CoolTeD: A tool for co-labeling and visual analysis of textual dataset

Chong Wang\textsuperscript{a,b,*}, Jingwen Jiang\textsuperscript{a}, Maya Daneva\textsuperscript{b}, Marten van Sinderen\textsuperscript{b}

\textsuperscript{a} School of Computer Science, Wuhan University, 430072, China
\textsuperscript{b} SCS Group, University of Twente, 7500AE, the Netherlands

\textbf{A B S T R A C T}

High-quality labeled textual data are reported as an important type of research data in data-driven requirements engineering (RE), especially in automatic mining and analysis of massive textual data produced by software systems. Several tools have been designed to facilitate manual labeling of textual data at different levels of granularity. However, these tools neither aim to provide visualized statistics and analysis of labeled textual data, nor support collaboration among the coders to reduce the time cost in manual labeling and enhance the quality of labeling results. Besides, these tools seldom explicitly serve RE researchers. In this paper, we developed a Web-based labeling tool named CoolTeD (available at http://williamsriver.cn) for collaborative labeling of the textual datasets for RE purposes. Specifically, CoolTeD can be used to: (1) label textual data with the tag category based on ISO 25010 or other user-defined tag categories in a collaborative way; (2) review the labeling results with different confidence levels and contradictory labels; (3) identify contradictory labels and disagreements online; (4) automatically calculate the Cohen’s Kappa coefficient of multiple coders; and (5) visualize the labeling results. The tool demo is available at https://youtube.com/KTVtLenvL.

© 2023 Elsevier B.V. All rights reserved.

\textbf{Code metadata}

\begin{tabular}{|l|l|}
\hline
\textbf{Code metadata description} & \textbf{V2.0} \\
\hline
Permanent link to code/repository used for this code version & N/A \\
Permanent link to Reproducible Capsule & git \\
Legal Code License & List one of the approved licenses \\
Code versioning system used & JavaScript, Python, and Vue \\
Software code languages, tools, and services used & NPM, Node.js, Django \\
Compilation requirements, operating environments and dependencies & williamsriver@whu.edu.cn \\
If available, link to developer documentation/manual & \\
Support email for questions & \\
\hline
\end{tabular}

\* Corresponding author.
E-mail address: cwang@whu.edu.cn (C. Wang).

https://doi.org/10.1016/j.scico.2023.102940
0167-6423/© 2023 Elsevier B.V. All rights reserved.
1. Introduction

1.1. Motivation and significance

Textual data of software has been reported to mainly construct the research datasets used in the requirements engineering (RE for short) community [1], especially in the studies conducting the elicitation and analysis of requirements from massive textual data of software. For example, some studies introduced supervised machine learning algorithms to classify requirements in app reviews [2] [3] [4] [5]. Whereas, other researchers investigated how to improve the classification of app reviews [6] or the trends of app updates [7] from the perspective of RE, by using a small-sized dataset with manual labeled app data. For these RE purposes, high-quality labeled textual data is becoming vital, since the research and experimental results usually depend on the quality of manually labeled data. Usually, the greater number of high-quality labeled data you have, the better results you could get.

Manual labeling is indeed a time-consuming task, and requires multiple coders to work together. First, at least two coders label the dataset independently, after they have relatively consistent understanding of the label categories. Second, the coders need to compare the labeling results to identify agreement and disagreement on the labels. Third, in the case that two coders cannot reach agreements, the researcher who is more senior in terms of expertise has to join in the discussion to help the two coders get consensus on the labeling results. At present, these labeling tasks are mainly performed by exchanging data with labels in Excel files and discussing inconsistent labels in face-to-face meetings. This leads to coordination efforts and possible delays which in turn renders the necessary researchers’ tasks more expensive. To counter these issues, it would be beneficial to develop a tool to facilitate manual labeling of textual dataset for the RE research in an efficient and collaborative way.

This paper is an extended version of our previous conference paper [8] published in the Proceedings of the 29th IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER 2022). Compared to the previous conference version, the main extension of this journal version lies in the following aspects: (1) we improved the CoolTeD tool to support manual labeling with user-defined tag categories; (2) we enriched the descriptions of some CoolTeD’s functionalities, i.e., data reviewing and dataset sharing; and (3) we added an evaluation study of CoolTeD and discussed the limitations of the results.

