# On the Vocabulary Agreement in Software Issue Descriptions

Oscar Chaparro, Juan Manuel Florez, Andrian Marcus
University of Texas at Dallas, Richardson, TX, USA
{ojchaparroa, jflorez, amarcus}@utdallas.edu

***Abstract*—Many software comprehension tasks depend on how stakeholders textually describe their problems. These textual descriptions are leveraged by Text Retrieval (TR)-based solutions to more than 20 software engineering tasks, such as duplicate issue detection. The common assumption of such methods is that text describing the same issue in multiple places will have a common vocabulary. This paper presents an empirical study aimed at verifying this assumption and discusses the impact of the common vocabulary on duplicate issue detection. The study investigated 13K+ pairs of duplicate bug reports and Stack Overflow (SO) questions. We found that on average, more than 12.2% of the duplicate pairs do not have common terms. The other duplicate issue descriptions share, on average, 30% of their vocabulary. The good news is that these duplicates have significantly more terms in common than the non-duplicates. We also found that the difference between the lexical agreement of duplicate and non-duplicate pairs is a good predictor for the performance of TR-based duplicate detection.**

## I. Introduction

Developers spend most of their time reading and understanding code and other software artifacts [1]. Comprehension of natural language software artifacts is hindered when stakeholders use different vocabulary to describe the same issues. This problem is known as the vocabulary mismatch problem [2] and it impacts many forms of human communication. For example, Furnas *et al.* [2] determined that people choose the same words to refer to the same object with less than 20% probability. More recently, Copeck *et al.* [3] determined that there is 24% average vocabulary agreement between two summaries of the same document produced by different individuals. The question is whether software engineering stakeholders exhibit similar levels of (dis)agreement when describing the same issues.

Answering this question has implications on the applications of Text Retrieval (TR)-based techniques, which have been used to support more than 20 software engineering tasks, ranging from duplicate bug detection to traceability link recovery [4]. The use of TR in duplicate bug detection and other software engineering tasks is based on the implicit assumption that there is common vocabulary among documents (*e.g.,* bug reports). There is some evidence to support this assumption, at least in the context of tracing bug reports to code. For example, Moreno *et al.* [5] measured the vocabulary agreement between bug reports and source code and found that 80% of the faulty classes share vocabulary with bug reports and the overall average agreement is 24%.

Our goal is to further study how different software stakeholders describe the same issues, within the same artifact types, by measuring the vocabulary agreement between duplicate issues. In this paper, we report the results of an empirical study we conducted to investigate to what degree duplicate issue descriptions use the same vocabulary, and the impact of the vocabulary agreement on TR-based duplicate detection. The importance of duplicate issue descriptions lies on the body of research in duplicate issue detection, aiming at minimizing developers' effort on bug triaging [6], [7], [8] and exploiting additional information from duplicates to help developers in such task [9]. Our work has implications on understanding the scope of TR in duplicate issue detection [6], [8], [10].

Before describing our empirical study, we briefly discuss the work related to our research.

## II. Related Work

Researchers worked on analyzing the vocabulary of distinct textual artifacts in software comprehension and other fields, such as document summarization.

In software maintenance, researchers have analyzed the textual characteristics of bug reports. For example, Ko *et al.* [11] performed a linguistic analysis of bug report titles and found that distinct types of words (*e.g.*, verbs), play a particular role in conveying the problem described in titles. With a different focus, Sureka *et al.* [12] analyzed the part-of-speech and distribution of the words contained in issue titles to find vocabulary patterns that would help predict the severity level of bug reports. Bettenburg *et al.* [9] showed improvement on automatic bug triaging when using additional information that duplicate bug reports add to their master documents.

Closer to our study, Moreno *et al.* [5] measured 24% lexical agreement between source code (*i.e.*, classes) and bug reports. Haiduc *et al.* [13] quantified the use of domain terms in the source code of a set of graph libraries. The results suggest that, on average, a pair of systems share more than 63% of their domain source code lexicon.

