

A Software Requirements Ecosystem: Linking Forum, Issue Tracker, and FAQs for Requirements Management

James Tizard, Peter Devine, Hechen Wang, Kelly Blincoe, *Member, IEEE*

Abstract—User feedback is an important resource in modern software development, often containing requirements that help address user concerns and desires for a software product. The feedback in online channels is a recent focus for software engineering researchers, with multiple studies proposing automatic analysis tools. In this work, we investigate the product forums of two large open source software projects. Through a quantitative analysis, we show that forum feedback is often manually linked to related issue tracker entries and product documentation. By linking feedback to their existing documentation, development teams enhance their understanding of known issues, and direct their users to known solutions. We discuss how the links between forum, issue tracker, and product documentation form a requirements ecosystem that has not been identified in the previous literature. We apply state-of-the-art deep-learning to automatically match forum posts with related issue tracker entries. Our approach identifies requirement matches with a mean average precision of 58.9% and hit ratio of 82.2%. Additionally, we apply deep-learning using an innovative clustering technique, achieving promising performance when matching forum posts to related product documentation. We discuss the possible applications of these automated techniques to support the flow of requirements between forum, issue tracker, and product documentation.

Index Terms—Requirements engineering, machine learning, natural language processing, deep learning, open source software, user feedback, software engineering

1 INTRODUCTION

USER feedback is a vital resource for modern software developers, helping to give insight into user needs and identify usability issues. Developers frequently elicit requirements from user feedback, such as bug reports and feature requests, to help guide the maintenance and evolution of their products [1]. Keeping users happy through continuous product evolution is an increasingly critical factor for development teams, as for many applications, the competitor is only a click away [2], [3].

Online channels such as app stores, social media, and product forums are a popular avenue for software users to give feedback and discuss the products they are using. These channels have been a focus for software requirements engineering research, which has found that the feedback contains valuable information for product development teams. These studies also highlight the need for automatic analysis tools to assist in requirement extraction, as manual extraction can be extremely time consuming due to the large volume of feedback [4], [5], [6], [7], [8], [9], [10], [11].

Open source software projects often use product forums to support their users, as well as issue trackers to document and track requirements. When a software user posts in a product forum they often describe an issue they have encountered and are seeking help to resolve [12]. These posts can describe bug reports, missing features, or un-

intuitive features. Experienced users, as well as members of the development team, can reply to forum posts giving guidance on the nature of the user's issue [5]. To the best of our knowledge, it is currently unknown whether development teams utilize this source of feedback to identify new software requirements and how software requirements flow between these different channels.

To address this gap, we perform an empirical analysis of the product forums of two large open source software projects, VLC media player and Firefox web browser. We examine replies to forum posts to understand the types of links made to the issue tracker and to product documentation. This analysis was guided by our first research question:

RQ1: *In the VLC and Firefox forums, what types of links exist to their issue tracker and product documentation?*

Through the identification of links posted in forum replies, we show that experienced users and project contributors help identify and transfer requirements from forum to issue tracker. Once in the issue tracker, these user-sourced requirements can be tracked as developers work to enhance the software product. Contributors also often reply to forum posts with links to existing issues in the issue tracker, indicating that the development team is already aware of the issue and working on a solution. Finally, we also find that contributors frequently reply to forum posts with links to product documentation describing a common issue or frequently asked question (FAQ). This documentation often contains solutions or workarounds that can potentially help to address the user's issue.

While VLC and Firefox contributors continually work to

• J. Tizard, P. Devine, H. Wang, and K. Blincoe are with the Human Aspects of Software Engineering Lab, University of Auckland, New Zealand.

E-mail: {jtiz003, pdev438, hwan531}@aucklanduni.ac.nz, k.blincoe@auckland.ac.nz

respond to the issues of their forum users, this undoubtedly requires significant manual effort. Additionally, if continuous monitoring isn't maintained, important user issues may be missed or not addressed in a timely manner. Thus, we investigate analysis tools to support the processing of forum feedback. This investigation was guided by our final two research questions:

RQ2: *Can VLC and Firefox forum posts be automatically matched to issue tracker posts describing the same software requirement?*

RQ3: *Can VLC and Firefox forum posts related to a documented common issue, or FAQ, be automatically linked to that documentation?*

To address these research questions, we propose two automated approaches, based on the state-of-the-art Universal Sentence Encoder (USE). First, we apply USE in a semantic search, matching forum posts with issue tracker entries describing the same requirement (RQ2). Second, we apply USE using an innovative clustering technique, matching forum posts with related product documentation (RQ3). For both approaches, we demonstrate promising performance matching requirements between platforms, in VLC and Firefox.

Applying our proposed matching techniques, new forum posts can quickly be assessed to evaluate if the user's issue has already been documented in the adjacent platforms. Through matching related requirements, new contextual details from a forum post can be added to a matched issue tracker entry, strengthening understanding of the issue [13]. Additionally, if no match is found, a new forum post may represent a previously unknown issue, which should be transferred to the issue tracker for developer attention. Therefore, the proposed matching techniques have the potential to help identify new software requirements (future work can validate this). This is further discussed in section 7.

This paper makes five significant contributions: (1) In a quantitative analysis, we show that open source contributors (VLC & Firefox) frequently link forum posts to issue tracker and product documentation. To the best of our knowledge, this is the first time requirements management through the manual linking of these three platforms has been investigated. We discuss how these platforms have created a requirement ecosystem (see Figure 1) in open source software projects that can benefit from analysis tool support.

(2) We show that the requirements described by users in their forum posts can be automatically matched to related requirements in an issue tracker. While Haering et al. [14] recently demonstrated that app store reviews containing bug reports can be automatically matched to related issue tracker entries, we use a modified approach to match product forum posts with issue tracker entries. Previous work has shown that product forums differ to app reviews in both content and structure and may require alternative analysis techniques [5]. Additionally, our approach can be applied to a range of requirement types, including bug reports and feature enhancements, while the approach proposed by Haering et al. focused only on bug reports.

(3) We show evidence that forum to issue tracker document pairs describing the same requirement, but using

fewer related words, may not be identified using state-of-the-art unsupervised techniques (USE similarity). We discuss the implications if current similarity techniques struggle to identify requirements described with different language, such as from users of different backgrounds.

(4) We show promising performance, applying state-of-the-art deep-learning, to automatically predict the most appropriate documented issue and solution for forum posts. This approach can help to quantify the ongoing impact of known documented issues and unintuitive features, while also reducing the workload of project contributors.

(5) Finally, we provide a replication package¹ to give additional insight into this work and help facilitate future work. The package contains the collected data from the project forums and issue trackers of the studied projects (VLC & Firefox). We also include all python scripts and evaluation data used to answer RQ2 and RQ3.

The rest of the paper is structured as follows: Section 2 reviews the related work that informed our research. Section 3 describes our research setting. In Sections 4, 5, and 6, we present the research methodology and results for each research question, respectively. Our findings are discussed in Section 7. Section 8 discusses the threats to the validity and, finally, Section 9 concludes the paper.

2 RELATED WORK

2.1 Motivation to mine online user feedback

Van Oordt et al.'s recent survey of 101 developers found that the vast majority of developers (97%) agreed that user feedback gives them a better understanding of user needs and makes them aware of usability issues [1]. They found that 80% of developers often use feedback to identify bugs and 68% often use it to identify new feature requests. These results are in line with Pagano and Bruegge's [15] previous developer survey, which found that user feedback contains important information for developers, and it helps to improve software quality and to identify missing features.

Researchers have previously identified requirements relevant information in user feedback in several online channels, including app stores, social media, and product forums [5], [16], [17]. A recent survey found that software users are motivated to influence improvement in the applications they are using when giving online feedback [18]. Harman et al. showed the importance of addressing user concerns, finding a strong correlation between customer ratings and the popularity of mobile applications [19].

We extend this existing literature, by showing that forum users in two large open source software projects often report previously undocumented requirements, which are subsequently identified by project contributors and transferred to an issue tracker.