1.2. Research design

In this paper, we designed and developed a Web-based COllaborative Labeling tool for TExtual Dataset of software systems (CoolTeD for short, available at http://williamsriver.cn). It aims to support labeling, reviewing, and visualizing the labeling results of textual datasets for the RE purposes. More specifically, we envision CoolTeD to be used: (a) to label textual data from the perspective of requirements types based on ISO 25010 or other user-defined categories in a collaborative way, (b) to review the labeling results with different confidence levels, (c) to identify contradictory labels and resolve disagreements online, (d) to automatically calculate the Cohen’s Kappa coefficient of multiple coders, and (e) to visualize the labeling results. Details of these characteristics of CoolTeD will be explained in Section 2.

Furthermore, we followed UTAUT [9] (Unified Theory of Acceptance and Use of Technology) to generate thirteen evaluation questions (available at https://tinyurl.com/4rp36dsb) from four perspectives, i.e., supporting to RE purposes, performance expectancy, effort expectancy, and facilitating conditions. We then conducted a survey of these thirteen questions with volunteer coders to evaluate the CoolTeD tool, as described in Section 3.

2. Software description

2.1. Software architecture

CoolTeD is designed to support collaborative labeling on textual data (artifacts) produced by software systems, such as user reviews, release notes, etc., for the RE purposes. As shown in Fig. 1(a), CoolTeD employs an MVC (Model-View-Controller) architecture. Specifically, the Controller module is responsible to get the requests corresponding to the labeling process that is initiated in CoolTeD, such as choosing or determining a tag for a data item. The Model module consists of data storage and configuration files, both of which are stored in cloud. After updating the data and implementing cloud data storage, CoolTeD will synchronize the View module with the latest data.

2.2. Software functionalities

As a tool supporting labeling textual datasets for the RE purposes, the default label category provided by CoolTeD is proposed from the perspective of requirements types, i.e., functional requirements (FR) and non-functional requirements (NFR). Furthermore, NFR is refined into the eight types defined in ISO 25010 [10]. These eight types are Functional Suitability, Performance Efficiency, Compatibility, Usability, Security, Reliability, Maintainability, and Portability. Besides, CoolTeD allows users to create their own label categories as specific tag configuration files for various RE purposes. These tag configuration files are coded in JSON by specifying the bidirectional mappings between each tag name and its tag type ID. Therefore, CoolTeD
makes it easy for users to create other tag categories and then import these categories by uploading the corresponding tag configuration file into CoolTeD.

Moreover, CoolTeD provides a collaborative and efficient way to help researchers and interested practitioners manually label the textual data of software systems. On one hand, the dataset to be or having been (partially) labeled by one coder can be shared with another coder for co-labeling or with a reviewer for evaluating the labels, by specifying their account names. This improves the efficiency in multi-researcher labeling. On the other hand, CoolTeD provides statistical information and its visualization about the labeled textual dataset. This increases the speed of reporting the labeling results and improves their representation.

The overall process of labeling textual data with CoolTeD is shown in Fig. 1 (b). The main functionalities of CoolTeD supporting this complete labeling process will be explained in the following subsections in detail. The tool demo is available at [https://youtu.be/KTVrLlenvLE](https://youtu.be/KTVrLlenvLE).

Note that only registered users are allowed to use CoolTeD for their labeling or reviewing tasks. Moreover, in CoolTeD, one user acting as both the coder and the reviewer should register for these two roles independently. In turn, the functionalities he/she can perform in CoolTeD rely on the role he/she logs on as.

### 2.2.1. Data labeling

As the main function of CoolTeD, data labeling aims to add an appropriate label to each item of a given dataset, which has been uploaded or shared by a registered coder. As shown in the top of Fig. 1(b), the registered coder needs to perform the following tasks: (a) choose a tag from a specified tag category, (b) to specify the confidence level when labeling a data item, and (c) to provide rationale of choosing this tag for a certain data item. The corresponding screenshot is shown in Fig. 2.

