B 仍然使用 A 作为查询项，但仅限于 p1 的域。预期的关系是 p1 应该仍然出现在 B 的搜索结果中。

在实验中，首先从一本英语词典中随机抽取500个不同的单词（不包括“of”等常用单词）作为源查询。这500个源查询及其动态构造的后续查询形成一批测试，每小时对每个被测试的搜索引擎执行一次。如前所述，对于相同的查询，搜索引擎可以在不同时间具有不同的行为，因此在不同时间使用相同的源查询集进行测试是有意义的。一批测试的结果是当前批次的ROCOA，其值在[0,1]范围内。

对于Top1Absent，四个搜索引擎同时测试了353个小时。每小时ROCOA得分的分布如下图所示：



尽管每小时使用相同的源代码查询（500个英文单词），但随着时间的推移，搜索引擎的性能变化很大。在最好的情况下，所有四个搜索引擎每小时的ROCOA都是0。最糟糕的情况是，谷歌、必应、CBing和百度的每小时ROCOA分别为3.6%、14.4%、10.6%和11.0%。ROCOA评分高意味着异常率高。

4、总结

实验结果表明，本次实验的方法对开发人员和用户都是有用的。首先，该方法可以有效地检测各种故障。第二，实验过程中发现操作配置文件对搜索质量有显著影响。对于给定的搜索引擎，不同的查询语言、不同类型的查询词和被搜索的不同域，其搜索质量可能会有很大的不同。这一发现为开发人员识别系统的优缺点提供了提示，也有助于用户选择合适的搜索引擎或更好地构造查

询。利用 MRs 自动检测故障和异常的能力也可以为运行时自校正机制的构建提供提示，这将是未来的研究课题。

类似于在突变分析中使用一组突变体来评估测试套件的有效性，一组正确选择的 MRs 可以潜在地用于评估某些软件质量特征。对这一课题的进一步研究将是今后的一个研究领域。

Metamorphic Relation Patterns for Query-Based Systems

Sergio Segura, Amador Durán, Javier Troya, and Antonio Ruiz-Cortés

2019 IEEE/ACM 4th International Workshop on Metamorphic Testing (MET)

1、研究背景及研究动机

蜕变测试已成功应用于不同类型的基于查询的系统中，包括 Google 和 Bing 等搜索引擎、Spotify 和 YouTube 等 RESTful api、亚马逊和沃尔玛等电子商务网站以及 NASA 的数据访问工具包等数据存储库，以缓解 oracle 的问题。

使用蜕变关系模式（MRP）有两个主要好处。首先，它们对确定蜕变关系非常有帮助。因为模式能够指导测试员寻找具有一定结构的蜕变关系关系，使得识别关系比从头开始时容易得多。第二，模式的识别是实现蜕变关系自动推断的关键点。

2、主要工作

本文提出了七个 MRP，以帮助测试人员识别和推断 QBS 中的蜕变关系。文中提出的模式是基于作者以前的工作，也基于 QBS 蜕变测试相关文献中常见的蜕变关系。为了说明该方法的可行性，文章展示了所提出的模式如何帮助识别三个拥有数百万用户的真实 QBS 中的变形关系：电子商务平台 PrestaShop、web 电子邮件服务 Gmail 和 HBO 的视频流应用程序。

3、蜕变关系模式

Input equivalence 此模式表示源测试用例和后续测试用例等价的关系，因此它们的输出应该包含相同顺序的相同项，即它们应该相等。 例如，在 YouTube 中指定 no order 的搜索应该产生与指定默认排序标准完全相同的结果，默认排序标准基于与搜索查询的相关性。

Shuffling 此模式表示：源输出和后续输出应包含相同的项，而不考虑它们之间排序的问题。例如，例如，在 Booking.com 中搜索“hotels in London”应返回相同的结果，而不考虑指定的订购标准（价格、审核分数、开始时间等）。

Conjunctive conditions 这个模式表示查询以迭代的方式细化，不断添加新的条件，这样每个测试用例的结果都应该包含在前一个测试用例的结果中。例如，假设我们搜索“宠物”的 YouTube 视频。接下来，我们搜索 3D 的“宠物”视频，最后搜索 2018 年后上传的 3D“宠物”视频。直观地说，第三次搜索的结果应该是第二次搜索结果集的一个子集，同时，这两次搜索结果也是原始搜索结果的一个子集。

Disjunctive conditions 与前一个模式类似，但查询通过输入析取条件不断泛化，这样每个测试用例的结果都应该是后续测试结果的子集。 例如，在某 ASE

查询“Metamorphic”，接下来将搜索范围扩展到“Metamorphic”or “Test” ,显然第一次的搜索结果应该是第二次搜索结果的子集。

Disjoint partitions 此模式表示那些后续测试用例的输出应该是不相交的关系（即，它们不应该有共同的项）。例如，假设在 PayPal 用户帐户中搜索状态为“已完成”的退款。接下来，让我们假设在同一个帐户中执行一个新的搜索，搜索状态为“已取消”的退款。两个搜索的结果集不应具有任何共同项。

Complete partitions 此模式与前一个模式相关，它表示后续输出的并集应包含与源输出相同的项的那些关系。例如，YouTube 视频按其持续时间分为短视频（少于 4 分钟）、中视频（介于 4 到 20 分钟）和长视频（长于 20 分钟）。考虑一个源代码测试用例，它包含搜索带有关键字“testing”的 YouTube 视频。假设通过搜索相同的关键字来构造三个后续测试用例，分别将搜索限制为短视频、中视频和长视频。直观地说，后续测试输出（短视频、中视频和长视频）的并集应该包含与源测试输出相同的视频 。

Partition difference 此模式派生自前两个模式，表示后续测试用例的输出结果不相交且它们的并集包含与源输出相同的项的关系。例如，在前面的例子中，YouTube 上所有关于“测试”的视频和关于该主题的长视频之间的区别应该等于中短视频的结合。

4、总结

本文提出了 7 种蜕变关系模式来帮助识别 QBS 中的蜕变关系。为展示提出的模式具有概括性，作者利用它们在三个不同领域中派生出一些可行的蜕变关系。

未来的工作可能包括识别新的模式以及使用它们来测试特定的 QBS。更重要的是，文章预见了一条富有成效的研究路线，即利用机器学习等技术，利用所提出的模式自动推断出 QBS 中可能的蜕变关系。