SCIENCE CHINA
Information Sciences

• LETTER •

HAF: A Hybrid Annotation Framework Based on Expert Knowledge and Learning Technique

Zhixing LI[1,2], Yue YU[1,2*], Tao WANG[1,2], Gang YIN[1,2], Xinjun MAO[2] & Huaimin WANG[1,2]

[1]*Key Laboratory of Parallel and Distributed Computing ;*
[2]*College of Computer, National University of Defense Technology, Changsha* 410073*, China*

Citation Zhixing LI, Yue YU*, Tao WANG, Gang YIN, Xinjun MAO, Huaimin WANG. HAF: A Hybrid Annotation Framework Based on Expert Knowledge and Learning Technique . Sci China Inf Sci, for review

Dear editor,
The increasing awareness of the potential value hidden in data has resulted in many data mining studies being conducted. In the domain of software engineering, for example, developers' behavioral data and code review data have been leveraged in social coding sites to automatically recommend relevant projects [1] and candidate reviewers [2, 3]. Additionally, the deployment data and test data of software projects have been analyzed to examine the influence of tools and third-party services on artifact evaluations [4, 5]. Such studies can help software practitioners improve the quality and efficiency of software development and crowd collaboration. In practice, a large number of data mining studies have been conducted based on labeled datasets and larger datasets could achieve more convincing research results. Each item in a labeled dataset can be represented as a tuple $< sample, label >$. For example, in the case of handwritten digit recognition, *sample* is a handwritten image, which is usually represented as a set of pixel values in grayscale and *label* is the actual digit value written in the image. However, unlike image recognition, which has public datasets such as MINIST [1)] and ImageNet [2)], some research fields do not have readymade labeled datasets. Instead, researchers may be faced with a raw dataset for which there is not even a predefined taxonomy schema. For example, researchers may wish to explore the discussion comments in social coding sites and to mine the distribution differences in developers' emotions across different programming languages. However, discussion comments on such sites are not labeled with any emotion tags. Consequently, researchers have to first manually label the discussion comments and, if necessary, construct a taxonomy of emotions in advance. However, manually annotating a large dataset is a time-consuming and tedious task. Moreover, the annotation process needs to be carefully designed and conducted because the quantity and quality of an annotated dataset has a significant influence on the research results.

To this end, we propose a ***H**ybrid **A**nnotation **F**ramework* (HAF) based on our research experience [6, 7] and a review of prior related work [8, 9]. The annotation framework combines expert knowledge and a learning technique to semiautomatically produce large-scale labeled datasets. Expert knowledge is obtained from a manual inspection of the raw dataset and can be used as a heuristic annotation rule. A learning technique is applied to find hidden factors or patterns that cannot be easily recognized by human beings but can improve the performance of the annotation. The combined application of expert knowledge and learning technique is expected to achieve a more effective and efficient automatic annotation of large-scale datasets. Note that our annotation frame-

* Corresponding author (email: yuyue@nudt.edu.cn)
1) http://yann.lecun.com/exdb/mnist/
2) http://www.image-net.org

