LogBERT: A BERT-Driven Approach to Log Instruction Quality Assessment

Team -3
Aayush Panchal, Sam Singh, Sushant Borse

OUTLINE

01 Context

02 Problem

03 Solution

04 Related Work

05 Approach Overview

06 Results & Analysis

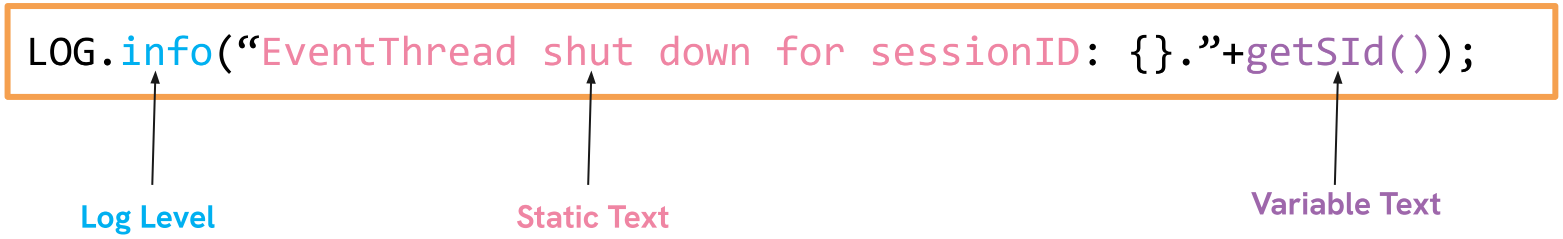
Lloge
lIcg
lIods
trace
rfoc
debog
warring
vaiiig
info
lfi
lrfio
efro
erer

01

▪ Context

Log Messages

- A log is a string that provides contextual information about a process during its runtime.
- A log is composed of three parts.



```
LOG.info(“EventThread shut down for sessionID: {}.”+getSId());
```

Why are logs important?

- Bridging the gap between code developers and system operators.
- Facilitating communication through log messages for monitoring processes and troubleshooting errors.
- Enabling system operators to perform these tasks without directly interacting with the underlying code.
- Inefficiently crafted log instructions have the potential to cause confusion, impede the effectiveness of troubleshooting processes, and escalate maintenance costs.

Log Message Quality

Log level :

- **INFO -** An event happened, the event is purely informative and can be ignored during normal operations.
  Example: API request to /api/v1/users completed successfully
- **ERROR -** One or more functionalities are not working, preventing some functionalities from working correctly.
  Example: Unhandled exception: division by zero.
- **WARN -** Unexpected behavior happened inside the application, but it is continuing its work and the key business features are operating as expected.
  Example: Disk usage warning

Linguistic Sufficiency:

- **Sufficient -** Message has enough information.
  Example: "Failed to retrieve data from API"
- **Insufficient -** Message does not have enough information.
  Example: “Failed”

02

Problem

Challenges in Logging

- There is a lack of comprehensive and consistent guidelines for developers on logging best practices.
- Developers make frequent log-related commits. In many cases, log messages are written as “after-thoughts” after a failure occurs.
- Various logging libraries, like syslog and log4j, offer logging interfaces but they are language-dependent and leave decisions regarding what to log to developers.
- There is a need for a programming language-agnostic automated approach to log quality assessment.

Research Questions

1. To what extent can an automated approach accurately determine the appropriate **log level** in log instruction quality assessment?
2. What is the effectiveness of the proposed automated methodology in accurately determining the **linguistic sufficiency** of log messages?
3. How does the efficacy of the proposed approach in log instruction quality assessment **compare** to that of the existing methodologies?

Uoge
lIg
llods
trace
rfoc
debog
warring
vaing
info
lft
lrfro
efro
erer

03

■ Solution

LogBERT

- LogBERT is an automated, programming language agnostic methodology for assessing log quality, harnessing the power of pre-trained transformers.
- The approach analyzes the **static text** of log messages, regardless of the programming language used.
- Using the static text, it is capable of identifying the appropriate **log level** - info, warning, or error, and **linguistic sufficiency** - sufficient and insufficient.

04

▪Related Work

QuLog: Data-Driven Approach for Log Instruction Quality Assessment

- QuLog is the foundation for LogBERT.
- In QuLog, the authors introduce the approach of using static text of log messages for analysis. Using a data-driven study, the prove that static text is enough to predict both log level and linguistic sufficiency.
- QuLog uses two custom transformer based neural networks, one for identifying log levels and the other for linguistic sufficiency.
- In addition to this, the authors incorporate explainable AI to make the decision-making process transparent and understandable.
- We follow QuLog's approach very closely with deviation in the use of pre-trained models and tokenization.