Outside software engineering, the analysis of vocabulary, lexical choice or vocabulary agreement, have been studied in human-computer interaction and document summarization. For instance, Furnas *et al.* [2] reported evidence that different people rarely agree on words to refer to the same object, *i.e.*, in less than 20% of the times. Malt *et al.* [14] further showed that people use substantially different naming patterns for a set of common containers. Reiter *et al.* [15] provided

Table I: Overall statistics about the data sets.

| Statistic | Bug Reports | SO Questions |
|---|---|---|
| # of individual issue descriptions | 359,581 | 355,813 |
| # of duplicate pairs | 8,572 | 5,058 |
| # of non-duplicate pairs | 785,867 | 54,450 |
| # individual issues in dupl. pairs | 15,519 | 8,611 |
| # individual issues in non-dupl. pairs | 15,815 | 1,100 |

evidence that people associate different particular meanings with words and phrases, depending on the context in which they can be used. In in the context of summary evaluation, Copeck *et al.* [3] studied to what degree different summaries use the same vocabulary and concluded that there is 24% avg. agreement between summaries when stemming is used and case sensitivity ignored. Teufel *et al.* [16] found between 70% and 80% agreement when humans define expressions containing the same information but not necessarily following the same wording.

Our research is primarily motivated by this prior work, while we further study how different users describe the same issues, by measuring the lexical agreement of duplicate issue descriptions (including bug reports and Stack Overflow questions).

## III. Empirical Study

We performed an empirical study using data extracted from past bug reports and Stack Overflow (SO) questions.

### A. Definition and Context

The *goal* of our study is twofold: determine the vocabulary agreement of issues describing the same problem and study the impact of such agreement on TR-based duplicate detection.

The *context* of our study is represented by 13,630 duplicate pairs of issue descriptions from a set of systems developed in Java (see Table I). From these, 8,572 pairs correspond to duplicate bug reports extracted from the Jira bug repositories of 159 Apache projects, and 5,058 pairs correspond to duplicate SO questions referencing eleven open source technologies extracted from the StORMeD dataset [17]. By selecting exclusively Java projects, we could use the Java island parser developed by Ponzanelli *et al.* [17] to ensure that undesired content is removed from the issue descriptions (*e.g.*, code snippets and stack traces). We provide details of this filtering process below.

In Jira and Stack Overflow, users can mark an issue description (*i.e.*, bug report or SO question) as duplicate when they consider that it refers to the same problem described in a past issue[1]. In this case, the marked issue references the issue being duplicated (*i.e.*, the master issue). In our study, we consider the pair of issue descriptions $(d_1, d_2)$ as a duplicate pair, if $d_1$ is an issue marked as duplicate of $d_2$. When we found that two duplicate issues reference each other, *i.e.*, the pairs $(d_1, d_2)$ and $(d_2, d_1)$ can be extracted, we discarded one of the pairs, using the issue IDs, which reflect their chronological order. We keep the pairs that have the first issue posted earlier than the second issue.

[1]From now on, we refer to an issue description simply as *issue.*

As our goal is to measure the vocabulary agreement of the Natural Language (NL) content in issue descriptions, we remove non-NL information such as stack traces, code snippets, and code references. We normalize the textual data in issue descriptions to accurately measure the vocabulary agreement. First, we remove the undesired content previously mentioned using island parsing [17]. Second, we normalize the remaining textual content using standard techniques [3], [5], [13]. In particular, we tokenize and lemmatize the text. We remove punctuation, tokens with special characters, and tokens with less than three characters. Finally, we perform stop word removal, using the adapted the stop word list provided by the Lemur project [18].

The complete list of projects, duplicate data set, and stop words are available in our online replication package [19].

### B. Research Questions

Considering that issue descriptions contain different vocabulary sources (*i.e.*, title and description), we formulate the following research questions:

**RQ1**: *What is the vocabulary agreement between duplicate issue descriptions and between non-duplicates?*
**RQ2**: *Does the vocabulary source impact the vocabulary agreement of duplicate/non-duplicate issue descriptions?*
**RQ3**: *Is the vocabulary agreement between duplicate issue descriptions different than the one between non-duplicates?*
**RQ4**: *What is the implication of vocabulary agreement on text-based automated duplicate issue detection?*

### C. Methodology

We built the corpus of duplicate pairs for bug reports and SO questions, using 3 vocabulary sources: *title*, *description*, and *title+description*. We performed the preprocessing procedure described in section III-A, considering unique terms only.

