

Universiteit
Leiden
The Netherlands

Healthcare information system engineering: AI technologies and open source approaches

Shen, Z.

Citation
Shen, Z. (2025, December 3). *Healthcare information system engineering: AI technologies and open source approaches*. Retrieved from https://hdl.handle.net/1887/4284431

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/4284431

Note: To cite this publication please use the final published version (if applicable).

Chapter 7

LOCATE: A web application to link open-source clinical software with literature

Nowadays, the effective utilization of open-source software could significantly boost both clinical research and practices, especially in resource-poor countries. However, the plethora of open-source clinical software has left many people unable to quickly locate the appropriate one for their needs. Commonly available software quality metrics and software documentation, such as downloads, forks, stars, and readme files, are useful selection criteria, but they only indicate the software quality from the perspective of IT experts. This paper proposes a method that offers additional insights on the performance and effectiveness of clinical software. It links open-source clinical software with relevant scientific literature, such as papers that use case studies of clinical software to reveal the strength and weakness of a given software from the clinical perspective. To interactively present the open-source clinical software and their related literature, we have developed the LOCATE web application that enables users to explore related literature for a given open-source clinical software. Moreover, the peer-review cycle of the application allows users to improve the application by confirming, adding or removing related literature. An evaluation experiment of the five most popular open-source clinical tools demonstrates the potential usefulness of LOCATE.

This work was originally published as: Shen, Zhengru, and Marco Spruit "LOCATE: A web application to link open-source clinical software with literature." *In Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2019)*-Volume 5. SciTePress, 2019. https://doi.org/10.5220/0007378702940301

7.1 Introduction

Open-source software plays a more important role in today's research, especially in the clinical field where using IT to leverage clinical practices has become ubiquitous (McDonald et al., 2003; Karopka et al., 2014; Reynolds and Wyatt, 2011). Furthermore, both the huge volume and the various types of clinical data accumulated on a daily basis require researchers to develop more advanced data analysis methods including both machine learning and deep learning algorithms (Raghupathi and Raghupathi, 2014). By using openly accessible tools or algorithms, both clinical researchers and practitioners can improve their work in many ways. To begin with, it significantly reduces the IT development cost, which, in return, allows us to focus (usually limited) resources on clinical issues (McDonald et al., 2003). Secondly, open-source clinical software, especially the well-supported in the open-source community, ensures the public accessibility of software platforms and tools in research and thus empower the scientific community to verify its reproducibility (McCormick et al., 2014). Lastly, your involvement contributes to the open-source community by verifying open-source clinical software in real-world settings (Kiah et al., 2014; Zettinig et al., 2015; Akowuah et al., 2015).

There are numerous open-source tools created across a variety of domains after decades of the open-source software advocacy (Anthes, 2016). With its accelerating popularity of open science, more and more will be added. Open-source clinical software covers its various research topics and clinical practices: medical images analyses (McCormick et al., 2014), medical text processing (Cunningham et al., 2013), clinical trials management systems (Haak et al., 2016), electronic health records systems (de Abajo and Ballestero, 2012) and so on. The plethora offers great opportunities for both clinical researchers and practitioners to accelerate their work with available open-source tools or algorithms. However, it also leaves many people unable to quickly locate the tools most suitable to their clinical research or practices. To make things worse, the lack of adequate usage examples in software documentation is a common issue for open-source software (McColl et al., 2014).

To select appropriate open-source clinical software, researchers or practitioners need to investigate a long list of available open-source software and go through a large volume of relevant literature. The search functions of most popular open-source software hosting platforms, like GitHub, enable users to easily retrieve clinical software to their specific needs. The commonly used databases, such as Google Scholar, PubMed, and Scopus, could help us obtain relevant literature for any open-

source software. But there is no unified platform that links open-source software directly to literature so that users can directly obtain related literature of a given open-source clinical software, instead of collecting and processing information from different sources. Recently, researchers in the life sciences have started to work on combining literature and open-source software (Wang et al., 2017). Wang et al. (2017) built an online biomedical software discovery platform based on data collected from PubMed literature and GitHub. It empowers biomedical researchers to easily find the suitable (open-source) tools they need. However, the study only included biomedical software that was reported in the biomedical literature from PubMed. To date, to the best of our knowledge, no single study has directly focused on building an online platform where users can easily locate and confidently select appropriate open-source clinical software.