2.2 Automatically Matching Requirement Documents

Manually eliciting software requirements from online user feedback can be extremely time intensive, due to the large volumes and varying quality of the text [20].

1. <https://figshare.com/s/6b583d1882bb3f1baa1b>

Much previous research has investigated traditional machine learning techniques to automatically mine requirements from online feedback, focusing on classification, prioritisation, and summarisation of relevant feedback [4], [5], [6], [7], [8], [9], [10], [11].

More recently, research has applied similarity techniques to requirements engineering applications. Oehri and Guzman proposed SIMBA, a SIMilarity Based Approach to find similar documents across different platforms and languages [21]. Similarity is calculated using a word alignment technique to identify pairs of semantically related words. SIMBA can help prioritise feedback by identifying reoccurring requests. SIMBA can identify similar documents across app stores and social media, while this study focuses on matching requirements between the product forum and issue tracker.

Haering et al. recently applied the deep-learning model DistilBERT, proposing DeepMatcher, to match problem reports in app reviews to bug reports in issue trackers [14]. By matching app reviews to known issue tracker entries, DeepMatcher can help enhance those entries by adding context such as hardware versions or steps to reproduce the problem. However, Deepmatcher has only been applied to app stores. Previous work has shown app reviews and forum threads differ in content and structure and may need modified analysis techniques [5]. In this study, we investigate the ability of deep-learning techniques to match requirements between forum and issue tracker.

State-of-the-art deep-learning techniques have also been applied in recent studies to identify latent topics, by grouping semantically similar feedback. Stanik et al. [22], used the deep-learning model SBERT to first embedded user tweets, then applied HDBSCAN to identify topic clusters. Devine et al. evaluated an extensive set of approaches to group semantically similar feedback [23]. They found that the Universal Sentence Encoder (USE) outperformed a comprehensive set of alternative approaches, including BERT, as well as several non deep-learning models [24]. We build on these previous studies by applying state-of-the-art deep-learning to match requirements between forums, issue trackers, and product documentation. We demonstrate promising performance, in this previously uninvestigated real-world application.

3 RESEARCH SETTING

The focus of this study are two large open source software projects, the VLC media player and Firefox web browser. We investigate the forums, issue trackers, and product documentation of each project (Figure 1). Both VLC and Firefox have a large active community (developers and users) and have been active for many years, making them an ideal research setting. Future work should investigate additional open source products to further validate our findings. This is further discussed in the Threats to Validity (Section 8).

The context of our research setting is described below by considering five facets, as recommended by Petersen and Wohlin: *product*, *people*, *organisation*, *market* and *processes* [25].

3.1 VLC

The first *product* is VLC, a free, open source, cross-platform multimedia player, initially released in 2001. It supports

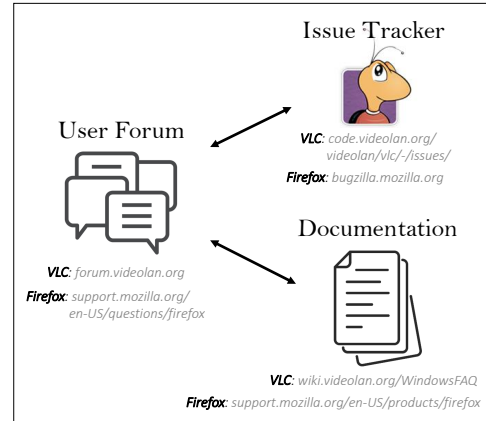


Fig. 1. Requirement Ecosystem: We identify links between product forum, issue tracker, and documentation (VLC & Firefox).

playing and streaming audio and video files in many different formats. The contributors (*people*) are a diverse and widely distributed collection of volunteers from over 40 countries.

VLC's development and administration is coordinated by VideoLAN, a French-based non-profit organisation.

VLC has a large existing *market*, with users downloading the application 73.2 million times as of October 2021 [26]. The popularity of VLC means effective product maintenance and evolution is important to meet the diverse needs and expectations of the users. With respect to *process*, the development community meets user needs with regular iterative releases and uses the Trac issue tracking system to track work for upcoming releases [27].

3.2 Firefox

The second *product* is Firefox, a free, open source, cross-platform web browser, initially released in 2004 [28]. Its primary features are tabbed browsing, private browsing, smart bookmarks, and a download manager. It allows third party add-ons. The *people* who contribute to Firefox are both volunteers and paid contributors, widely distributed around the world. Firefox's development and administration is coordinated by the Mozilla Foundation, a California-based non-profit organisation [29]. Firefox is one of the world's most popular browsers and has a large existing *market* of around 500 million yearly active users [30]. With respect to *process*, Firefox has regular, iterative releases and uses Bugzilla to track work for upcoming releases [31].

3.3 Data Collection

We developed custom web-scrappers to automatically mine the user feedback from the forums and issue trackers of both products. We collected all available data at the time of collection. For all collected data, the title and main body of the initial post were extracted. For each forum thread, the body text of all replies were also collected.

The forum data used in this research was collected for a previous study [5]. For VLC's forum, 38,000 posts, with all replies, were collected in May 2018 from the Windows platform sub-forum (the largest topic). From Firefox's forum,

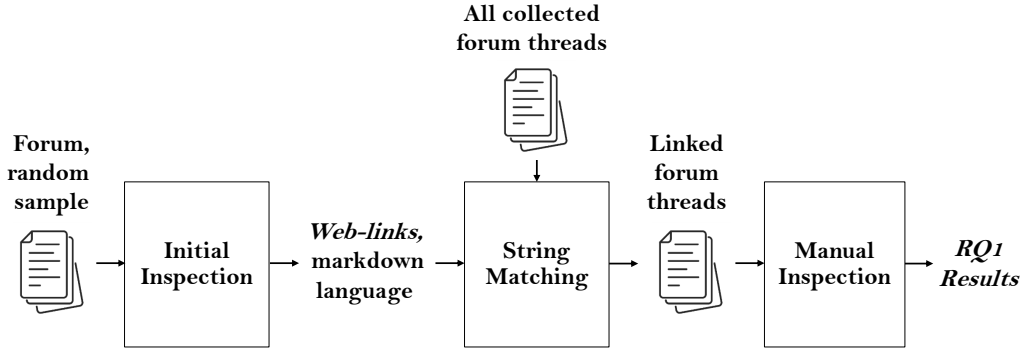


Fig. 2. *Empirical analysis overview (RQ1)*: Initially, we inspected a sample of forum threads to identify web-links and Markdown language syntax used to reference related issue tracker and documentation entries. Next, string matching was applied to find all forum threads containing the identified links. A final manual inspection was performed on the identified forums threads.

13,000 posts, with all replies, were collected in November 2018. The issue tracker data is newly collected for this study. From VLC's issue tracker, 20,452 entries were collected in April 2020. From Firefox's issue tracker, 499,159 entries were collected in February 2021.

4 EMPIRICAL ANALYSIS

RQ1: *In the VLC and Firefox forums, what types of links exist to their issue tracker and product documentation?*

4.1 Research Method

To investigate what types of links are made to the issue tracker and product documentation in replies to forum posts, we studied the forums of VLC and Firefox.

An overview of the full analysis is shown in Figure 2 and detailed below.

Initial inspection: Initially, we directly inspected a random sample of threads from each forum, to identify references to the issue tracker and product documentation. For both VLC and Firefox, the size of the random sample was determined by calculating the population needed to obtain a 99% confidence level with a confidence interval of 10%, from all collected forum posts [32]. This resulted in a manual analysis of 167 VLC posts and 165 Firefox posts.

In both products, links were identified in forum replies to their respective issue tracker and product documentation. These links took the form of explicit web-links and Markdown language, which enables plain text to be converted to valid HTML links. The syntax of the identified links are shown in Table 1.

Next, each sample thread containing one of the identified links was categorised into common themes using Thematic Content Analysis [33]. This identified four main types of links in forum threads, three types to the issue tracker and one to product documentation (see Section 4.2). The identification of links as well as the thematic content analysis was conducted by the first author and validated through iterative discussions with another author.