**Choose a tag** As mentioned, the default tag category in CoolTeD is defined from the perspective of requirements types, as shown in Part 1 of Fig. 2. Taking this tag category as an example, we can see that besides FR and NFR, CoolTeD provides the type ‘Others’ for the data items referring to neither FR nor NFR. Since ISO 25010 [10] lists not only the eight types of NFR but also their sub-categories, Part 2 of Fig. 2 dynamically shows the definitions of the corresponding sub-categories for each of the eight NFR types. This can help the coder better understand the meaning of each NFR and suggest the best one or more labels for each data item. Note that all the definitions of sub-categories are cited from ISO 25010 [10].

**Choose a confidence level** When specifying a label to a data item, the coder may be convinced of his/her choose, or be wondering if his/her choose is the most appropriate one. Therefore, the coder should choose a confidence level (from one to five stars, as shown in Part 3 of Fig. 2) to denote the degree of his/her confidence when labeling this data item. Note that in the stage of reviewing labels of a given dataset in Section 2.2.2, the reviewers can use the value of confidence as a threshold of filtering the labeled data items to be checked. Similarly, the coder should specify his/her confidence when labeling the dataset with CoolTeD.

**Provide rationale** To further understand the process of labeling textual dataset of software systems and facilitate the process of resolving disagreements on the labels, CoolTeD allows the coder to record the reason why he/she choose a specific label for a data item. These reasons refer to the rationale linked to the data item with this label, as shown in Part 4 of Fig. 2. Different from the aforementioned two tasks in data labeling, it is optional for the coders to provide rationale for the labeled data item. Nonetheless, the coders are strongly suggested to record the rationale when labeling data items, since this information could supplement the process of decision-making on tags in data reviewing (see Section 2.2.2).
2.2. Data reviewing

In CoolTeD, the process of data reviewing is conducted by the reviewers. Data reviewing serves to evaluate the labels given by different coders, identify the uncertain labels in the dataset, as well as resolve disagreements on the labels of the same dataset. More specifically, data reviewing can be divided into two tasks, i.e., (a) check data items and (b) correct the tag of certain data items if necessary. The corresponding screenshot of CoolTeD is shown in Fig. 3. Each task is detailed below.

Check data items To assure the quality of labeled dataset, CoolTeD allows reviewers to check the labels of the data items in a dataset. These data items are labeled by more than one coder independently, by following the steps of data labeling in Section 2.2.1. Especially, the reviewers can check the labeling details of each labeled data item, including the labels given by different coders, confidence level specified by the coders, and the corresponding rationale. To allow for diverse reviewing scenarios, CoolTeD provides the following three checking strategies.

1) Check All: allows the reviewer to check the labels of all the data items in a dataset, as shown in the Part 1 of Fig. 3. It usually takes the longest time, but gives the most accurate labels to the whole dataset. This strategy is suggested to apply for small size textual datasets.

2) Choose Contradictions: filters the data items with contradictory labels for the reviewers. It is common that different coders may have different understanding of the data items and then give different labels to the same data item, since CoolTeD allows different coders to share and label the same dataset. To reduce the time spent on identifying inconsistent understanding of labels and then getting consensus on the labels of these data items provided by different coders, the reviewer can only focus on the contradictory labels of the same data item by applying this strategy. More specifically,
when selecting the data item with contradictory labels (see Part 2.1 of Fig. 3), the detailed contradictions will be listed in Part 2.2 of Fig. 3 to help reviewers trace the origins of these contradictions. Based on these details shown in Part 3.2 of Fig. 3, the reviewers can introduce a correct label for the data item with contradictory labels and remove inconsistencies in the labeled datasets. This strategy is suggested to apply for the datasets with a larger size, when the reviewer treats ‘reaching agreement’ as priority.

3) **Choose items with Low Confidence:** *filters the data items with lower confidence for the reviewers.* As mentioned in Section 2.2.1, a confidence level must be specified to each data item to be labeled, in order to show whether the coders are convinced with the labels. To enhance the quality of data labeling results, the reviewers need to focus on the uncertain labels provided by the coders, when checking a larger number of labeled data items. For this purpose, this strategy allows the reviewers to choose the data items with low confidence by specifying the range of confidence level in Part 3.1 of Fig. 3. Similar to the strategy of choosing contradictions, this strategy reduces the time spent on reviewing labeled datasets of a larger size.