Thus, the objective of this work is to bridge the research gap by developing a tool that links open- source clinical software to their literature. Currently, links between clinical software and their literature are described either in the readme files, like the Attention-Gated-Networks repository from GitHub, or in a separate GitHub repository where papers for some clinical software, such as deep learning methods for medical image processing, or blockchain for medical platforms and healthcare, are summarized. Both readme files and literature summaries have related papers hidden in unstructured text so that it is troublesome to extract such information and present it in a structured way. In this work, we first collected a large number of available open-source clinical software and then retrieved literature related to each of them through Google Scholar. Based on the collected data we built a web application with the following main functionalities: 1) searching open-source clinical software with any given topic; 2) showing relevant literature for a selected software, if there is any; 3) updating the existing clinical software and related literature; 4) adding new open-source software and related literature.

The remainder of the paper is organized as follows: section 7.2 explains the research approach, including data collection and processing methods, and artifact design strategies. In section 7.3, we present the system and two common use cases. Evaluation is discussed in section 7.4. section 7.5 and section 7.6 conclude the paper and outline future work.

7.2 Methods

7.2.1 Design Science

Our research followed the design science research as we built the application, because of its strength and popularity in solving a real-world problem by designing and building an innovative IT artefact (Hevner et al., 2008). In our case, the artefact is a system that links open-source clinical software and literature so that both clinical researchers and practitioners are able to efficiently locate their supporting resources, including both open-source software and papers. Specifically, we follow the design science research methodology (DSRM) proposed by Peffers et al. (2007), which consists of six steps: problem identification and motivation, a definition of the objectives for a solution, design and development, demonstration, evaluation, and communication.

The DSRM was initiated by the (I) problem identification and motivation, which we addressed by literature study and by reviewing other relevant online resources. As stated before, so far little research has been performed on linking open-source clinical tools to literature. But literature is a reliable resource that could provide additional information about clinical software such that we can make an informed decision on choosing the most suitable software. Such additional information could be the technical details of a clinical software, the strength and weakness of a given clinical software when comparing with other similar ones, or the software implementation advices learned from case studies. Based on the identified problem, we inferred (II) the objectives for a solution: creating a tool that links clinical open-source clinical software to their literature. In the (III) design and development, we built a web application. At first, we started by building a data pipeline in which open-source clinical software and literature were collected, processed and stored in our database. Then the artefact, namely the web application, was developed with Node.js and the React framework. To (IV) demonstrate the use of the system, two common use cases were presented. An (V) evaluation experiment measures the system reliability by comparing the automatically generated results from the system with the manually evaluated results. The final step of the DSRM is the (VI) communication. This paper serves as the start of our communication on this topic.

7.2.2 Data Pipeline

This section describes the data pipeline which collects open-source clinical software data and literature

Data Sources

Nowadays the open-source community has adopted a transparent version control system to manage source codes and other related files for long-term reproducibility and usability (Russell et al., 2018). Git dominates the open-source community with 87.2% of developers using it according to the 2018 Stack Overflow Developer Survey (Stack Overflow Survey, 2018). Open-source software hosted on Git platforms often refers to as a Git repository in which files along with all tracked changes are stored under version control. There are a number of online hosting platforms for Git repositories, including GitHub, SourceForge, Bitbucket and GitLab.

As the largest code host in the world, GitHub has reached 24 million developers working across 67 million repositories in 2017 (Octoverse, 2018). It hosts source codes of numerous open-source software in various domains. Figure 7.1 demonstrates GitHub's rising popularity in the clinical field by the proportion of PubMed articles mentioning GitHub in the title or abstract. In comparison with other platforms, GitHub has become the most used one and grows at the highest rate. Therefore, Given that GitHub repositories provide a good representation of available open-source software in the clinical domain, this study obtained open-source clinical software solely from GitHub. Besides, GitHub offers an easy API for external users to retrieve data from repositories (Russell et al., 2018).