We also built a corpus of non-duplicate issue pairs (see Table I). As it is more likely to have much more non-duplicates than duplicates, we performed sampling to create a more manageable non-duplicate corpus. For this, we randomly sampled 100 bug reports and 100 SO questions not marked as duplicates or referenced by duplicates from each project. These questions were then used to generate 4,950 pairs (*i.e.*, the number of 2-combinations of 100 distinct elements).

We rely on the *Lexical Agreement* (LA) metric used by Haiduc *et al.* [13] to measure the vocabulary agreement between two issue descriptions, formally defined as:

$$LA(d_i, d_j) = \frac{|V_i \cap V_j|}{(|V_i| + |V_j|)/2} \quad (1)$$

where, $V_i$ and $V_j$ are the sets of unique terms of issues $d_i$ and $d_j$, respectively. As the vocabulary sizes (*i.e.*, $|V_k|$) of the issues in a pair do not highly differ (see our replication package for the actual number of terms [19]), neither does the average vocabulary size in the denominator. Therefore, we can use LA as a fair estimator of the proportion of shared vocabulary between individual issues.

Table II: Number of pairs with no shared vocabulary, number of shared terms, and overall vocabulary agreement for all duplicate and non-duplicate pairs.

| Vocabulary Source | # of pairs with no shared vocab. [a] | | # of shared terms [b] | | Overall Lexical Agreement [b] | |
|---|---|---|---|---|---|---|
| | Duplicates | Non-duplicates | Duplicates | Non-duplicates | Duplicates | Non-duplicates |
| Title | 2,877 (33.6%) | 700,864 (89.2%) | 1.4 (1) | 0.1 (0) | 28.3% (22.2%) | 2.4% (0.0%) |
| Description | 1,627 (19.0%) | 431,997 (55.0%) | 5.2 (3) | 1 (0) | 20.3% (14.8%) | 3.8% (0.0%) |
| Title + Description | 436 (5.1%) | 356,700 (45.4%) | 6.3 (4) | 1.2 (1) | 25.0% (19.1%) | 4.7% (3.3%) |

*a.* in parenthesis, percentage values, *b.* average values and, in parenthesis, median values

(a) Bug reports

| Vocabulary Source | # of pairs with no shared vocab. [a] | | # of shared terms [b] | | Overall Lexical Agreement [b] | |
|---|---|---|---|---|---|---|
| | Duplicates | Non-duplicates | Duplicates | Non-duplicates | Duplicates | Non-duplicates |
| Title | 1,375 (27.2%) | 37,215 (68.3%) | 1.6 (1) | 0.4 (0) | 32.2% (28.6%) | 6.7% (0.0%) |
| Description | 332 (6.6%) | 6,283 (11.5%) | 4.9 (4) | 3 (3) | 20.4% (17.5%) | 9.4% (9.0%) |
| Title + Description | 140 (2.8%) | 4,030 (7.4%) | 5.8 (5) | 3.4 (3) | 23.3% (20.7%) | 10.2% (9.8%) |

*a.* in parenthesis, percentage values, *b.* average values and, in parenthesis, median values

(b) SO questions

For answering RQ1, 2, and 3, we compute LA on duplicate and non-duplicate pairs for each vocabulary source and summarize LA for both bug reports and SO questions using the average and median statistics. For answering RQ4, we implemented a traditional TR-based duplicate issue detector, which uses the classic document retrieval process [10]. Given a duplicate pair of issues, $(d_1, d_2)$, we use the vocabulary of $d_1$ to query the document space built from all the issue descriptions (except $d_1$) of the corresponding software project, in order to retrieve $d_2$. The algorithm will return a ranked list of potential duplicate issues. To compute the textual similarity, we rely on Lucene [20], which combines the Vector Space Model and Boolean model.

We use *effectiveness* and HIT@*N* to measure the accuracy of the TR-based duplicate detection. *Effectiveness* is defined as best rank obtained (*i.e.*, the rank of the duplicate issue $d_2$) by the set retrieved issues to a particular issue (*i.e.*, the issue $d_1$, used as a query). The lower the *effectiveness* value is, the more accurate the TR-based duplicate detection technique is. Hit@*N* is the number of issues with the corresponding duplicate issue retrieved in the top N results. The percentage of HIT@*N* is equivalent to the metric used in previous duplicate detection research [10], [7], *i.e.*, *recall rate*. As in previous research, we report *effectiveness* and HIT@{1,5,10} [10], [7].

