Supplementary Material
QuLog: Data-Driven Approach for Log Instruction Quality Assessment

Anonymous Author(s)

ABSTRACT

This is a supplement material for the paper QuLog: Data-Driven Approach for Log Instruction Quality Assessment. It contains key issues and details in addition to the main manuscript. This paper is accepted for publication at International Conference on Program Comprehension 2022 (ICPC 2022).

KEYWORDS

log quality, deep learning, program comprehension

ACM Reference Format:

Anonymous Author(s). 2022. Supplementary Material QuLog: Data-Driven Approach for Log Instruction Quality Assessment. In *Proceedings of The 30th International Conference on Program Comprehension (ICPC 2022).* ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 EXAMPLES OF QULOG OUTPERFORMS/UNDERPERFORMS THE BASELINES

One set of examples where we find that QuLog's log level prediction outperforms other methods is in the logs like "created with buffersize=* and maxpoolsize=*" (with original log level INFO, QuLog prediction is "info", "DeepLV" and "BERT_SVM" are predicting "warning") and "stopping the wall procedure store, isabort= | (self aborting)" (with original log level INFO, QuLog prediction is "info", "DeepLV" and "BERT_SVM" are predicting "error"). These two messages contain words like "maxpoolsize", "buffersize" or "isabort" which are not standard in general language. Therefore, the log representation by general language models is inferior in comparison to QuLog, which directly learns log representations from the static text. Since QuLog, is directly trained on log-specific words it can learn better representations and relate the words to the correct target.

When considering the linguistic comparison, as seen by the results, QuLog and BERT_SVM are performing similarly. We find one consistent example where QuLog performs better than SVM, i.e., when HBase is used for testing QuLog correctly predicts the linguistic group "VERB VERB" as insufficient. We hypothesise that QuLog can extrapolate this information having the groups "VERB" and "VERB NOUN VERB" as insufficient during training. The two baselines, SVM and RF are performing slightly differently on the same representation, we consider the observed difference to reside in the type of decision boundary they fit. SVMs are characterized to perform better in very high dimensional spaces (because of the properties of such spaces). RF is characterized by diverse volume pockets in different regions. Since the samples in the high dimensions are very far apart from one another the pocket regions created by RF lead to more miss-classifications.

2 RULE-BASED APPROACH AND QULOG

The process of identification of sufficient linguistic structure was conducted as follows. Two human annotators examined the linguistic structure and the raw static texts organized by linguistic groups. A linguistic group is a set of static text instructions'. A linguistic group is identified by the sequence of POS tags. Each human annotator examined the individual static text within each group (within each of the randomly sampled 361 groups). Since each log message should describe an event verbosely and convey sufficient information on one side, and following the maxim of quality and quantity for short texts from general language properties, on the other side, the static text should have minimal linguistic structure for sufficient expression of the information. By examining the raw static text within each group the annotators relate their understanding of the sufficient information a log message has. They assign labels 0 for "linguistically sufficient" or 1 for the "linguistically insufficient" group. Then the labels provided by the two human annotators are compared, and the linguistic groups with overlapping labels are retained. Each of the static texts is then labeled with the label of the linguistic group associated with it. However, the model is trained using the linguistic group representation of the static text obtained by the pos tag. The labels can be used as rules what is a linguistically good and bad static text.

The generated rules can be used to detect logs with sufficient and insufficient structure (that is how they are generated). Whenever we have a new system, we calculate their linguistic representations and match them against the rules. Any match is considered as a log with insufficient quality. For example, if we find a static text with "NOUN NOUN" from the new system, we can say that the static text is of "insufficient" quality. However, we found examples, where QuLog maybe interpolates between two nearby rules, and correctly identify a rule from a new project that was not part of the training data. For example, when HBase is in the test set, (the rules in the training set are extracted from other systems), the rules do not cover the linguistically insufficient group "VERB VERB". In contrast, QuLog correctly predicts this group as "insufficient" because of its similarities with the linguistically insufficient groups of "VERB" and 'VERB NOUN VERB' (that are slightly different from "VERB VERB"). Since QuLog has strong performance, and can interpolate between rules we opt for the given design. Nevertheless,

ICPC 2022, May 21–22, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

there is still much space for improving the input training dataset that will lead to new insights.

3 LINGUISTIC QUALITY ASSESSMENT ADDITIONAL EVIDENCE

The idea for the "sufficient linguistic quality" is built around the Jira issue ZOOKEEPER-2126. This issue represents the fact that sufficient event information should have a minimal linguistic structure. By enriching the event description with additional linguistic properties, the log messages are easier for reading, comprehension, and contain sufficient verbose information.