Data Collection

This study refers to clinical software as software that is developed for either clinical practices or clinical research. Biomedical software, such as genome sequencing (Pabinger et al., 2014) or cell screening (Omta et al., 2016), are excluded. Therefore, we selected "(clinical OR medical) OR (patient OR doctor)" as the search term while using GitHub API to retrieve Git repositories. Furthermore, English terms were chosen because more than 97% of GitHub repositories have their names and descriptions in English. Figure 7.2 shows the numbers of Git repositories returned using the search term 'clinical' in different languages.

We chose Google Scholar to conduct our literature search. The literature search for each Git repository contains two main steps: 1) determining search terms for the

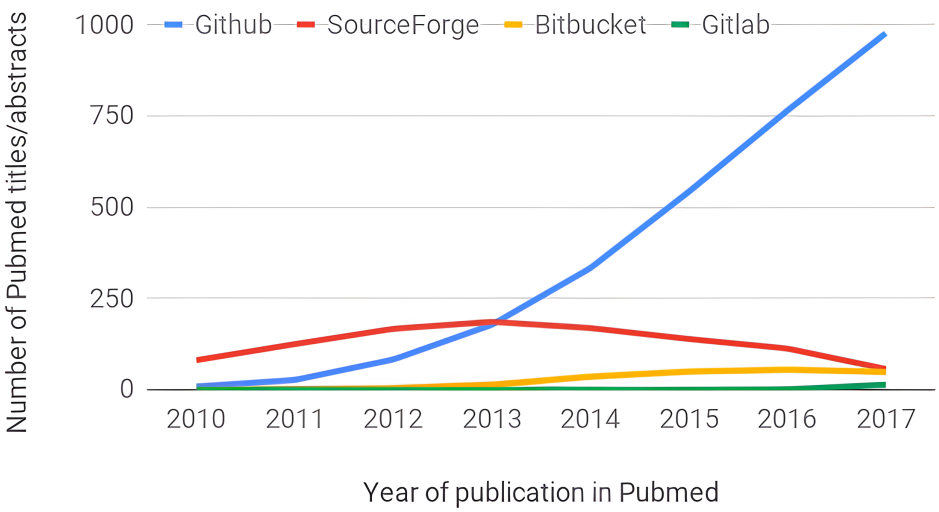


Figure 7.1: Trends of Git platforms mentioned in Pubmed titles or abstracts.

repository; 2) searching literature via Google Scholar with the search terms. Each Git repository derives two types of search terms. The first term is the full name of a Git repository, including both the owner name and the repository name, which retrieves papers that specifically mention the Git repository. Another search term contains both the repository name and keywords extracted from the repository description. This term identifies relevant literature that covers similar topics as a given Git repository. For each term, the top ten papers in terms of relevance are collected while there are more than ten papers discovered.

Data Extraction

GitHub REST API v3 exposes GitHub repository data to external users. Its search API offers an optimized solution for users to locate the specific items that interest them most, such as Git repositories, users, and issues. GitHub repositories in this study were first obtained by using the search repository API with the above terms. Then we extracted relevant data for each Git repository including name, description, readme files, stars, forks, and programming languages. Afterwards, a filter process excluded repositories based on their popularity and whether they contain source codes or not. We argue that Git repositories that receive no forks or stars three months after their creation are not reliable software or tools. Therefore, we filtered out such Git reposi-