String matching: Using the identified link and Markdown language syntax shown in Table 1, string matching was applied to find all forum replies containing one of the identified link types.

Manual inspection: To quantify the occurrence of each type of link, another round of inspection was done on the full set of links identified using the string matching process. Links to the issue tracker were categorised into one of the three types, identified in the initial inspection. Both the content of the forum post and the issue description, were considered in this analysis. We also considered which of the linked forum and issue tracker entries were created first, to assess their relationship. This manual inspection was done by one coder (first author).

For product documentation, we manually inspected all linked documentation pages and report their contents.

4.2 RQ1 Results

We identified four main types of links to the issue tracker and product documentation in the forum post replies. Three to the issue tracker and one to product documentation:

Related issue tracker entry: Replies often linked to a pre-existing related issue in the issue tracker. There were 667 of these types of links in VLC forum post replies and 352 in

TABLE 1
Identified Web-links and Markdown Language Syntax

	Links
VLC - Issue Tracker	<ul style="list-style-type: none"> - <code>trac.videolan.org/vlc/ticket/[issue_id]</code> - <code>ticket [issue_id]</code> - <code>trac.videolan.org/vlc/newticket</code> - <code>trac.videolan.org</code> - VLC Trac
VLC - Documentation	<ul style="list-style-type: none"> - <code>wiki.videolan.org/WindowsFAQ/[issue_title]</code> - <code>wiki.videolan.org</code>
Firefox - Issue Tracker	<ul style="list-style-type: none"> - <code>bugzilla.mozilla.org/show/[issue_id]</code> - <code>bug [issue_id]</code> - <code>bugzilla.mozilla.org/enter_bug.cgi</code> - <code>bugzilla.mozilla.org</code> - <code>developer.mozilla.org/en-US/docs/Mozilla/QA/Bug_writing_guidelines</code>
Firefox - Documentation	<ul style="list-style-type: none"> - <code>support.mozilla.org/en-US/kb/[issue_title]</code>

Through an initial inspection of the VLC and Firefox forums, links were identified in post replies to the issue tracker and product documentation. These links took the form of web-links and Markdown language.

screensaver on vlc 0.8.4

by [redacted] » 28 Nov 2005 15:24

hi, i've installed the latest version of vlc. the screensaver is supposed to be disabled by default (box checked) but, mine still keeps running while playback.

by [redacted] » 28 Nov 2005 21:05

There is a reported issue of this in the TRAC database for windowed mode (so not fullscreen) under Windows:
<https://trac.videolan.org/vlc/ticket/426>

Fig. 3. Linking example, from a VLC forum thread. The replier confirms the issue is known and links the existing issue tracker entry.

Firefox. Figure 3 shows an example thread where a replier confirms the discussed issue is already known, and gives a link to the existing issue tracker entry. In this way, the forum user is shown that their issue is known and may be worked on. Additionally, the linked issue can contain solutions or workarounds for the user's issue.

Issue tracker entry created from the forum post: Forum post replies also contained links to newly created issues, which were found to be created because of the forum post. There were 417 such links in VLC forum post replies and 106 in Firefox. For example, one VLC forum reply says, "I've created the ticket 20175: <https://trac.videolan.org/vlc/ticket/20175>". The linked issues often refer back to the forum post in their description. In the example above the linked issue says "For logs, please visit the forum post about this bug: <https://forum.videolan.org/viewtopic.php?f=14&t=143355>".

Request to create a new issue tracker entry: In addition to direct links to issues in the issue tracker, we also found more generic links to the issue tracker with the reply encouraging the user to create a new issue in the issue tracker. The string matching and manual inspection identified 245 of these links in VLC and 77 in Firefox. In one example, a Firefox contributor says, "Perhaps you can take a few minutes to file a bug report; <https://bugzilla.mozilla.org>. This is in case others get the same problem". Sometimes users respond to these creation prompts confirming that they created a new issue, but not always.

Related product documentation: The most common type of link we found in the forum replies were links to related documentation. VLC's product documentation is contained within the VideoLAN Wiki² and Firefox provides product documentation in their Firefox Support platform³. Using string matching, we identified 3,374 links to VLC's product documentation and 7,456 links to Firefox's documentation.

For VLC, their product documentation is divided into topic subsections such as a Quick Start Guide, an Interface Description, Common Issues. The Common Issues subsection, which is titled WindowsFAQs⁴, contains a list of 52

common questions (FAQs) and problems. It is the most frequently linked, with 2,153 of the 3,374 links (63.8%) linking to one of these common issues. An example of a common issue in the product documentation is: *Why does VLC only give black, white or garbled video output?*

Each of these common issue pages describes the issue and gives a solution or workaround to help resolve the issue.

Firefox's product documentation also describes a substantial number of common issues and frequently asked questions (FAQs). We found links to 227 distinct common issues. The common issues in their documentation are also divided into topics, and each common issue can be associated with multiple topics. Table 2 shows the eight most common topics associated with the links made in the forum replies. The most commonly referenced topic was "Fix slowness, crashing error messages and other problems", with 3106 (41.6%) of the links associated with this topic. Where this topic describes ways Firefox may be malfunctioning, the remaining topics describe Firefox's features and how to use them.

Table 3 summarises the occurrences of each of these types of links. Overall, there were 4,703 links to the issue

TABLE 2
Firefox, links to Product Documentation Issues

Topic Title	Links	(%)
Fix slowness, crashing, error messages and other problems	3106	41.66
Manage preferences and add-ons	1531	20.53
Basic browsing	1076	14.43
Protect your privacy	495	6.64
Learn the Basics: get started	432	5.79
Download, install and migration	340	4.56
Install and update	325	4.63
Sync and save	171	2.29
Other topics	261	3.50
All Topics	7456	100.0

*Several common issues are contained in two topics, therefore the links per topic sum to more than the number of individual links found.

2. <https://wiki.videolan.org>

3. <https://support.mozilla.org/en-US/products/firefox>

4. <https://wiki.videolan.org/WindowsFAQ-2.1.x>

TABLE 3
Types of Links Posted in Forum Threads

Linked to	Type	VLC	Firefox
Issue tracker	Related issue tracker entry	667	352
"	Issue tracker entry created from the forum post	417	106
"	Request to create a new issue tracker entry	245	77
Product documentation	Related product documentation	3,374	7,456
Total		4,703	7,991

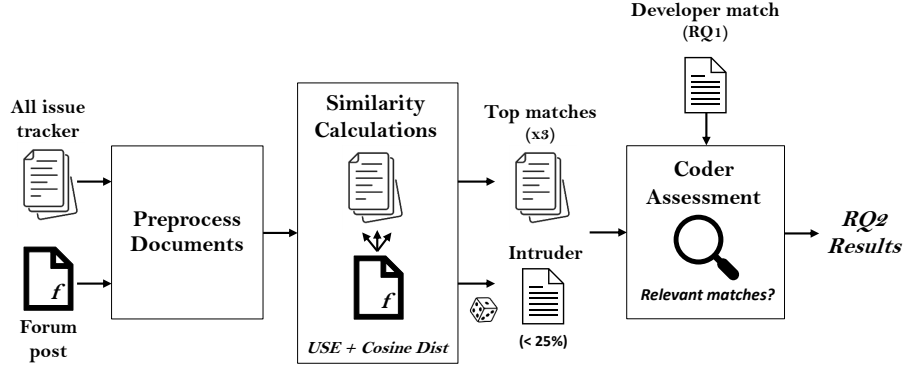


Fig. 4. Overview of the USE semantic search (RQ2). A forum post is compared to each issue tracker document, in a pool of potential matches. The issue tracker documents are rank in terms of similarity to the forum post.

tracker or product documentation in 38,000 VLC forum threads, and 7,991 links in 13,000 Firefox forum threads.