Note that in practice, the aforementioned three checking strategies can be either applied independently or combined to generate labeled datasets with a high quality. Meanwhile, there is no specified sequence in using these strategies. This means that the RE researchers could creatively combine them in a way that they see fit.

**Determine a tag**  After reviewing the data items with lower confidence or contradictory labels, the reviewers can set labels for these data items. Consequently, these labels given by the reviewers are treated as the final labels of the datasets for further research, such as the inputs of supervised machine learning algorithms for requirements elicitation and analysis.

### 2.3. Dataset sharing

To promote efficient co-labeling on the same dataset, CoolTeD allows users to share one dataset that he/she is working on or is going to work on with other users, rather than exchanging data with their labels in the Excel file. More specifically, one coder can share the dataset that he/she is working on with another coder or a reviewer, and then they can independently label or review the same dataset in CoolTeD. Meanwhile, one coder could accept the invitations by other coders on co-labeling one or more datasets. In CoolTeD, dataset sharing is implemented by specifying the username of the collaborator (either a coder or a reviewer) that you want to share the dataset with, as shown in Fig. 4.

### 2.4. Cohen’s kappa value calculation

In manual data labeling, Cohen’s Kappa coefficient [11] is often used to measure the agreement between two coders who each classify items into several mutually exclusive categories. It reflects the inter-rater reliability of the tags given by these two coders. As a tool for manual data labeling in a collaborative manner, CoolTeD provides automatic calculation of the Cohen’s Kappa value for a dataset which has been labeled by two coders, as shown in Part 1 of Fig. 5.

### 2.5. Data visualization

In order to present the distribution of the labeled data items intuitively, CoolTeD uses pie charts to visualize the statistical results of manual labels on the datasets over the specified label category. As the two exemplary pie charts in Part 3 of Fig. 5 shows, both the number and the proportion of the labels of a specified dataset are listed, referring to different requirements types and the subcategories of NFR, respectively. Note that for the data item with multiple labels, the number of labels might be greater than that of the data items.
3. Perception-based evaluation of CoolTeD

In this section, we designed and conducted an online survey based on UTAUT [9] to evaluate the CoolTeD. The questionnaire used in this survey lists 13 questions from four perspectives, i.e., Supporting to RE purposes (S1-S3), Performance expectancy (S4-S6), Effort expectancy (S7-S10), and Facilitating conditions (S11-S13), with two links to the website of CoolTeD and its demo respectively. On one hand, these 13 questions cover the main functionalities of CoolTeD to be evaluated, such as labeling textual data (e.g., S1), identifying and resolving conflicting labels (e.g., S2 and S6), collaboration in labeling tasks (e.g., S5), etc. On the other hand, these questions are designed to make the volunteers take different roles in the evaluation. For instance, S1 is designed for the taggers to evaluate data labeling, and S2 is for the reviewers to identify and resolve conflicting labels. The questionnaire is available at https://bit.ly/3QDvzgk.

The online questionnaire was delivered to 12 students in the middle of August, 2022. These volunteer students major in software engineering at Wuhan University and had some experience in data labeling, which is one of their current or previous RE-related research tasks. In order to facilitate the online training of CoolTeD, we briefly introduced the main functionalities of CoolTeD when sending the invitations, and then familiarized the participants with the demo to learn how to use the CoolTeD tool for their labeling tasks. Plus, they were told to contact us if they have any problem on labeling with CoolTeD. Finally, ten respondents out of the 12 completed the survey within one week. Specifically, five of these 10 students are undergraduate students, four are PhD students, and one is a master student. These volunteered students evaluated the tool for their RE-related research tasks, and teamwork was not required in this evaluation study. The nature of the task they completed is to evaluate the main functionalities of CoolTeD, and these functionalities have been covered by the questions in the survey.