### D. Results and Discussion

We present and discuss the results of the vocabulary agreement measurement and then the relationship of the measurement with the performance of TR-based duplicate detection.

*1) Vocabulary Agreement Measurement:* Table II summarizes the overall vocabulary agreement for bug reports and SO question pairs, based on each vocabulary source.

For the bug reports dataset, 33.6% pairs that rely on *titles* do not share any vocabulary. This proportion is lower for pairs extracted from *descriptions* and *titles+descriptions* (*i.e.*, 19% and 5.1%, respectively). Somewhat similar results were obtained for SO question pairs. The lower number of duplicate pairs without shared vocabulary, when both vocabulary sources are used, indicates that the issues in a pair share a significant part of the vocabulary between the *title* of one issue and the *description* of the other one. It is interesting to note that the number of pairs sharing vocabulary is somewhat higher for SO questions than for bug reports. Also, as expected, the average number of pairs with no common vocabulary is higher for non-duplicates than for the duplicate issues. On average (across vocabulary sources), 63.2% pairs of non-duplicate bug reports and 29.1% pairs of non-duplicate SO questions, do not share terms. This time, the number of non-duplicate pairs sharing vocabulary is significantly higher for SO questions than for bug reports, which means that it is more likely to find non-duplicate SO questions than non-duplicate bug reports with common words. Summarizing across vocabulary sources, we observe that:

> On average, 19.2% of duplicate bug report pairs and 12.2% of duplicate SO question pairs do not share vocabulary.

Table III summarizes the lexical agreement measurements for the subset of duplicate pairs that share vocabulary. On average, duplicate bug reports share between 24.7% and 42.5% of their vocabulary, and duplicate SO questions share between 21.8% and 44.2% of their terms, depending on the vocabulary source. This agreement level is slightly higher than the 24% reported by Copeck *et al.* [3]. In the case of non-duplicates, the

Table III: Number of shared terms, and vocabulary agreement for the duplicate and non-duplicate pairs that actually share vocabulary.

| Vocabulary Source | # of shared terms | | Lexical Agreement | |
|---|---|---|---|---|
| | Duplicates | Non-duplicates | Duplicates | Non-duplicates |
| Title | 2.2 (2) | 1.1 (1) | 42.5% (33.3%) | 22.3% (20%) |
| Description | 6.4 (4) | 2.1 (2) | 24.7% (17.7%) | 8.4% (7.1%) |
| Title + Descr. | 6.6 (5) | 2.2 (2) | 26.3% (20%) | 8.6% (7.1%) |

average values and, in parenthesis, median values

(a) Bug reports

| Vocabulary Source | # of shared terms | | Lexical Agreement | |
|---|---|---|---|---|
| | Duplicates | Non-duplicates | Duplicates | Non-duplicates |
| Title | 2.1 (2) | 1.1 (1) | 44.2% (40%) | 21.2% (20.0%) |
| Description | 5.2 (4) | 3.4 (3) | 21.8% (18.5%) | 10.6% (6.5%) |
| Title + Descr. | 6 (5) | 3.7 (3) | 24.0% (21.2%) | 11.1% (10.3%) |

average values and, in parenthesis, median values

(b) SO questions

Table IV: *Effectiveness* and HIT@{1,5,10} achieved by Lucene on finding the duplicate issues.

| Vocabulary Src. | Effect.[b] | HIT@1[a] | HIT@5[a] | HIT@10[a] |
|---|---|---|---|---|
| Title | 100 (11) | 1,394 (24.5%) | 2,352 (41.3%) | 2,817 (49.4%) |
| Description | 377.7 (30) | 1,317 (18.9%) | 2,246 (32.3%) | 2,732 (39.3%) |
| Title + Descr. | 289 (14) | 1,779 (21.9%) | 3,164 (38.9%) | 3,774 (46.4%) |