Answer to RQ1: For both VLC and Firefox, a significant number of links were posted in forum threads directing to their respective issue trackers and product documentation (see Table 3). Four primary types of links were found, links to: (1) a related (pre-existing) issue tracker entry; (2) an issue tracker entry created from the forum thread; (3) prompt the user to create a new issue tracker entry; and (4) related product documentation, with the most common of these being to documentation on known issues or frequently asked questions.

5 AUTOMATICALLY FINDING RELATED ISSUE TRACKER ENTRIES

RQ2: Can VLC and Firefox forum posts be automatically matched to issue tracker posts describing the same software requirement?

5.1 Research Method

To automatically find related requirements between the user forum and issue tracker, we started with a single forum post and calculated its similarity to each issue tracker entry collected from its respective issue tracker. The issue tracker entries were then ranked based on their similarity to the forum post, with the top three entries taken as the top matches (detailed below). This process was repeated for each forum post in the evaluation set, giving three top issue tracker matches for each. Next, coders manually evaluated the top matches for each forum post to assess their relevance. The full method is shown in Figure 4 and detailed below.

Data selection: We selected forum posts from VLC and Firefox that had a link to a related issue tracker entry, identified during RQ1, which we call the developer match. For both products, each issue tracker entry can be labelled as either a defect or an enhancement. The majority of the developer matched issue tracker entries (95%) were labelled as defects. Therefore, in order to evaluate a mix of requirement types, we selected all forum posts with enhancement type developer matches, giving 39 VLC and 36 Firefox posts. For defects, we randomly selected 100 forum posts with defect type developer matches, for each VLC and Firefox. This gave a total evaluation set of 275 forum posts with known developer matches, which is similar in scale to related studies [14].

Similarity technique selection: To answer RQ2, we performed a semantic search, where similarity was calculated between a forum post and each potential issue tracker match. We selected the Universal Sentence Encoder (USE) to perform these similarity calculations [24]. First USE is applied to transform two documents into the same vector space, then we use cosine distance to measure the distance between them. Cosine distance has previously been found to outperform other similarity measures for dense vector embeddings, such as USE [34], [35].

USE, as well as BERT similarity [36] and Word alignment [37], has been effectively applied to similarity tasks in recent requirement engineering studies [14], [21], [23]. USE and BERT are both deep-learning text encoders. We selected USE over BERT, as unlike BERT, USE has been specifically trained to identify semantic similarity between sentence pairs⁵. Additionally, in their recent work, Devine

5. <https://blog.floydhub.com/when-the-best-nlp-model-is-not-the-best-choice>

et al. found that USE outperformed a comprehensive list of approaches (including BERT) in grouping semantically similar feedback [23]. We selected USE over a word aligner method like the one proposed by Sultan et al. [37] since word aligner approaches are very slow compared to deep-learning approaches, and, thus, can be impractical for large searches.

Preprocessing: For all forum and issue tracker documents, the post title and main body were concatenated into a single string. As USE is designed to work on natural text, non-natural text, including logs, code blocks, and web-links, was removed from all evaluation documents using simple string matching.

Similarity calculations: For each of the 275 forum posts, we calculated its USE similarity to every collected entry from the respective (VLC/Firefox) issue tracker. The issue tracker entries were then ranked based on their similarity to that forum post. We recorded the top three issue tracker matches, with the highest USE similarity score, following Haering et al.’s approach [14]. It should be noted that, like Haering et al.’s recent work, we simply took the most similar issue tracker entries, for each forum post, as the best matches, and didn’t apply a threshold similarity score.

Additionally, we recorded the position, in the USE similarity list, of each developer match, to evaluate the ability of USE to find them.

Coder assessment: To evaluate the quality of the matches found with USE, we used manual coders to assess their relevance. For each forum post, the three top USE matches were manually assessed.

To balance the coding process we also added one intruder match for each forum post, shuffled into these top USE matches. The intruders were selected to be false matches to the forum post, randomly selected from the issue tracker entries that received a low USE similarity score (<25%). The intruder selection is similar to the process used by Guzman et al. [6]. Note, unlike intruder detection evaluation methods, we didn’t ask the coders to identify the intruder, only to evaluate the relevance of each match.

In addition to the top three matches and the intruder, the coders also assessed the relevance of the developer match. The developer match was separate and known to the coders, since they are identified in the forum posts (linked), and therefore couldn’t be anonymous. This gave five matches (including one intruder or false match) to be assessed per forum post.

Two coders independently read the five matches for each forum post and assessed which were relevant. The coders were instructed to consider a match as relevant when both the described application feature (e.g. video subtitles) and the behaviour (e.g. subtitles missing) were the same as the forum post. After the individual assessment, the coders came together to discuss disagreements and come to a consensus on each match. The inter-coder reliability after each stage is shown in Table 4. The matches that did not reach consensus between the coders (12 VLC, 9 Firefox) were labelled as irrelevant in order to not over estimate performance. The two coder process of manually assessing the relevance of each match (described above), directly follows the approach detailed by Haering et al. [14].

TABLE 4
Inter-coder Reliability

	Initial Agreement	Reconciled Agreement
VLC	90.6%	98.3%
Firefox	86.8%	97.8%

Top USE matches: To measure the accuracy of USE’s top matches, we report the number of (coder judged) relevant matches found per forum post in the three top USE matches. We also calculated the mean average precision (MAP), which describes average precision (AveP) for each USE match m , averaged over all USE matches M [38]. In our study, we applied a stricter version of MAP, known as MAP@3, in which only matches ranked in the top 3 positions, for each forum post, contribute to the AveP score (AveP@3) [39].

$$MAP@3 = \frac{\sum_{m=1}^M AveP@3(m)}{M}$$

MAP@3 is a conservative metric that may underrepresent performance as it assumes that there are at least three relevant issues tracker documents to be found per forum post. Each forum post has one known developer match, however beyond that, the actual number of relevant matches is unknown. We therefore also report the hit-ratio@3, which is the proportion of forum posts that had at least one relevant match in the three USE matches.

Developer matches: We separately evaluated the ability of USE to identify the developer match for each forum post. During the coding phase, 32 Firefox and 16 VLC developer matches were judged to not describe the same requirement as their forum post. These 48 irrelevant developer matches were excluded from this analysis, leaving 228 in the evaluation. We examine the position of the developer match, for each remaining forum post, in the USE similarity list. We report the mean and median position of the developer matches in the list. We also report the hit-ratio@3, which is the proportion of developer matches appearing in the top three USE matches, over the evaluated forum posts.

In any cases where developer matches received a low USE score, we manually analysed these to identify potential reasons. This analysis included manually reading the posts and quantitatively examining overlap using word alignment. For each forum, we applied Sultan et al.’s word aligner to the 20 developer matches with the lowest USE rank position and to the top USE match for the associated forum posts [37]. Following Oehri and Guzman [21], we filtered the aligner input for the most informative words, keeping only nouns, verbs, adjectives, and adverbs. We also updated the aligner to use PPDB 2.0 [40].

5.2 RQ2 Results

Top USE matches: The results of this analysis are presented in Table 5.

From the 275 forum posts (139 VLC and 136 Firefox), the USE issue tracker matches were found to contain matching requirements with a mean average precision (MAP@3) of

TABLE 5
RQ2: Top USE Matches

Forum	Forum Posts	Hit-Ratio@3 (%)	Relevant Matches Average	MAP@3 (%)
VLC	139	81.3	1.50	57.8
Firefox	136	83.1	1.68	60.0
Total	275	82.2	1.59	58.9

For each forum post, a semantic USE search returned the three most similar issue tracker documents. The hit-ratio@3 describes the proportion of forum posts with at least one relevant match.

