Information Sciences 448–449 (2018) 36–52

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Learning to answer programming questions with software documentation through social context embedding

Jing Li[a,*], Aixin Sun[a], Zhenchang Xing[b]

[a] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[b] College of Engineering and Computer Science, Australian National University, Australia

ARTICLE INFO

Article history:
Received 31 January 2017
Revised 3 January 2018
Accepted 9 March 2018
Available online 13 March 2018

Keywords:
Community-based question answering
Software documentation
Social context
Neural network

ABSTRACT

Official software documentation provides a comprehensive overview of software usages, but not on specific programming tasks or use cases. Often there is a mismatch between the documentation and a question on a specific programming task because of different wordings. We observe from Stack Overflow that the best answers to programmers' questions often contain links to formal documentation. In this paper, we propose a novel *deep-learning-to-answer* framework, named `QDLinker`, for answering programming questions with software documentation. `QDLinker` learns from the large volume of discussions in community-based question answering site to bridge the semantic gap between programmers' questions and software documentation. Specifically, `QDLinker` learns question-documentation semantic representation from these question answering discussions with a four-layer neural network, and incorporates semantic and content features into a learning-to-rank schema. Our approach does not require manual feature engineering or external resources to infer the degree of relevance between a question and documentation. Through extensive experiments, results show that `QDLinker` effectively answers programming questions with direct links to software documentation. `QDLinker` significantly outperforms the baselines based on traditional retrieval models and Web search services dedicated for software documentation retrieval. The user study shows that `QDLinker` effectively bridges the semantic gap between the intent of a programming question and the content of software documentation.

1. Introduction

For most programming languages and software packages, there exist comprehensive language specifications, Application Programming Interface (API) documentation, and tutorials. Such official documentation[1] provides information about functionality, structure, and parameters, but not on specific issues or specific usage scenarios [31,42]. On the other hand, programmers often face very specific issues which are not explicitly stated in software documentation. For many such issues, software documentation does serve as a good reference for why the issues happen and how to address them. However, it is challenging to use a question as a keyword query to search for relevant software documents. This is because the software

* Corresponding author.
E-mail addresses: jli030@e.ntu.edu.sg (J. Li), axsun@ntu.edu.sg (A. Sun), zhenchang.xing@anu.edu.au (Z. Xing).

[1] The term 'software documentation' refers to the collection of documents consisting of language specification, API documentation, and official tutorial.

https://doi.org/10.1016/j.ins.2018.03.014
0020-0255/

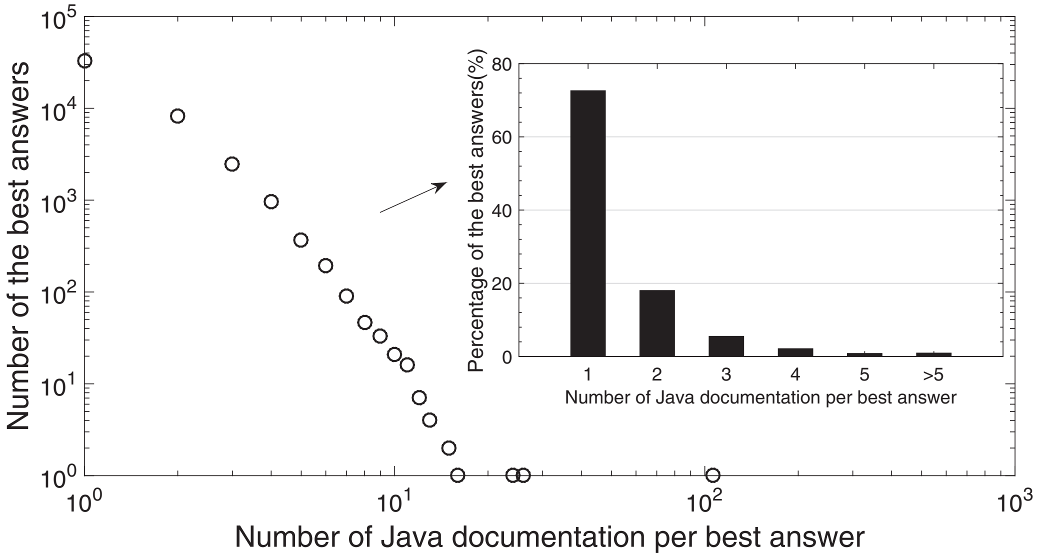

Fig. 1. Distribution of links to Java documentation per best answer, among 45,288 best answers from Stack Overflow. The absolute numbers are plotted in log scale, and the percentages are plotted in bar chart.

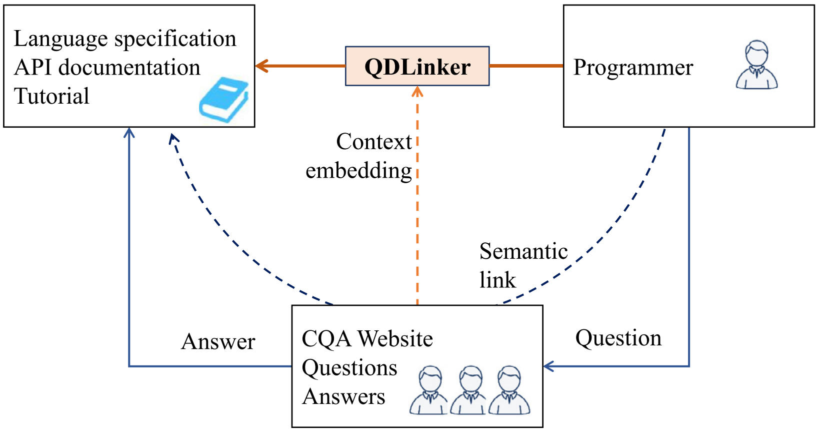

Fig. 2. Overview of `QDLinker`. It directly links programmer's question to formal documentation through embedding the semantic context in CQA. Before `QDLinker`, the semantic links between questions and software documentation are established through the CQA community.

documentation and question are often in different wordings; one is for generic reference and the other is from a specific usage scenario in practice.

With the emergence of Web 2.0 in modern software development, the behavior of developers is changed, in relation to how they search for crowd-generated knowledge to fulfill their needs [21,22,25]. The mismatch between the needs of documentation consumers and the knowledge provided, leads to the overwhelming discussions accumulated at various Community-based Question Answering (CQA) websites such as Quora[2] and Stack Overflow[3]. In these discussions, the community users often refer to software documentation when answering programming questions. From Stack Overflow, we collected 45,288 best answers each contains at least one link to Java official documentation. Fig. 1 plots the distribution of the number of links to Java documentation per best answer, which obeys a power-law distribution. It shows that 72.6% of best answers have exactly one link to Java documentation and fewer than 10% have more than three links. This distribution suggests that for many Java programming questions, there exists a Java official document as a good reference to address the question. The large volume of discussions also create the '*semantic link*' between programmers' questions and software documentation, through the community of programmers, illustrated in Fig. 2.

Posting questions and waiting for answers from other programmers may take much time. The immediate question is: *can we answer a programmer's question by providing a link to the most relevant software documentation?* In this research, we aim

[2] https://www.quora.com/.

[3] http://stackoverflow.com/.