*a.* in parenthesis, percentage values, *b.* avg. values and, in parenthesis, median values

(a) Bug reports

| Vocabulary Src. | Effect.[b] | HIT@1[a] | HIT@5[a] | HIT@10[a] |
|---|---|---|---|---|
| Title | 1,071 (54) | 567 (15.4%) | 1,045 (28.4%) | 1,271 (34.5%) |
| Description | 6,147 (499) | 479 (10.1%) | 777 (16.4%) | 913 (19.3%) |
| Title + Descr. | 4,621 (134) | 645 (13.1%) | 1,106 (22.5%) | 1,367 (27.8%) |

*a.* in parenthesis, percentage values, *b.* avg. values and, in parenthesis, median values

(b) SO questions

Table V: Difference between the overall *Lexical Agreement* of duplicate and non-duplicate pairs. The difference is computed from the average values.

| Vocabulary Source | Bug reports | SO questions |
|---|---|---|
| Title | 25.9% | 25.5% |
| Description | 16.5% | 11.0% |
| Title + Description | 20.2% | 13.1% |

agreement is significantly lower. Bug reports share between 8.6% and 22.3% of their vocabulary, and SO questions between 11.1% and 21.2%, depending on the vocabulary source. Summarizing across vocabulary sources, we answer **RQ1** as follows:

> On average, *duplicate issues* share 31.1% and 30% of their vocabulary, for bug reports and SO questions, respectively. In addition, *non-duplicate issues* share 13.1% and 14.3% of their vocabulary, for bug reports and SO questions, respectively.

Comparing the agreement levels across vocabulary sources, Table III reveals that the issues relying on *titles* only are the ones that have the highest vocabulary agreement, whereas issues composed by only *descriptions* are the ones with the lowest agreement. The lexical agreement of *descriptions* and *title+descriptions* is comparable. *Titles* share significantly higher proportion of terms than the other vocabulary sources, *i.e.*, +16% and +10% more vocabulary for duplicates and non-duplicates, respectively. To illustrate these numbers, for duplicates bug reports and SO questions, *titles* have, on average, five terms, which is much lower, compared to the other vocabulary sources (24 terms for *descriptions* and 26 for *title+descriptions*). If we analyze the number of shared terms (see Table III), we note that, on average, software stakeholders agree on 2.2, 5.8, and 6.3 terms when they use the *title*, *description*, and *title+description*, respectively, to describe their problems. The question is whether these terms are sufficient to find duplicates, especially when using only *titles*. We answer **RQ2** as follows:

> The *title* of the issues is the vocabulary source that has the highest percentage vocabulary agreement between duplicates/non-duplicate issue descriptions.

We compared the agreement of duplicates and non-duplicate pairs. Table III shows that for each vocabulary source, the vocabulary agreement is higher for duplicates than for non-duplicates. Our data analysis found that the agreement of duplicates is statistically significant higher than the agreement of non-duplicates (Mann-Whitney test, $p-\text{value} < 0.05$), for each vocabulary source. For each case, the average magnitude of such difference is *large*, according to the corresponding Cliff's delta [21] (see our replication package [19] for the specific $d$ estimators). Based on the results, we answer **RQ3** as follows:

> Duplicate issue descriptions have statistically significant more terms in common than non-duplicate issues. On average, duplicate issues have 18% and 15.7% more common terms than non-duplicates, for bug descriptions and SO questions, respectively.

*2) Duplicate Detection Impact:* We report the results of our study regarding the impact of vocabulary agreement on TR-based duplicate detection.

Table IV reports the retrieval accuracy using Lucene. The *effectiveness* values exhibit the same trend as the agreement levels reported in Table III. In particular, the issues *title* is the vocabulary source that achieves the highest accuracy and has the highest vocabulary agreement, followed by *titles+descriptions*, and then by *descriptions* only.

Note that the duplicate detector achieves the highest performance when only the *titles* are used, even though, they only share 2.2 terms, on average. We argue that stakeholders tend to include key terms in the *titles* to briefly summarize the underlying problem that they are reporting, leaving specific contextual details of the problem to the *description*. We assume that these words are essential to retrieval. This assumption will be studied in future work.