Developer Match, with low USE similarity
(4 alignments, USEsim=42.5%)

<p><u>Cannot keep Firefox centered on my screen</u></p> <p>Up to recently Firefox would remember the window position after quitting. Now it always loads at a fixed position and I cannot keep it centered. How do I fix this?</p>
<p><u>Window isn't restored in prior position on external-monitor with > 100% DPI</u></p> <p>there is feedback from some users on various support channels after the 62 update yesterday that their browser windows aren't restored in the last position in a new session anymore - all from users with a dual-monitor set-up so far: the privacy.resistFingerprinting pref isn't in play here apparently.</p>

Top USE Match
(10 alignments, USEsim=78.8%)

<p><u>Cannot keep Firefox centered on my screen</u></p> <p>Up to recently Firefox would remember the window position after quitting. Now it always loads at a fixed position and I cannot keep it centered. How do I fix this?</p>
<p><u>Firefox maintains position even if it is offscreen after a resolution change</u></p> <p>On my monitor, when I change to picture by picture, it cuts my resolution in half vertically. If I start Firefox after having it opened on my fullscreen it starts with the titlebar off the window to the top (and no way to move/resize except via keyboard). If the top of the window is going to display off screen, we should move/resize the window to be onscreen (this is what other applications do). My guess is that we're either favoring size over position or we're somehow using the bottom left to position instead of top left.</p>

Fig. 5. USE limitation example: The example shows two matches for one forum post. The forum post is repeated as the top (blue) document on each side (left & right). On the left, is a match that USE rates low, but developers rate highly. On the right, is a match, for the same forum post, that USE scores highly. We observe that matches with a high USE score generally have more aligned words than developer matches that get a low USE score. The aligned words in the examples have been highlighted.

58.9% and a hit-ratio@3 of 82.2%. We observed consistent performance in the relevance of the USE matches between VLC and Firefox, as shown in Table 5. All 275 intruder matches were assessed to be irrelevant by the human coders, which is in line with their low USE similarity scores (<25%).

Developer matches: The average similarity positions of the developer matches in the USE search are reported in Table 6. The developer matches often appeared low in the USE similarity list, with just 36.7% (VLC) and 11.0% (Firefox) appearing in the top three results (hit-ratio@3). Many developer matches had a very low USE similarity score, leading to an average similarity rank position of 479 (out of 20.5k) for VLC and 14,006 (out of 499.2k) for Firefox.

Applying word alignment to the lowest ranked devel-

oper matches showed that they generally share less related words with their forum post than the equivalent top USE match. For Firefox, the top matches had a USE similarity of 73.1%, compared to 39.7% for the worst performing developer matches. The top matches had an average of 24.5 aligned word pairs, compared to just 13.1 aligned pairs for the developer matches. With 16/20 top matches having more aligned pairs than the respective developer match.

For VLC, the top matches had a USE similarity of 72.4%, compared to 53.9% for the worst performing developer matches. The top matches had an average of 21.7 aligned word pairs, compared to just 16.4 aligned pairs for the developer matches. With 15/20 top matches having more aligned pairs than the respective developer match.

TABLE 6
RQ2: Developer Matches, USE Search Rank

Forum	Forum Posts	Hit-Ratio@3 (%)	Rank of Match Median	Rank of Match Average	Search Pool
VLC	124	36.7	7.0	479.1	20,452
Firefox	104	11.0	162.5	14,006.1	499,159

Each forum post had one known, developer matched, issue tracker document. This table reports the USE search rank of the known matches. Hit-ratio@3 describes the proportion of known developer matches returned in the top three search results.

An example of a developer identified match with a low USE similarity score is shown in Figure 5. The developer match received a USE similarity of 42.5% compared to 78.8% for the top USE match. The word aligner identified four word pairs in the developer match, compared to ten in the top USE match.

Answer to RQ2

Finding 1: Forum posts were automatically matched to issue tracker documents containing the same software requirement, with a MAP@3 of 58.9% and Hit-Ratio@3 of 82.2%. The matched requirements consisted of both defect reports and enhancement requests.

Finding 2: Poor performance was observed in automatically matching some forum posts to developer identified issue tracker documents with USE similarity. These poor performing developer matches were compared to the top USE matches, for the same forum post, using a word aligner. This showed that the best USE matches generally contain more aligned words than developer matches that aren't found with USE.

6 AUTOMATICALLY FINDING RELATED DOCUMENTED ISSUES

RQ3: *Can VLC and Firefox forum posts related to a documented common issue, or FAQ, be automatically linked to that documentation?*

6.1 Research Method

In RQ1, we found many instances where a forum post was linked to product documentation describing a common issue or FAQ, with an associated solution or work around. For simplicity, we refer to each documentation entry as an FAQ in this section, where an FAQ refers to a common issue or question and its related documentation including workarounds and solutions. For RQ3, we aimed to automatically match forum posts to the most relevant FAQ.

As in RQ2, we started with an evaluation set of forum posts, then apply USE to vectorize documents and then calculate similarity based on cosine distance. Unlike matching forum posts to issue tracker issues (RQ2), each FAQ can be linked in many forum posts. This allowed us to evaluate a clustering approach, with the goal of improving performance over matching individual documents.

Data selection: Evaluation data for RQ3 was taken from the VLC and Firefox forum posts with linked FAQs (see RQ1). For VLC, we focused on links to the 52 FAQs (common issues) with associated solutions within the VideoLAN wiki. This documentation addresses specific FAQs, unlike the general wiki, and is therefore most suited to directly address user concerns and make specific predictions. All Firefox product documentation links are to one of 227 specific issues and are suitable for this evaluation.

We began with 2,153 VLC and 7,456 Firefox forum posts, linked to their respective FAQs. To remove ambiguity, forum posts linked to multiple FAQs were removed from the

TABLE 7
RQ3 Evaluation Data

#	VLC FAQs	Number (%)
1	Why does VLC only give black, white or garbled video output?	646 (52.9%)
2	Crackles, pops, hisses and other audio anomalies	188 (15.4%)
3	H.264/MPEG-4 AVC playback is too slow	172 (14.1%)
4	How can I separate playback controls from playback window?	38 (3.1%)
5	DVD movies don't playback smooth (they stutter, lag, etc.)	21 (1.7%)
6	How do I handle the broken AVI files?	21 (1.7%)
7	How do I set the default deinterlace method?	20 (1.6%)
8	How do I change my output device in case I have multiple audio devices connected to my PC?	15 (1.2%)
9	How can I disable fullscreen controller?	14 (1.1%)
10	How can I select a Unicode font?	13 (1.1%)
11-25	Other FAQs	73 (6.0%)
#	Firefox FAQs	Number (%)
1	Troubleshoot Firefox issues caused by malware	232 (9.6%)
2	Upgrade your graphics drivers to use hardware acceleration and WebGL	145 (6.0%)
3	Troubleshoot Firefox issues using Safe Mode	115 (4.8%)
4	Troubleshoot extensions, themes and hardware acceleration	78 (3.2%)
5	Back up and restore information in Firefox profiles	77 (3.2%)
6	Profiles - Where Firefox stores your bookmarks, passwords and other user data	76 (3.2%)
7	How to download and install Firefox on Mac	71 (2.9%)
8	Profile Manager - Create, remove, or switch Firefox profiles	66 (2.7%)
9	Update Firefox to the latest release	63 (2.6%)
10	Can't add, change or save bookmarks - How to fix	58 (2.4%)
11-150	Other FAQs	1427 (59.3%)

evaluation set, leaving 1,233 VLC and 2,455 Firefox posts. Finally, 12 VLC and 47 Firefox FAQs with only one linked forum post were removed, as no post would be available to form the FAQ centroid (detailed below). This left 1,221 VLC and 2,408 Firefox forum posts, each linked to one of 25 FAQs for VLC, and 150 FAQs for Firefox. An overview of the most common FAQs, in the evaluated RQ3 forum posts, for both VLC and Firefox is given in Table 7.

Evaluation: For each forum post in the evaluation data, we aimed to match it to the most relevant FAQ. Each possible FAQ match was represented with a centroid vector calculated with USE. To calculate each FAQ centroid, we first randomly selected forum posts from the evaluation set, linked to that FAQ. We evaluated using one forum post to calculate each centroid, then increased to 10% of the available forum posts for each FAQ (minimum of one), finding the average USE vector embedding. Posts used to form the centroids were then removed from the evaluation data.