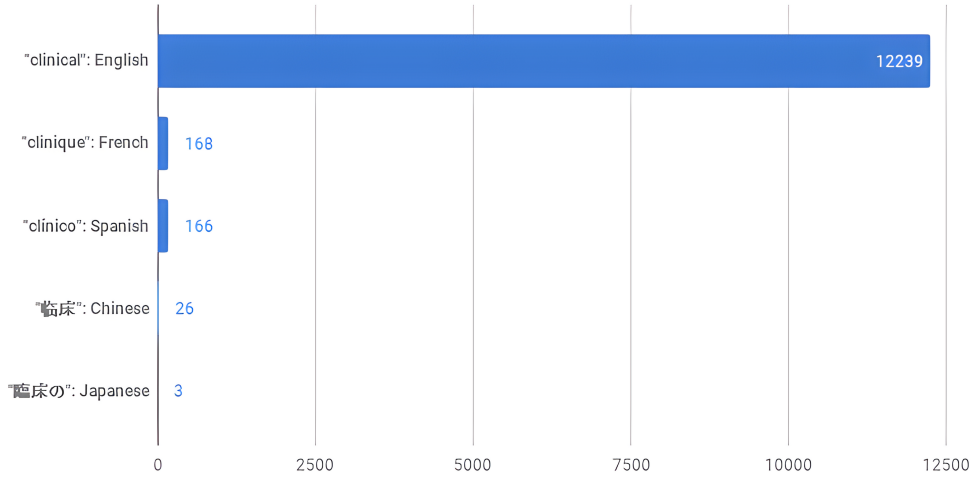


Figure 7.2: The number of repositories returned with the search term 'clinical' in different languages.

tories.

Data Processing

Keywords extracted from a GitHub repository description present an insightful indication of the repository. An external API, namely IBM Watson, is called to process the descriptions, which yields a list of informative keywords for each Git repository.

The extracted keywords cannot be directly employed as search terms for the literature retrieval as the list of keywords will contain many semantically similar keywords. A number of examples are: "patients data" vs "patients information", "medical appointments" vs "doctor appointments", "diabetes patients" vs "diabetic patients". To address this issue, we developed a normalization process in which semantically similar keywords were combined. The process consists of three steps: 1) calculating the semantic similarity between keywords pairs; 2) labelling keyword pairs based on their similarity; 3) replacing keywords with their semantically similar keywords.

Not all papers collected with the abovementioned method are in the clinical scope, especially those retrieved based on the second search term. For instance, ClearCanvas, a medical imaging tool, obtained a few papers about video games, titled as "Implementing Common Components of Video Games", "Build Your Own 2D Game Engine and Create Great Web Games". Therefore, we filtered out literature according

to their relevance to the clinical domain. Specifically, the abstract of each paper was examined to see if it mentions clinically relevant terms like 'clinical', 'medical', 'patient' or 'doctor'. If not, we exclude the paper. Prior to the paper filtering, our GitHub filtering helped us obtain a subset of all collected GitHub repositories. As mentioned earlier, the criteria include whether it has source code, the number of forks larger than 0, the number of stars larger than 0, and the readme file is not empty. In the end, 5119 GitHub repositories and 8820 related papers were collected. Figure 7.3 gives a detailed view of the above-mentioned data pipeline.

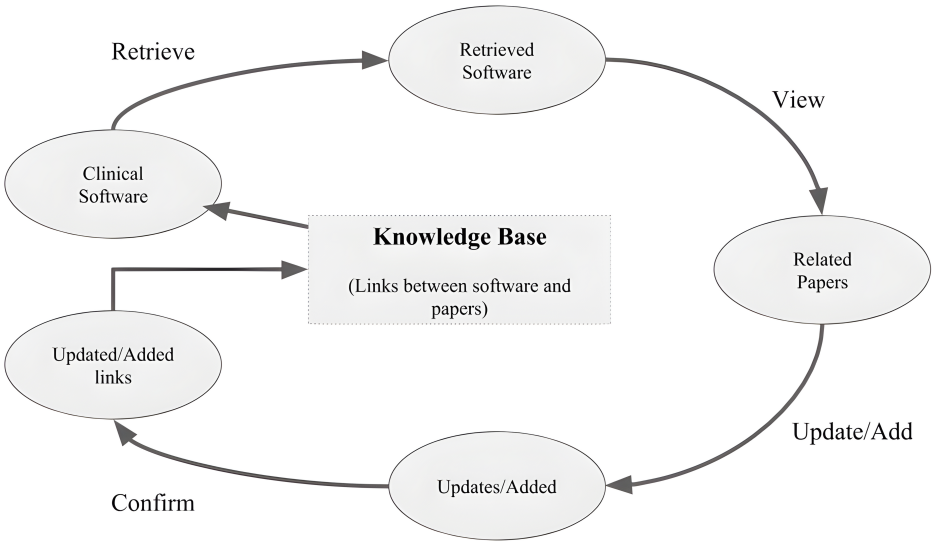


Figure 7.3: The peer review cycle.