Fig. 6 shows the detailed results of the questionnaire-based survey on the CoolTeD. We found that all the 10 respondents (strongly) agree to recommend CoolTeD for requirements elicitation and analysis, because it facilitated the whole process of textual data labeling, including individual labeling as well as conflict resolution among multiple coders. In addition, all these 10 respondents (strongly) agree that CoolTeD not only brings convenience to the RE tasks, but increases the speed in either textual data labeling or resolving conflicting labels.

In regard to the effort expectancy, 90% of the respondents (strongly) agree that the interaction with CoolTeD is clear. Whereas, all these 10 respondents (strongly) agree that the CoolTeD is easy to use, and that the visualization of the labeling results is very useful. In addition, most of the 10 respondents think that there are enough resources to use CoolTeD, including the assistance from the developers of CoolTeD.

4. Discussion

4.1. Comparison of CoolTED and other tools

Several tools [12][13][14] have been designed to facilitate manual labeling on textual data on different levels of granularity, as shown in Table 1.
More specifically, KCAT [12] (Knowledge-Constraint Typing Annotation Tool) was designed to provide an efficient Annotator Client to accelerate the annotation process and a comprehensive Manager Module to analyze crowdsourcing annotations. KCAT can generate an accuracy matrix to measure the consistency among multiple annotators. Similarly, our CoolTeD not only speeds up the annotation process by preparing the tag category, but simplifies the analysis process with the visualization of the labeling results. In contrast to KCAT, CoolTeD implements the calculation of the Cohen’s Kappa value, rather than the accuracy matrix, to measure the agreement of two coders. Unlike CoolTeD, KCAT is not a specified annotation tool for the RE community, which is the main difference between these two tools.

Considering MAE (Multi-purpose Annotation Environment) and MAI (Multi-document Adjudication Interface) [13] designed by Stubbs et al., they are lightweight annotation and adjudication tools for creating corpus. To create a simple annotation task, MAE&MAI needs the task definitions of annotation tags and the attributes in the form of Document Type Definition (DTD) files. Especially, DTD files are one of the inputs of both MAE and MAI. This means that in MAE&MAI, the same file should be uploaded twice to perform text annotation and reviewing task of a single text file. Compared to MAE&MAI, labeling results of the textual datasets are stored in the cloud server. Therefore, either the specified tag category or the textual dataset is required to upload only once.
Finally, WebAnnotator [14] is proposed as a Firefox extension to annotate text and images of the Web page. This tool keeps the rendering of HTML while annotating, and these generated annotations are stored as HTML tags. As a result, it is inconvenient to analyze the annotating results. Differently, the labeling results in CoolTeD are stored as JSON data objects in the Web server. In addition, CoolTeD visualizes the distribution of labeling results as pie charts.

4.2. Impact

As a Web-based tool for manual labeling on textual dataset of software systems, CoolTeD provides the following five features to support collaborative labeling conducted by multiple users.

4.2.1. Supporting full spectrum of manual labeling for RE purposes

CoolTeD gets started from importing the dataset to be labeled, and then it moves to label data with the default or pre-uploaded tag categories by the coders, review and decide the labels by the reviewers, as well as calculate the Cohen's Kappa value of two independent coders, and finally ends in visualizing the labeling results in various perspectives. That is, CoolTeD supports the full spectrum of manual data labeling. As Fig. 6 shows, the results of the survey on CoolTeD indicate that our tool is useful in requirements elicitation, analysis, and negotiations activities.

4.2.2. Collaborative labeling

CoolTeD enables one coder to share the dataset that he/she is working on with another coder, so that they can label the same dataset independently. Meanwhile, he/she can accept the invitations from other coders to co-labeling one or more textual datasets. In CoolTeD, dataset sharing can be conducted by simply specifying the username of the coder that you want to share the dataset with, as shown in Fig. 4. This also speeds up the process of multi-coder labeling on one dataset.

In addition, the coders and reviewers can work together easily when using CoolTeD. As mentioned in Section 2.2.1, the coders can provide their confidence level and rationale in the process of data labeling. This means that all the information relevant to the data items and their labels are stored in the Web server, and then can be reviewed, evaluated, or corrected by the reviewers.