Next, we vectorized each remaining forum post with USE and calculated similarity to the FAQ centroids, based on cosine distance. For each forum post, the FAQ centroids

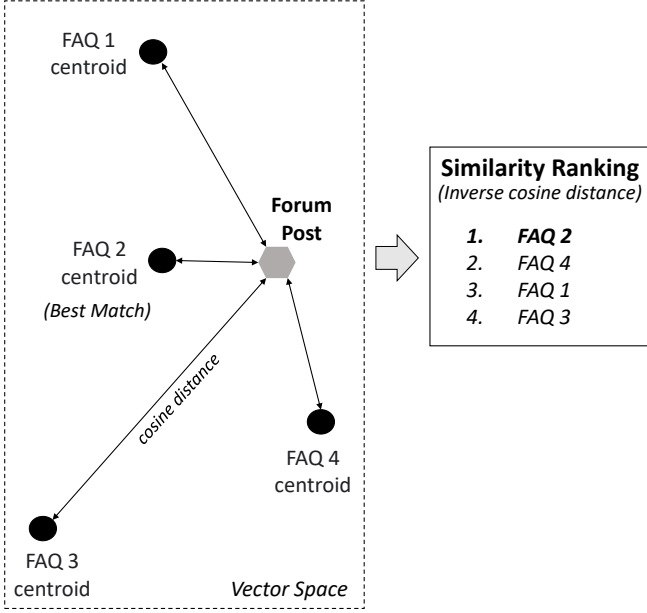


Fig. 6. Illustrative Example: Predicting the most relevant FAQs described in a forum post. Each forum post is vectorized with USE and the most similar FAQ centroids are ranked.

were ranked from most to least similar. An overview of the approach is shown in Figure 6.

For each forum post, we used the linked FAQ, identified during RQ1, as ground truth. To evaluate performance, for each forum post, we compared the *true* FAQ to the predicted most similar FAQ centroid.

6.2 RQ3 Results

The evaluation results for RQ3 are shown in Table 8. The *FAQs* column shows the number of possible FAQ matches for each forum post based on the total FAQs remaining after applying the above filters. For example, VLC has 25 FAQ entries that each forum post can be matched to. The *Forum Posts* column shows the number of evaluated forum posts, each being automatically matched to FAQ's, and does not include the posts used to form the FAQ centroids. The *Centroid Posts* column shows the number of forum posts used to calculate the FAQ centroids. Finally, the *Rank of True FAQ* columns show the mean and median positions the *true* FAQ match was ranked, over all forum posts.

For both forums, the best results were obtained by using 10% of each FAQ's forum posts to form the centroids. For VLC, the *true* FAQ matches had a median position of 1 out of

the 25 possible matches, with an average of 1.96. For Firefox, the *true* FAQ matches had a median position of 8 of 150 possible matches, with an average of 18.35. The reduced performance on the Firefox forum, compared to VLC, is discussed in Section 7.

Answer to RQ3: We applied USE similarity to match forum posts to FAQs, described in product documentation. For VLC, the *true* FAQ match for each post was similarity ranked with a median position of **1.0/25** and an average of **2.0/25**. For Firefox, the *true* FAQ matches were ranked with a median position of **8.0/150** and an average of **18.4/150**.

Note: The variation in performance between VLC and Firefox is discussed in Section 7.

7 DISCUSSION

Forums as a source of requirements: In RQ1, we identified many instances of software requirement-relevant information being posted by users in a product forum before being known to developers. We observed forum issues that were transferred to create issue tracker entries, as well as requests for users to document their forum post in the issue tracker. These observations are in line with the recent study by Haering et al., which found that users often submit problem reports in the Google Play Store months before the bug is documented in an issue tracker [14]. Haering et al. recommended that developers continuously monitor user feedback in app stores to discover problems early. We endorse this recommendation and present strong evidence that open source software projects (VLC and Firefox) already monitor their forums to identify new requirements and facilitate their transfer to the issue tracker.

The monitoring and transfer of issues from forum to issue tracker is evidently an important source of requirements for both VLC and Firefox, but likely requires substantial human effort. We see potential for automated tools to assist in this process. The proposed USE semantic search (RQ2) could perform a key role in forum monitoring, by finding and filtering issues that have already been documented in the issue tracker. Additionally, forum posts without an issue tracker match may represent a previously undocumented issue, which should be entered into the issue tracker for developer attention. Future work could apply our proposed semantic search approach to identify potential undocumented user issues. Then state-of-the-art deep-learning classifiers [41]

TABLE 8
RQ3 Results

Forum	FAQs	Forum Posts	Centroid Posts (Total Posts)	Rank of True FAQ Median	Rank of True FAQ Mean
VLC	25	1,196	1 per issue (25)	4 (/25)	5.61 (/25)
VLC	25	1,091	10% per issue (130)	1 (/25)	1.96 (/25)
Firefox	150	2,258	1 per issue (150)	24 (/150)	40.37 (/150)
Firefox	150	2,138	10% per issue (270)	8 (/150)	18.35 (/150)

We apply USE similarity to rank the best matching documented FAQ for a set of forum posts, where each post has a known true FAQ match. We report the similarity rankings (median, mean) of the true matches.

could be used to identify whether these represent requirements relevant information, such as bug reports or feature requests. Finally, a validation, with the development team, could examine whether the identified requirements relevant posts do in fact represent new software requirements, that the development team was unaware of.

We see an important role for semantic search in grouping matching requirement information to extract all important contextual details. Online user feedback often contains context that is critical for developers to understand and address issues. Important contextual details include steps to reproduce the issue, the application version, the hardware version, and the user's operating system [5], [42]. However, individual reports often don't include some or all these contextual details [13]. By identifying multiple reports of the same issue, context can be gathered to give a more complete picture of the issue, helping developers better understand its scope.

Limitations of unsupervised similarity techniques: While the top semantic search results (high USE score) often contained matching requirements, we observed limitations in finding developer identified matches that are described using different terminology (see RQ2). An analysis of these missed developer matches with a word aligner confirmed that they generally contain fewer related words. Having fewer word alignments suggests that word alignment based similarity would also struggle to identify some matches that developers consider good, even when using a comprehensive database of synonyms as our applied aligner did [40].

Future work can directly evaluate alternative similarity techniques, such as BERT similarity or word alignment, to see if they can yield improved performance. However, the performance of these techniques found in related work, is in line with this study [14], [21], [23]. Additionally, in their work, Haering et al. also reported missed requirement matches, suggesting that BERT similarity (applied there) likely has similar limitations [14].

If popular similarity approaches for grouping feedback, based on unsupervised deep learning models (USE, BERT), miss related requirements described using different words, this suggests potential inclusivity and representation issues. When considering new software requirements, it is important that the development team considers the needs and perspectives representative of all users, not just those who describe particular issues in a similar way, to ensure inclusive software design. Online software feedback is given by users around the world, with a wide variety of communication styles that may be influenced by culture [43]. Therefore, current similarity techniques may miss related feedback from users of different backgrounds and ways of communicating. Looking forward, work should be done to understand what features of feedback humans utilise to judge relatedness that isn't considered by current similarity techniques. This challenge is made more difficult by explainability issues with state-of-the-art deep learning models such as USE.

Matching Firefox to product documentation: In RQ3, we observed promising performance applying USE similarity to predict the documented FAQs described in VLC forum posts. However, the performance was not as good when ap-

plied to Firefox. This lower performance is likely impacted by the greater number of documented FAQs being assessed for Firefox (150), compared to VLC (25). To investigate this, we considered a smaller number of FAQs for Firefox. In our analysis for RQ3, we examined all forum posts which had a link to one of the FAQs. By looking at only the 25 most common FAQs, we investigated whether the reduced search space would improve the prediction accuracy. From the RQ3 evaluation data, 68% of Firefox forum posts (1,445/2,138) link to one of the 25 most common FAQs. Evaluating the USE approach on matching to only the 25 most common FAQs improved the median *true* FAQ match rank from 8 to 4 and the average from 18.4 to 6.5.