7.3 LOCATE

7.3.1 Overview

LOCATE is a web application developed purely in JavaScript with support of popular open-source web development tools: Node.js and React. The web application, running online, is composed of interactive user interfaces and a REST (Representational State Transfer) API as the backend.

7.3.2 Key Design Principles

Continuously improving software is crucial in today's ever-changing environment. It is especially true to open-source software developed in academia where reproducibility is essential (Boettiger, 2015). Therefore, we adopted continuous integration which is a software development practice where software is continuously improved (Dingsøyr and Lassenius, 2016). Specifically, we implemented two methods to improve the core of our system, i.e. the data source. Firstly, since both open-source clinical software and literature are constantly added, we accordingly update our database on a regular basis in an incremental manner. Secondly, peer review was introduced to assess and modify the automatically extracted knowledge, particularly the links between open-source clinical software and papers. Figure 4 outlines the peer review cycle. Firstly, experts can retrieve clinical software of their interest and obtain a list of retrieved software. Experts give feedback on current links between the retrieved clinical software and literature, such as confirming, adding or removing links, then the system administrators make the final decisions upon the aggregated feedback. While there is no related paper found, experts can add relevant papers.

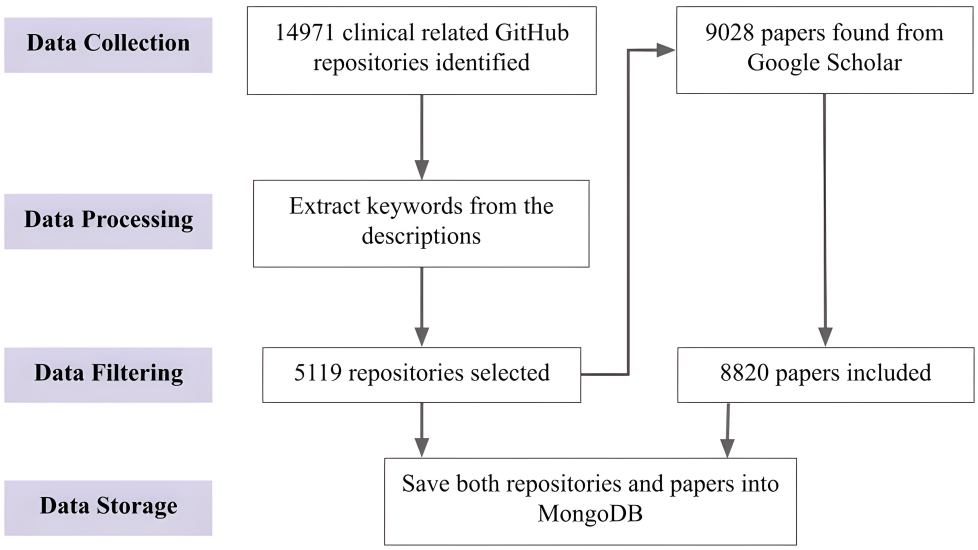


Figure 7.4: The data pipeline.

In the spirit of open-source, we made a choice to open source the application. Stewart (2016) grouped open-source software success into two broad types: development success which measures the success of attracting contributors from the open-

source community, and usage success that refers to the user interest/adoption. To ensure development success, source codes are modularized with the Model View Controller (MVC) architecture (Pop and Altar, 2014). The code and architecture simplicity clears barriers faced by newcomers to open-source software projects (Steinmacher et al., 2015). Moreover, we built a REST API that allows you to complete the CRUD (Create, Read, Update, Delete) operations. The REST API handles the server-side requests of the web application while providing a great deal of flexibility in expanding to other applications or the additional requirements of new use cases.