There are different issues where similar observations can be made. For example in ZOOKEEPER-3659 it is reported that WatchManagerFactory log is not sufficiently readable. The fix of this issue is to change the prior statitc text *Using org.apache.zookeeper.server.watch .WatchManager as watch manager*, into *dataWatches is using org.apache .zookeeper.server.watch.WatchManager as watch manager.*. From the linguistic perspective this means that the linguistic structure ['VERB', 'PUNCT', 'ADP', 'NOUN', 'NOUN'] is transformed into [**'NOUN', 'AUX',** 'VERB', 'PUNCT', 'ADP', 'NOUN', 'NOUN']. The additional linguistic concepts improve the comprehensability of the event.

Similarly in the Jira issue Zookeeper-259 despite the correction of the log levels, the text in the log instructions is changed as well. One example is the change on lines 524, through 526. Specifically, the prior static text "Got ping sessionid:0x" was replaced with "Got ping response for sessionid:0x". Linguistically speaking this means that the text is changed from ['AUX', 'VERB', 'NOUN'] into ['VERB', 'NOUN', 'NOUN', 'ADP', 'NOUN'] which improves the comprehensibility and makes the log line easer for the operators to understand.

4 ADDITIONAL EXPERIMENTS

4.1 GitHub Evaluation

4.1.1 Experimental Design. We considered QuLog* log level assignment approach because it is system-agnostic. We considered the two-class IE scenario for log level assignment assessment. The experiment is designed as follows. We start with the 100 repositories collected during the data collection procedure. We randomly sampled 60% of the repositories for training, 20% for validation and 20% for evaluation. To reduce the variance of the results due to the random repositories selection, we repeated the sampling procedure 30 times and reported the average results. To assess the correctness of the decisions, we used the F_1 score for both properties.

4.1.2 Results and Discussion. Figure 1 depicts the results of the evaluation. The left part depicts the scores. It is observed that all of the scores are high (in accordance with our prior evaluation). Therefore, the learned models correctly predict the log levels. The right part of the image represents the average of the linguistic and log level scores. It is seen that the representative software systems Flink, Cassandra and Kafka achieve high average scores (all are higher than most of the other systems). Notably, this single number can be seen as a model-driven KPI score for the quality of the log instructions.

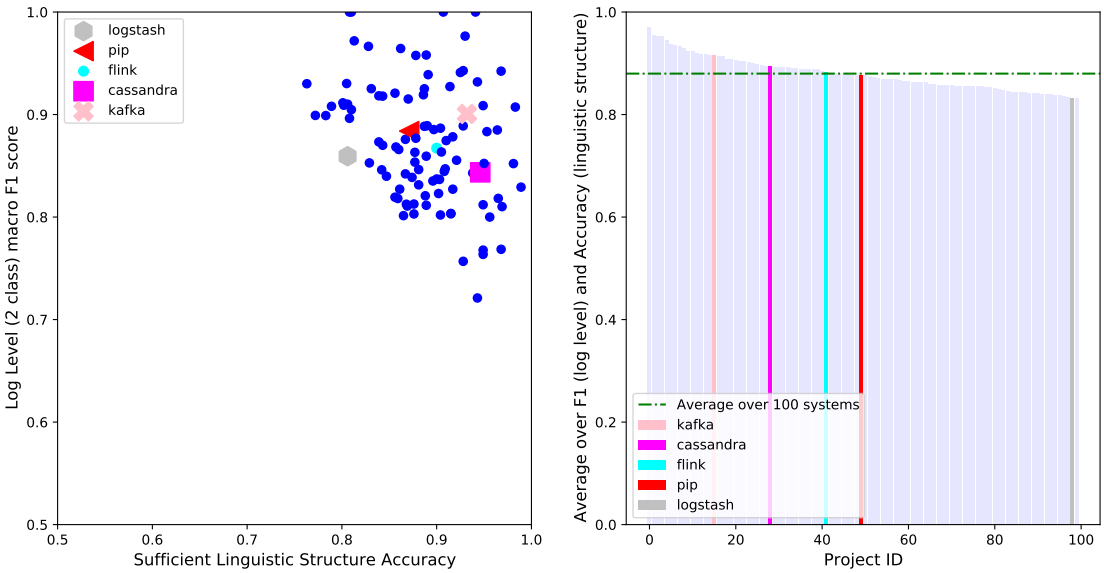

Figure 1: QuLog evaluation on 100 GitHub software systems.