This demonstrates that prediction accuracy in Firefox can be improved by limiting the problem space, while still assessing the majority (68%) of forum posts since most link to the most common documented FAQs. Therefore, the application of the proposed approach is likely best suited to smaller projects or projects with only a small number of FAQs. Future work can investigate this, as well as other factors that affect accuracy.

Addressing reoccurring FAQs: We identified many instances where a forum post was linked to an FAQ in product documentation (see Table 3). The linked FAQs often contain a solution or work around and appear to be a key resource for assisting forum users.

Applying our proposed matching approach, we envision a system that takes the user's post and automatically recommends the top most related FAQ and associated solution. Quickly addressing user issues, without waiting for human attention, would likely help reduce user frustration, while also minimising the manual effort needed to direct users. Additionally, automatically identifying FAQs would help to quantify their impact on user experience and highlight areas where development efforts may be warranted. While this approach focuses on previously documented requirements (FAQs), it can assist the development team to prioritise these known issues for attention. For example, an FAQ describing a work-around for a known bug that occurs frequently, could be prioritised for further development.

A requirements ecosystem: Finally, in this study, we observed significant links between the product forum, issue tracker, and product documentation of two large open source software projects (VLC and Firefox). These three platforms are hosted on separate websites for both projects and serve distinct functions. However, we observe a consistent flow of new requirements from forum to issue tracker, as well as forum users frequently being directed to product documentation. It is also likely that some product documentation arises in the adjacent platforms. We see these platforms as having formed a type of requirements ecosystem, where users get product support and developers can elicit and document the most salient requirements. Future online platforms could consolidate the features of forum, issue tracker, and documentation into a single platform, giving integration features to support its users, such as automated requirement matching.

8 THREATS TO VALIDITY

In this section, we discuss the internal and external threats to the validity of our research.

Internal Validity: The manual relevance assessment of forum to issue tracker matches are a possible source of bias and coder error (see RQ2). We evaluated 275 forum posts, each with one manual match and three top USE matches, giving 1,100 document pairs to evaluate. The first step we took to mitigate potential bias was to introduce one intruder (false) match for each forum post (275 total), randomly shuffled into the top USE matches. These intruders helped to balance the evaluation set, giving both true and false matches. All 275 intruder matches were correctly coded as irrelevant. However, this manual assessment is still a possible source of bias.

Additionally, we mitigated bias by using two independent coders to evaluate each match, with three participating coders all having previous content-coding experience. The coders followed well-established coding practices [44] and were able to achieve a high level of intercoder reliability (see Section 5.1). However, again we cannot claim coder bias has been completely mitigated.

Finally, the evaluation of RQ2 primarily relates to the accuracy in recommending three top matches (MAP@3 and hit-ratio@3). The MAP@3 metric conservatively assumes there are at least three relevant matches, per forum document, and likely underestimates performance in many cases. The hit-ratio@3 is less conservative, simply testing if one good match is present in the top three. However, the actual number of relevant matches for each forum document is unknown, as is also the case in related work [14]. Therefore, the results can not speak directly to the accuracy in finding all good matches.

External Validity: One threat to external validity is the generalisability of our findings. To help mitigate this we investigated two diverse open source forums, VLC and Firefox. The findings we present in RQ1 were observed in both forums, and the analysis approach evaluated for RQ2 achieved consistent performance across each.

However, in matching forum posts to product documentation FAQs (RQ3), we observed a drop in performance from VLC to Firefox. This is likely related to the number of possible matching FAQs for each forum post (detailed in Section 7). The proposed USE approach is likely better suited to smaller projects or applications where there are fewer false matches.

Overall, we do not claim that the presented findings extend to all open source product forums. However, the findings of this research are a promising step, documenting how requirements flow from forum to issue tracker in two open source software projects, while also demonstrating the application of state-of-the-art analysis tools to support the elicitation of these requirements. Future work should validate our findings on additional open source products.

9 CONCLUSION

In this paper, we analyse the product forums of two large open source software projects, VLC and Firefox. In both forums, we observed that web-links are often posted in thread replies to issue tracker and product documentation.

We identified four primary reasons to for these links: 1) a link to an issue tracker entry describing an existing, related requirement; 2) a link to a issue tracker entry created from that forum thread; 3) a link to prompt the creation of an issue tracker entry; 4) a link to a related issue or FAQ described in product documentation.

Applying state-of-the-art USE similarity, we performed a semantic search for 275 forum posts (VLC and Firefox) in the respective issue trackers to identify entries describing the same software requirement. For each forum post, we analysed the top three search results and found they contained a matching requirement with a MAP@3 of 58.9% and Hit-Ratio@3 82.2%. This search performance matches Haering et al.s [14] recent study that focused on app store bugs. We provide contributions over this prior work by developing techniques for forums, which have different content and structure to app reviews and validating the techniques on a mix of requirements including both defects and enhancements. We also show that, despite these high hit-ratios, some feedback may be missed. Future work is needed to ensure diversity of user feedback is considered when grouping feedback related to a software requirement to ensure diverse perspectives are considered in the software design.

Finally, we applied USE similarity to forum posts to identify the most relevant entry in the respective product documentation. For each VLC and Firefox forum post, documented FAQs were ranked from most to least relevant. Promising performance was achieved for VLC, ranking the *true* match with a median of 1/25 and mean of 1.96/25. However, performance was reduced when applied to Firefox, achieving a median *true* match position of 8/150, and mean of 18.35/150. This is likely impacted by the larger number of documented issues in Firefox, meaning this approach may be limited to smaller projects.

For both VLC and Firefox, we observed strong links between product forum, issue tracker, and product documentation, forming a type of requirements ecosystem. We propose this ecosystem can be supported using state-of-the-art techniques, to both support product users and assist developers to elicit and document the most salient requirements. We encourage future work to investigate whether such links between product forum, issue tracker, and product documentation are present for other software products. Our proposed matching processes can also be trialled on other types of online feedback, such as app stores, to evaluate their viability beyond product forum feedback.

REFERENCES

- [1] S. Van Oordt and E. Guzman, "On the role of user feedback in software evolution: a practitioners' perspective," in *IEEE 29th International Requirements Engineering Conference (RE)*. IEEE, 2021, pp. 221–232.
- [2] J. M. Leimeister, H. Österle, and S. Alter, "Digital services for consumers," *Electronic Markets*, vol. 24, no. 4, pp. 255–258, 2014.
- [3] J. v. d. Hoven, "Ethics for the digital age: Where are the moral specs?" in *Informatics in the Future*. Springer, Cham, 2017, pp. 65–76.
- [4] E. Guzman, R. Alkadhi, and N. Seyff, "A needle in a haystack: What do twitter users say about software?" in *IEEE 24th International Requirements Engineering Conference (RE)*. IEEE, 2016, pp. 96–105.