7.3.3 Technical Details

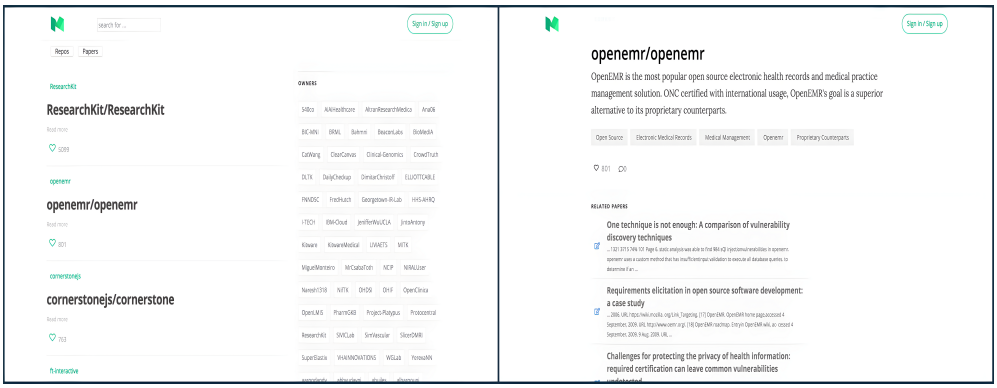
Since we intend to continuously improve the application, its data schema evolves accordingly. To support the dynamic schema of our data and the need for continuously redefining data structures, a non-relational (NoSQL) database, specifically MongoDB, was implemented. As one of the most popular document-based NoSQL databases, MongoDB allows us 1) to update schemas without modifying the existing data, 2) to easily manage the database without complicated database administrator skills, 3) to have good performance and availability (Bradshaw et al., 2019).

Full-text search is utilized to enable users to retrieve clinical software with specified search terms. Since both open-source clinical software and papers in our database contain large chunks of free text that describes them in detail, full-text search is a reasonable option. MongoDB offers a built-in full- text search feature that supports case-insensitive searches on text content.

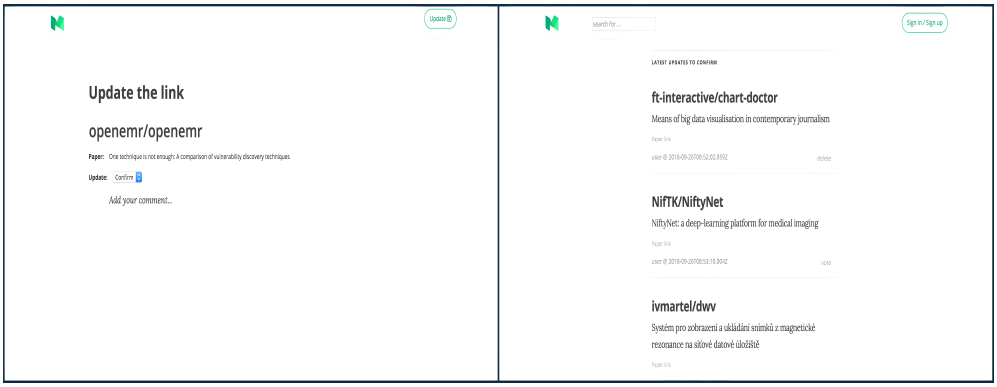
The following screenshots capture several key user interfaces of the application. Figure 7.5a shows the process of searching for open-source clinical software with or without a search term and assessing them based on related papers. The left screenshot shows a list of software returned by a request, while the right one displays a software called 'openemr' and its related papers. The peer review cycle that helps improve the application continuously is demonstrated in Figure 7.5b. The left interface demonstrates how a user update a link between software and literature. To better understand how the application works, we invite you to examine the web application at https://locate-repo.herokuapp.com.

7.3.4 Use Cases

As indicated above, LOCATE is capable of assisting both clinical researchers and practitioners. This section describes two use cases in detail to show how the applica-



(a) the search and view of open-source clinical software and related papers



(b) interfaces of the peer-review cycle

Figure 7.5: Screenshots of the application LOCATE

tion can be useful in clinical settings.

The first use case is that of suggesting open- source tools based on literature: given a clinical research topic, e.g. medical image segmentation, researchers who are conducting research on this topic need to investigate relevant literature and explore the potential appropriate tools to support them, preferably open-source ones. In this case, our application is able to recommend useful open-source software based on related literature. Without our application, researchers might need to search literature and software separately, which is a tedious process.

The second use case discussed in detail is that of choosing open-source tools. The lack of sufficient documentation in much open-source clinical software makes it difficult to assess their usability and reliability. In this context, the system matches open-

source software to their relevant literature which elucidates open-source clinical software in various aspects: ranging from detailing the technical features, comparing similar tools, to applications of tools in solving practical issues. With such additional information, researchers can make a more informed decision with regard to which tools to use.