The duplicate detection performance is significantly lower for SO questions than for bug descriptions, for all the accuracy measures (see Table IV). We conjecture that the lower detection performance of the SO questions occurs because the lexical agreement between non-duplicates is higher for SO questions than for bug reports. The results from Table II support our hypothesis. On average (summarizing across vocabulary sources), the overall lexical agreement between non-duplicate SO questions is 8.8%, which is higher than the one between non-duplicate bug reports, *i.e.*, 3.6% average agreement.

However, such agreement levels do not explain the detection performance trend across vocabulary sources, as using only *descriptions* leads to the poorest accuracy, even though this vocabulary source does not present the lowest lexical agreement for non-duplicates. Intuitively, if the agreement of duplicates is as high as the agreement between non-duplicates, we would expect low retrieval performance, as the underlying TR algorithm would not be able to distinguish between duplicates and non-duplicates. The results shown in Table V support this hypothesis. The lexical agreement difference between duplicates and non-duplicates is higher for bug reports than for SO questions, for each vocabulary source. If we take a closer look to Table II, we can explain the meaning of such

differences in terms of the actual number of shared terms. For *title+description* of SO questions, non-duplicates share 3.4 terms, which is close to the number of terms shared by duplicate pairs, *i.e.*, 5.8 terms, on average. In contrast, the number of terms shared by non-duplicate bug reports is 1.2, which is a low value, compared to the number of terms shared by duplicates, *i.e.*, 6.3 terms. Such difference makes it hard for the TR algorithm to detect duplicate SO questions. Note that in TR-based retrieval the actual number of shared terms is more important than the percentage of shared terms. Therefore, we answer **RQ4**:

> The lower the difference between the overall lexical agreement of duplicate and non-duplicate pairs is, the harder it is to detect duplicate issues for a TR-based technique.

## IV. Conclusions and Future Work

Our empirical study, conducted with 13K+ pairs of duplicate bug reports and Stack Overflow questions, reveals that developers and other software stakeholders tend to use rather different vocabulary when describing the same problem. Our lexical agreement analysis reveals that 19.2% of the duplicate bug descriptions and 12.2% of the duplicate SO questions do not have any common terms. However, the duplicate issues with common terms share 31.1% and 30% of their vocabulary, for bug reports and SO questions, respectively, when using both the issue title and/or description. These numbers are slightly higher than the agreement levels reported by existing research outside software engineering [3] (*i.e.*, 24%). Nonetheless, the vocabulary agreement is low when reporting issues. We expected a somewhat higher level of agreement, given that these issues are within a specific domain (*i.e.*, software engineering - java programming). Our study did not help revealing the reasons behind the low overall agreement. We conjecture that the use of synonyms could be responsible for this disagreement. This is an issue to be further studied in future work.

Fortunately, we found that duplicate issues have significantly more terms in common than the non-duplicates. On average, duplicate issues have 18% and 15.7% more terms than non-duplicates, for bug descriptions and SO questions, respectively. The difference is even higher when we combine pairs that do and do not share words (20.9% and 16.5%).

These findings have an impact on TR-based duplicate detection. First, the low agreement between duplicates negatively impacts the detection accuracy. We found that the performance of a traditional TR-based duplicate detector is rather low: in the (average) best case, the correct duplicate issue is retrieved in position 100 (11 median position). Second, we found that the difference between the overall lexical agreement of duplicate and non-duplicate pairs is a good predictor for the performance of TR-based duplicate detectors. In particular, the lower this difference is, the harder it is to detect duplicate issues. This means that one can use the properties of the corpus built with past issue descriptions to determine what strategy to use for duplicate detection.

## Acknowledgments

This research was supported in part by the following NSF grants: CCF-0845706 and CCF-1526118.

## References

[1] J. Sillito, G. C. Murphy, and K. De Volder, "Asking and Answering Questions during a Programming Change Task," *Transactions on Software Engineering*, vol. 34, no. 4, pp. 434–451, 2008.

[2] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais, "The vocabulary problem in human-system communication," *Communications of the ACM*, vol. 30, no. 11, pp. 964–971, 1987.

[3] T. Copeck and S. Szpakowicz, "Vocabulary agreement among model summaries and source documents," in *Proceedings of the Document Understanding Conference (DUC)*, 2004.