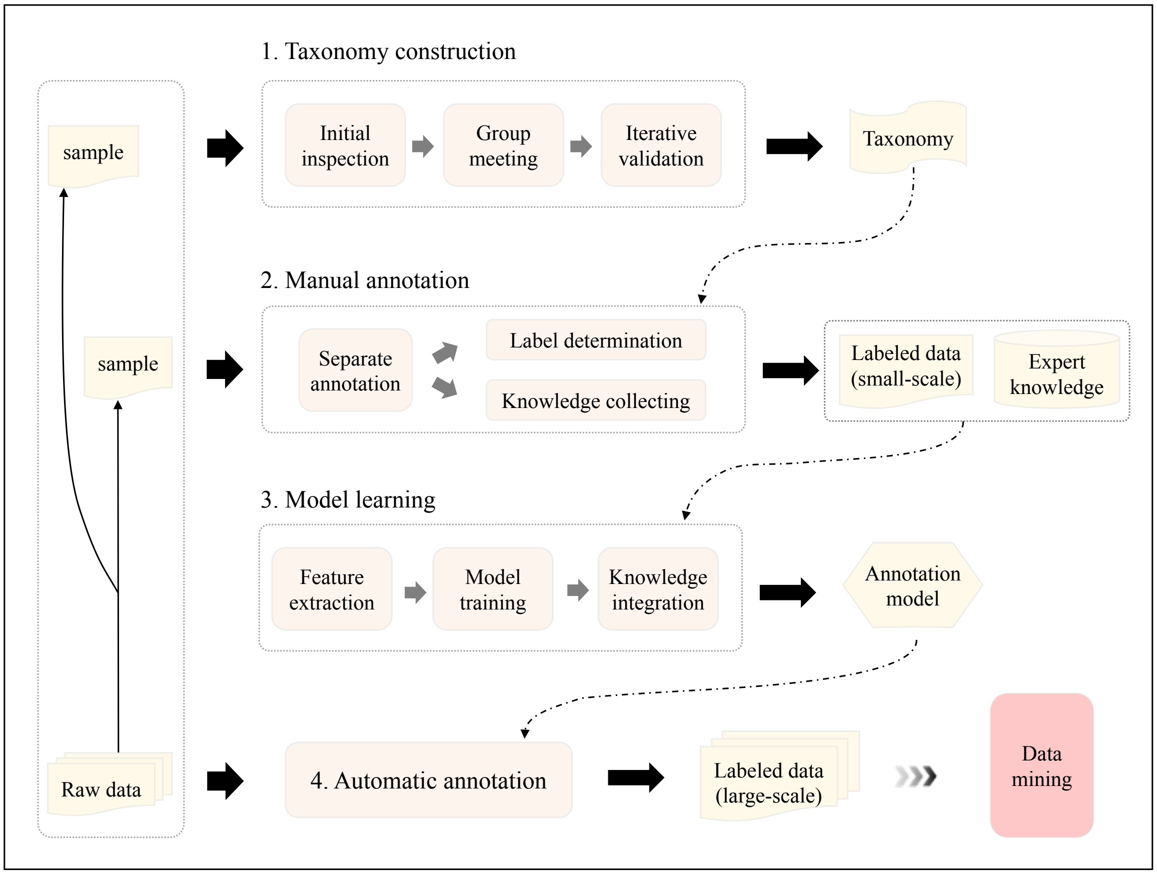

Figure 1 An overview of HAF

work is generic and can be applied in different research contexts. As shown in Figure 1, HAF includes the following four stages.

Stage 1 - Taxonomy construction. As discussed, a taxonomy that labels the raw data into different categories is not always available in some contexts; therefore, HAF starts with the construction of taxonomy.

In practice, *open coding* is a popular method used to define a taxonomy and primarily involves three steps.

- *Initial inspection*: Open coding is usually conducted by multiple participants. First, each participant separately inspects a random sample of the raw data and constructs their own initial taxonomies.
- *Group meeting.* All of the participants gather together and discuss their findings in a group meeting. The topic of the meeting is to resolve the differences and inconsistencies in the initial taxonomies and to reach an agreement on a concurrent taxonomy.
- *Iterative validation.* With the concurrent taxonomy, the participants continue to label a random sample of the raw data to validate the taxonomy and introduce necessary refinements. Finally, a formal taxonomy is constructed when the taxonomy becomes relatively stable after several rounds of iterative validation.

Stage 2 - Manual annotation. In this stage, a random sample of the raw data is manually labeled according to the taxonomy defined in the previous stage. However, manual annotation is tedious and time-consuming; therefore, we suggest that the manual annotation be assigned to individuals who care the most about the work and have abundant time to undertake the annotation.

- *Separate annotation.* When multiple annotators are involved, it is better to adopt the process of separate annotation and to have each data item labeled by more than one annotator.
- *Label determination.* If the annotators label the same data item differently, the final label of the data item needs to be determined by some sort of agreement mechanism, such as a majority vote of the annotators or excluding that data item.
- *Knowledge collection.* Moreover, the annotators should make necessary notes and summarize the experience by annotating a data item with a specific label. The summarized experience forms

the expert knowledge.