7.4 Evaluation

This section presents the results of our evaluation experiment on whether the system offers additional knowledge about open-source clinical software. We selected five popular open-source clinical software projects to review the value of linked papers. Specifically, we examine the associated papers of each software and compare them with their GitHub repository documentation.

Software	Description	Docs	Papers from LOCATE
OpenEMR	Electronic health records and medical practice management solution	Features, manuals, and forum where people share their knowledge	Comparing OpenEMR with other open-source EMR systems; Case studies on specific issues of OpenEMR, such as performance and security.
DICOM Viewer	Medical image viewer	Setup manual	A survey of DICOM viewer software; Web case studies
DLTK	Deep Learning Toolkit for Medical Image Analysis	Setup manual, tutorials, sample applications	Paper explaining the tool; Case studies
Open Clinica	Open-source clinical trial software for Electronic Data Capture (EDC) Clinical Data Management (CDM)	Features, setup manual, forum	Case studies
NiftyNet	An open-source convolutional neural networks platform for research in medical image analysis and image-guided therapy	Features, manuals, other resources including StackOverflow questions	Paper explaining the platform; Survey papers; Paper on similar tools

Table 7.1: Comparison between software documentation and linked papers from LOCATE.

As shown in Table 7.1, related papers from LOCATE provide additional knowledge about open-source clinical software from several aspects: 1) a more detailed description of the development of the software; 2) studies that compare a number of similar open-source software projects; 3) case studies that apply the software to solve a specific clinical issue. Nevertheless, such knowledge cannot be obtained from software documentation which commonly exists as readme files, setup manuals, tutori-

als and so on. Therefore, the results confirm our assumption that enriching software documentation with its linked papers enables more informed decision making.

7.5 Discussion

This study developed a web application that links open-source clinical software with their related literature. To our best knowledge, no such studies have been conducted and our study is the first attempt to combine the two valuable components of today's clinical research. The tool which is available as an online web application offers an easy and openly accessible representation of our study. Moreover, an evaluation experiment which compared software documentation with the linked papers from LOCATE, outlines how the application enables more informed decision making.

The application has the potential of being beneficial for both practitioners and researchers in the clinical community. Practitioners obtain knowledge about the clinical tools of their interest from related literature so that they can better assess which one to choose. On the other hand, clinical researchers are provided with a list of potential useful open-source tools based on their research topics so that it might save a substantial amount of time by directly using available software or customizing them to fit their own research goals.

Furthermore, source codes of our study are open- sourced under the MIT license and hosted at https://github.com/ianshan0915/locate. People in other domains are able to utilize the source codes and conduct similar research in other domains.

Nevertheless, some limitations regarding this study should be noted. Firstly, GitHub is the only source for open-source clinical software in the study. Although GitHub is a major platform where developers work and share their codes, there are other platforms which are of great importance to the open-source community, such as SourceForge and GitLab. Collecting more data from other platforms might yield a more complete list of open-source clinical software.

Secondly, a more comprehensive evaluation is necessary, for example through a system usability measurement with a customized System Usability Scale (SUS) and case studies with clinical researchers or participating practitioners. Then, the usability study will quantitatively assess our application in terms of functionality and user-friendliness. Furthermore, the case study could provide a qualitative evaluation of the application from the perspective of potential users. Specific suggestions for further improvement are the expected results.

Last but not the least, the development of LOCATE is ongoing. A revised version

is being developed. New features will also be added. For instance, natural language processing techniques will be used to process the textual data so that the application can support keyword-based retrieval.

7.6 Conclusions

The primary goal of the paper has been to develop an application that supports the clinical community to easily and confidently locate open-source software. The web application we built offers user-friendly interfaces and are publicly accessible online. Furthermore, its iterative development process ensures the continuous improvement of the application and that the rapid updates of open-source clinical software are incorporated. As the first attempt to link open-source clinical software to literature, we have laid down the groundwork for more research on this topic so that open-source clinical software can be better utilized to contribute to both research and practice.