4.2.3. Quality assurance of labeling results

In CoolTeD, one mandatory task (i.e., to choose a confidence level for each data item as described in Section 2.2.1) and one optional task (i.e., to provide rationale for each label as detailed in Section 2.2.1) are designed to record the coder’s current understanding of the data item and the reasons of choosing this tag. Both the value of the confidence level and the rationale attached to the labeled data item can be further employed as the supplementary information for the reviewers to check the labels provided by multiple coders and evaluate the appropriateness of these manual labels.

4.2.4. Data tracing

In CoolTeD, the imported textual dataset, each data item in the dataset, and the label of each data item have their unique global IDs. Moreover, CoolTeD records the information about the operators and operations involved in the labeling and reviewing process of textual datasets, such as which individual coder (co-)labels the data item and how many labels are linked to the specified data item. In this way, CoolTeD makes it possible to trace either the process of data labeling and reviewing or the artifacts (e.g., the labels of the same data item provided by the coders and reviewer, respectively) involved in this process.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>fine-grained entity typing annotation</td>
<td>corpus creation</td>
<td>text annotation</td>
<td>collaborative text annotation for RE</td>
</tr>
<tr>
<td>Is the tool Web-based?</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Does the tool support</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>statistic analysis on</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>the labeled data?</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Is it needed to provide annotation schema?</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Does the tool support collaborative labeling by multi-users?</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Does the tool aim to support RE activities?</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
4.2.5. Reaching consistent labels efficiently

As for collaborative labeling by multiple coders, CoolTeD enables to reach a consistent understanding of the labeling results in an easy and efficient way. First, the label categories in CoolTeD are the common and consistent understanding among the specific coders and reviewers. As already indicated, the default label category in CoolTeD is adopted from ISO 25010. This widely used and accepted standard in the RE community helps bridge the understanding gap among coders and reviewers for RE purposes. Second, CoolTeD enables reviewers to find out different understandings of the same data item quickly – to choose the “Contradiction” in Fig. 3. That is, in CoolTeD, contradictory labels of the same data items can be filtered automatically, rather than manually comparing the labeling results of two coders.

5. Limitations

This empirical have some limitations according to the guidelines in [15]. First, regarding the design objective of CoolTeD, it mainly aims to facilitate the manual labeling tasks in RE research, and the default tag category in CoolTeD is adopted from ISO 25010. One might think that including only one standard, might be limiting. However, as reported in Section 3, all the 10 respondents think CoolTeD is useful in manual labeling and solving disagreements on labels, which are parts of requirements elicitation, analysis, and negotiation activities. Plus, the respondents are willing to recommend CoolTeD other fellow researchers for these RE tasks. Although CoolTeD is designed to support RE relevant research, it allows users to upload other tag categories for different RE purposes or non-RE purposes. This partially increases the more general applicability of CoolTeD beyond the contexts in which ISO 25010 is used.

Second, as a newborn tool for co-labeling the textual dataset for the RE purposes, CoolTeD is easy to learn and use, as reported in the answers to S9 and S10 in Fig. 6. Due to its recency, more detailed instruction and help manuals are expected to be made available in order to assist registered users and attract more users.

Third, as mentioned in Section 3, the respondents of the survey on CoolTeD are mainly from academia and the sample size is limited. The results of this survey may be changed if introducing more respondents from both academia and industry, which will be conducted in our next step.

6. Conclusions and future work

This paper introduced CoolTeD, a Web-based collaborative labeling tool designed and developed for labeling textual dataset of software systems. It aims to label contextual datasets for the RE purpose. Its strength is twofold: on one hand, CoolTeD can support the full spectrum of manual labeling, including collaborative labeling by coders, reviewing labels by reviewers, and visualization of labeling results. On the other hand, the confidence of coders as well as the rationale of choosing the labels is recorded in CoolTeD to assure the quality of labeling results. For researchers and research-minded practitioners, we think that CoolTeD is a candidate tool worthwhile considering, due to its ability to facilitate the manual labeling of textual data in a more efficient and effective manner.

In the near future, potential improvements of CoolTeD could be designed and implemented to support exporting labeling results and customizing tag categories more flexibly.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant Nos. 61702378, 61972292, and 61832014.

References