In practice, the size of the manually labeled dataset can be determined by two observations: a) the expert knowledge is stable and new knowledge can no longer be obtained from manual annotation and b) the performance of the automatic annotation model introduced in *Stage 3* reaches its peak and can no longer be improved by the introduction of more data.

Stage 3 - Model learning. Manual annotation can only produce a small-scale dataset of high quality. To power data mining on a large-scale dataset, an automatic annotation model needs to be trained based on the manually labeled data and the collected expert knowledge.

- *Feature extraction.* The first step to train a learning model is to select and extract a set of features. Feature selection can be guided by the researchers' experience and optimized by engineering practices. After feature extraction, each training sample is represented as a tuple consisting of a feature vector and a label.
- *Model training.* With the training dataset, some kind of learning algorithm is applied to train the annotation model. The choice of learning algorithm depends on the research domain, *e.g.,* the support vector machine technique usually achieves excellent performance for text classification.
- *Knowledge integration.* Expert knowledge can be integrated into the annotation model in two ways: a) the annotation of expert knowledge is treated as a feature of the model, or b) expert knowledge supplements and updates the annotation produced by the model.

Stage 4 - Automatic annotation. After the annotation model has been trained, it needs to be tested on the manually annotated dataset. Since the trustworthiness of future research depends heavily on the quality of the annotated data, the model can only be used to automatically annotate the raw data if it achieves a reasonable performance. In comparison with the manual annotation, the automatic annotation will produce a large-scale labeled dataset that could permit more convincing data mining studies.

Conclusions. This study proposed HAF, *i.e.*, hybrid annotation framework, to semiautomatically produce a large-scale labeled dataset. HAF starts with the manual inspection of the raw data, and a taxonomy of data labels is constructed in this stage. Then, the individuals manually label a small-scale sample of the raw dataset according to the taxonomy and their understanding and experience of how to annotate a data item with a specific label are summarized as expert knowledge. The manually labeled dataset is used as the training dataset to train an automatic annotation model and the model is integrated with and enhanced by the expert knowledge. Finally, the annotation model can be used to automatically annotate all the raw data and produce a large-scale labeled dataset to support more convincing research. In summary, we believe HAF can provide a practical reference for researchers engaged in data mining.

Acknowledgements This work was supported by the National Grand R&D Plan (Grant No. 2018YFB1004202) and the National Natural Science Foundation of China (Grant Nos. 61702534).

References

1 Xu W, Sun X, Hu J, et al. REPERSP: recommending personalized software projects on GitHub. In: Proceedings of 2017 IEEE International Conference on Software Maintenance and Evolution. IEEE, 2017: 648-652.

2 Yu Y, Wang H, Yin G, et al. Reviewer recommendation for pull-requests in github: What can we learn from code review and bug assignment?. Information and Software Technology, 2016, 74: 204-218.

3 Jiang J, Yang Y, He J, et al. Who should comment on this pull request? Analyzing attributes for more accurate commenter recommendation in pull-based development. Information and Software Technology, 2017, 84: 48-62.

4 Yu Y, Yin G, Wang T, et al. Determinants of pull-based development in the context of continuous integration. Science China Information Sciences, 2016, 59(8): 080104.

5 Zhang Y, Wang H, Yin G, et al. Social media in GitHub: the role of@-mention in assisting software development. Science China Information Sciences, 2017, 60(3): 032102.

6 Li Z X, Yu Y, Yin G, et al. What Are They Talking About? Analyzing Code Reviews in Pull-Based Development Model. Journal of Computer Science and Technology, 2017, 32(6): 1060-1075.

7 Fan Q, Yu Y, Yin G, et al. Where is the road for issue reports classification based on text mining? In: Proceedings of 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. IEEE, 2017: 121-130.

8 Zhou Y, Tong Y, Gu R, et al. Combining text mining and data mining for bug report classification. Journal of Software: Evolution and Process, 2016, 28(3): 150-176.

9 Gachechiladze D, Lanubile F, Novielli N, et al. Anger and its direction in collaborative software development. In: Proceedings of IEEE/ACM 39th International Conference on Software Engineering: New Ideas and Emerging Technologies Results Track. IEEE, 2017: 11-14.