[4] V. Arnaoudova, S. Haiduc, A. Marcus, and G. Antoniol, "The use of text retrieval and natural language processing in software engineering," in *Companion Proceedings of the 38th International Conference on Software Engineering (ICSE'16)*, pp. 898–899, ACM, 2016.

[5] L. Moreno, W. Bandara, S. Haiduc, and A. Marcus, "On the relationship between the vocabulary of bug reports and source code," in *Proceedings of the 2013 International Conference on Software Maintenance (ICSM'13)*, pp. 452–455, IEEE, 2013.

[6] J. Anvik, L. Hiew, and G. C. Murphy, "Coping with an open bug repository," in *Proceedings of the 2005 OOPSLA workshop on Eclipse Technology eXchange (ETX'05)*, pp. 35–39, ACM, 2005.

[7] J. Lerch and M. Mezini, "Finding duplicates of your yet unwritten bug report," in *Proceedings of the 17th European Conference on Software Maintenance and Reengineering (CSMR'13)*, pp. 69–78, IEEE, 2013.

[8] C. Sun, D. Lo, S.-C. Khoo, and J. Jiang, "Towards more accurate retrieval of duplicate bug reports," in *Proceedings of the 26th International Conference on Automated Software Engineering (ASE'11)*, pp. 253–262, IEEE Computer Society, 2011.

[9] N. Bettenburg, R. Premraj, T. Zimmermann, and S. Kim, "Duplicate bug reports considered harmful... really?," in *Proceedings of the International Conference on Software Maintenance (ICSM'08)*, pp. 337–345, IEEE, 2008.

[10] P. Runeson, M. Alexandersson, and O. Nyholm, "Detection of duplicate defect reports using natural language processing," in *Proceedings of the 29th International Conference on Software Engineering (ICSE'07)*, pp. 499–510, IEEE, 2007.

[11] A. J. Ko, B. A. Myers, and D. H. Chau, "A linguistic analysis of how people describe software problems," in *Proceedings of the Symposium on Visual Languages and Human-Centric Computing (VL/HCC'06)*, pp. 127–134, IEEE, 2006.

[12] A. Sureka and K. V. Indukuri, "Linguistic analysis of bug report titles with respect to the dimension of bug importance," in *Proceedings of the Third Annual ACM Bangalore Conference (COMPUTE'10)*, pp. 9:1–9:6, ACM, 2010.

[13] S. Haiduc and A. Marcus, "On the use of domain terms in source code," in *Proceedings of the 16th International Conference on Program Comprehension (ICPC'08)*, pp. 113–122, IEEE, 2008.

[14] B. C. Malt, S. A. Sloman, S. Gennari, M. Shi, and Y. Wang, "Knowing versus naming: Similarity and the linguistic categorization of artifacts," *Journal of Memory and Language*, vol. 40, no. 2, pp. 230–262, 1999.

[15] E. Reiter and S. Sripada, "Human variation and lexical choice," *Computational Linguistics*, vol. 28, no. 4, pp. 545–553, 2002.

[16] S. Teufel and H. Van Halteren, "Agreement in Human Factoid Annotation for Summarization Evaluation," in *International Conference on Language Resources and Evaluation (LREC'04)*, 2004.

[17] L. Ponzanelli, A. Mocci, and M. Lanza, "StORMeD: Stack Overflow ready made data," in *Proceedings of the 12th Working Conference on Mining Software Repositories (MSR'15)*, pp. 474–477, IEEE, 2015.

[18] J. Allan, J. Callan, K. Collins-Thompson, B. Croft, F. Feng, D. Fisher, J. Lafferty, L. Larkey, T. N. Truong, P. Ogilvie, *et al.*, "The lemur toolkit for language modeling and information retrieval. The Lemur Project," 2003.

[19] "Online replication package: https://seers.utdallas.edu/projects/duplicates-vocabulary-agreement."

[20] M. McCandless, E. Hatcher, and O. Gospodnetic, *Lucene in Action: Covers Apache Lucene 3.0*. Manning Publications Co., 2010.

[21] R. J. Grissom and J. J. Kim, *Effect sizes for research: Univariate and multivariate applications*. Routledge, 2012.