- [5] J. Tizard, H. Wang, L. Yohannes, and K. Blincoe, "Can a conversation paint a picture? mining requirements in software forums," in *IEEE 27th International Requirements Engineering Conference (RE)*. IEEE, 2019, pp. 17–27.
- [6] E. Guzman, M. Ibrahim, and M. Glinz, "A little bird told me: mining tweets for requirements and software evolution," in *IEEE 25th International Requirements Engineering Conference (RE)*. IEEE, 2017, pp. 11–20.
- [7] W. Maalej and H. Nabil, "Bug report, feature request, or simply praise? on automatically classifying app reviews," in *IEEE 23rd international requirements engineering conference (RE)*. IEEE, 2015, pp. 116–125.
- [8] A. Di Sorbo, S. Panichella, C. V. Alexandru, C. A. Visaggio, and G. Canfora, "Surf: summarizer of user reviews feedback," in *IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*. IEEE, 2017, pp. 55–58.
- [9] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall, "Ardoc: App reviews development oriented classifier," in *24th ACM SIGSOFT international symposium on foundations of software engineering*. ACM, 2016, pp. 1023–1027.
- [10] N. Chen, J. Lin, S. C. Hoi, X. Xiao, and B. Zhang, "Ar-miner: mining informative reviews for developers from mobile app marketplace," in *36th International Conference on Software Engineering*. ACM, 2014, pp. 767–778.
- [11] J. A. Khan, Y. Xie, L. Liu, and L. Wen, "Analysis of requirements-related arguments in user forums," in *IEEE 27th International Requirements Engineering Conference (RE)*. IEEE, 2019, pp. 63–74.
- [12] J. Tizard, T. Rietz, and K. Blincoe, "Voice of the users: A demographic study of software feedback behaviour," in *IEEE 28th International Requirements Engineering Conference (RE)*. IEEE, 2020, pp. 55–65.
- [13] D. Martens and W. Maalej, "Extracting and analyzing context information in user-support conversations on twitter," in *IEEE 27th International Requirements Engineering Conference (RE)*. IEEE, 2019, pp. 131–141.
- [14] M. Haering, C. Stanik, and W. Maalej, "Automatically matching bug reports with related app reviews," in *IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 2021, pp. 970–981.
- [15] D. Pagano and B. Bruegge, "User involvement in software evolution practice: a case study," in *35th International Conference on Software Engineering (ICSE)*. IEEE, 2013, pp. 953–962.
- [16] D. Pagano and W. Maalej, "User feedback in the appstore: An empirical study," in *21st IEEE International Requirements Engineering Conference (RE)*. IEEE, 2013, pp. 125–134.
- [17] E. Guzman, R. Alkadhi, and N. Seyff, "An exploratory study of twitter messages about software applications," *Requirements Engineering*, vol. 22, no. 3, pp. 387–412, 2017.
- [18] J. Tizard, T. Rietz, X. Liu, and K. Blincoe, "Voice of the users: an extended study of software feedback engagement," *Requirements Engineering*, vol. 27, no. 3, pp. 293–315, 2022.
- [19] M. Harman, Y. Jia, and Y. Zhang, "App store mining and analysis: Msr for app stores," in *9th IEEE working conference on mining software repositories (MSR)*. IEEE, 2012, pp. 108–111.
- [20] E. C. Groen, N. Seyff, R. Ali, F. Dalpiaz, J. Doerr, E. Guzman, M. Hosseini, J. Marco, M. Oriol, A. Perini, and M. Stade, "The crowd in requirements engineering: The landscape and challenges," *IEEE software*, vol. 34, no. 2, pp. 44–52, 2017.
- [21] E. Oehri and E. Guzman, "Same same but different: Finding similar user feedback across multiple platforms and languages," in *IEEE 28th International Requirements Engineering Conference (RE)*. IEEE, 2020, pp. 44–54.
- [22] C. Stanik, T. Pietz, and W. Maalej, "Unsupervised topic discovery in user comments," in *IEEE 29th International Requirements Engineering Conference (RE)*. IEEE, 2021, pp. 150–161.
- [23] P. Devine, Y. S. Koh, and K. Blincoe, "Evaluating unsupervised text embeddings on software user feedback," in *IEEE 29th International Requirements Engineering Conference Workshops (REW)*. IEEE, 2021, pp. 87–95.
- [24] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Céspedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil, "Universal sentence encoder," *arXiv preprint arXiv:1803.11175*, 2018.
- [25] K. Petersen and C. Wohlin, "Context in industrial software engineering research," in *3rd International Symposium on Empirical Software Engineering and Measurement*. IEEE Computer Society, 2009, pp. 401–404.
- [26] VideoLAN organisation website. Accessed: September 2018. [Online]. Available: <https://www.videolan.org/index.html>
- [27] VideoLAN organisation website/ news archive. Accessed: September 2018. [Online]. Available: <https://www.videolan.org/news.html#news-2018-02-09>
- [28] Bbc news. Accessed: 4 July 2019. [Online]. Available: <http://news.bbc.co.uk/2/hi/technology/3993959.stm>
- [29] Mozilla foundation website. Accessed: 4 July 2019. [Online]. Available: <https://foundation.mozilla.org/en/about/>
- [30] Firefox public data report. Accessed: 4 July 2019. [Online]. Available: <https://data.firefox.com/dashboard/user-activity>
- [31] Bug report writing guidelines. Accessed: 4 July 2019. [Online]. Available: https://developer.mozilla.org/en-US/docs/Mozilla/QA/Bug_writing_guidelines
- [32] T. H. Wonnacott and R. J. Wonnacott, *Introductory statistics*. Wiley, 1990.
- [33] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative research in psychology*, vol. 3, no. 2, pp. 77–101, 2006.
- [34] D. Ayata, "Applying machine learning and natural language processing techniques to twitter sentiment classification for turkish and english," Master's thesis, Bogazici University, Istanbul, Turkey, 2018.
- [35] J. Ghosh and A. Strehl, "Similarity-based text clustering: A comparative study," in *Grouping Multidimensional Data*, J. Kogan, C. Nicholas, and M. Teboulle, Eds. Springer, Berlin, Heidelberg, 2006, pp. 73–97.
- [36] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019, pp. 3982–3992.
- [37] M. A. Sultan, S. Bethard, and T. Sumner, "Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 219–230, 2014.
- [38] M. Zhu, "Recall, precision and average precision," in *Department of Statistics and Actuarial Science, University of Waterloo Technical Report*, 2004.
- [39] J. Lin, Y. Liu, Q. Zeng, M. Jiang, and J. Cleland-Huang, "Traceability transformed: Generating more accurate links with pre-trained bert models," in *IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 2021, pp. 324–335.
- [40] E. Pavlick, P. Rastogi, J. Ganitkevitch, B. Van Durme, and C. Callison-Burch, "PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification," in *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, 2015, pp. 425–430.
- [41] R. R. Mekala, A. Irfan, E. C. Groen, A. Porter, and M. Lindvall, "Classifying user requirements from online feedback in small dataset environments using deep learning," in *IEEE 29th International Requirements Engineering Conference (RE)*. IEEE, 2021, pp. 139–149.
- [42] T. Zimmermann, R. Premraj, N. Bettenburg, S. Just, A. Schroter, and C. Weiss, "What makes a good bug report?" *IEEE Transactions on Software Engineering*, vol. 36, no. 5, pp. 618–643, 2010.
- [43] J. Tizard, T. Rietz, X. Liu, and K. Blincoe, "Voice of the users: A study of software feedback differences between germany and china," in *IEEE 29th International Requirements Engineering Conference Workshops (REW)*. IEEE, 2021, pp. 328–335.
- [44] K. Krippendorff, *Content analysis: An introduction to its methodology*, 4th ed. Los Angeles, CA, USA: Sage publications, 2018.



James Tizard recently completed his PhD at the University of Auckland, New Zealand (2022). His research primarily focuses on crowd sourcing software requirements (CrowdRE), often leveraging machine learning and natural language processing techniques. He has published work in the Requirement Engineering Journal, as well as at the IEEE International Requirements

Engineering conference, winning a distinguished paper award in 2020. For further details see jtizard.blog, or contact him at james.tizard@auckland.ac.nz.



Peter Devine is a final year PhD student at The University of Auckland, New Zealand. His research interests include requirements engineering, software user feedback, and natural language processing. He has published work at the IEEE International Requirements Engineering conference. Contact him at pdev438@aucklanduni.ac.nz.



Hechen Wang is a final year PhD student at The University of Auckland, New Zealand. His research focuses on using state-of-the-art techniques to automatically extract software requirements from online user feedback. He has published work at the IEEE International Requirements Engineering conference. Contact him at hwan531@aucklanduni.ac.nz.



Kelly Blincoe is a Senior Lecturer in Software Engineering at the University of Auckland, where she leads the Human Aspects of Software Engineering Lab (<https://hasel.auckland.ac.nz/>). Her research areas include collaborative software development, software analysis, and software requirements. She currently serves on the editorial board of the IEEE Transactions on Soft-

ware Engineering, the Empirical Software Engineering Journal, and the Journal of Systems and Software. She is also on the Executive Board of Software Innovation New Zealand. Contact her at k.blincoe@auckland.ac.nz.