05

▪ Approach Overview

Data Collection

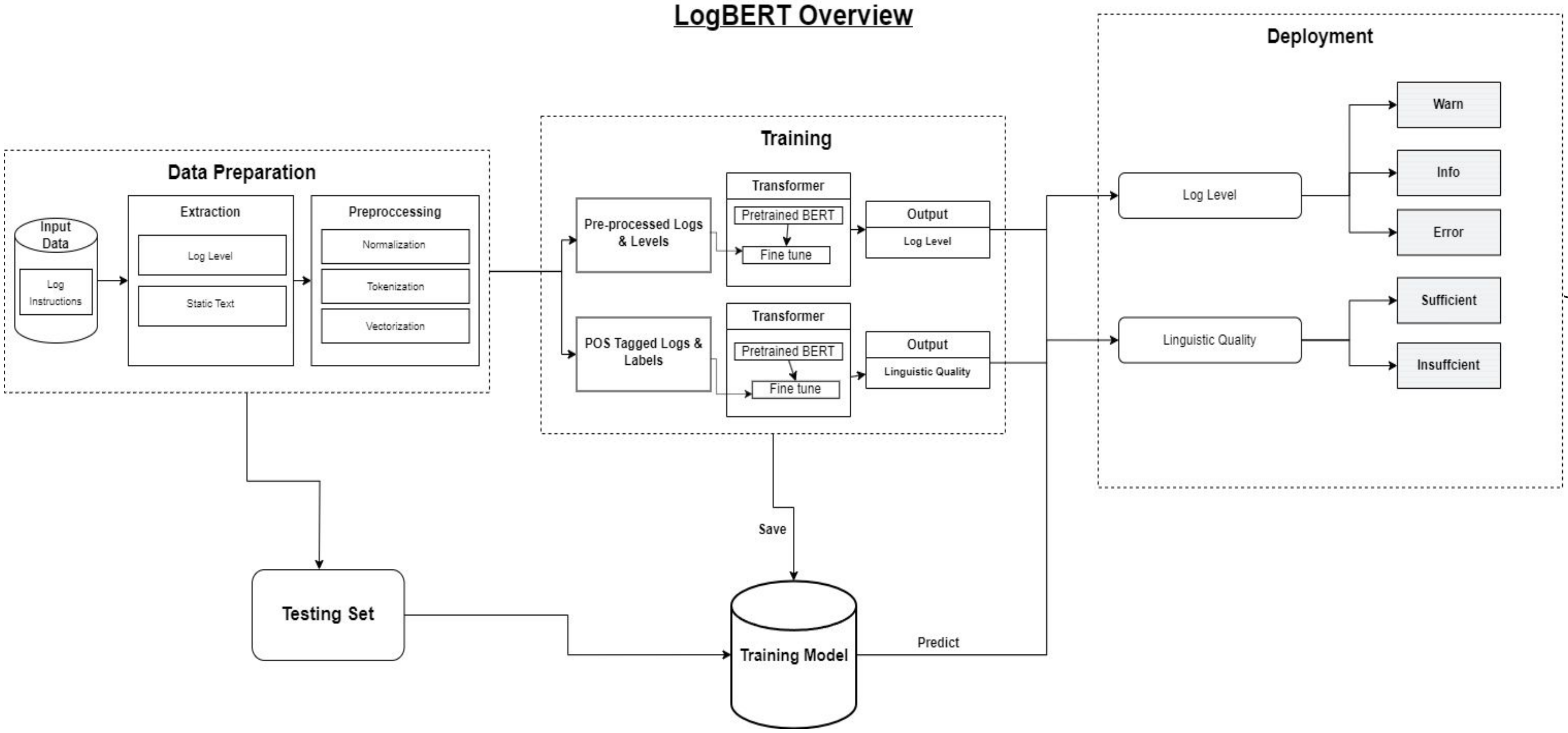
- We used the data source provided by the authors of QuLog.
- Log messages were extracted from 9 popular open source projects.
- The static text and log messages were extracted from each log message.
- The authors of the paper manually labelled randomly sampled static text as sufficient or insufficient.

Software systems studied
HBase
JMeter
Zookeeper
Cassandra
ElasticSearch
Flink
Kafka
Karaf
wicket

static_text	sufficiency
subscribed pattern	insufficient
write failed	sufficient

static_text	log_level
Exception inside handler	error
Loading directories from HDFS	info
Table * does not exist	warn

Architecture

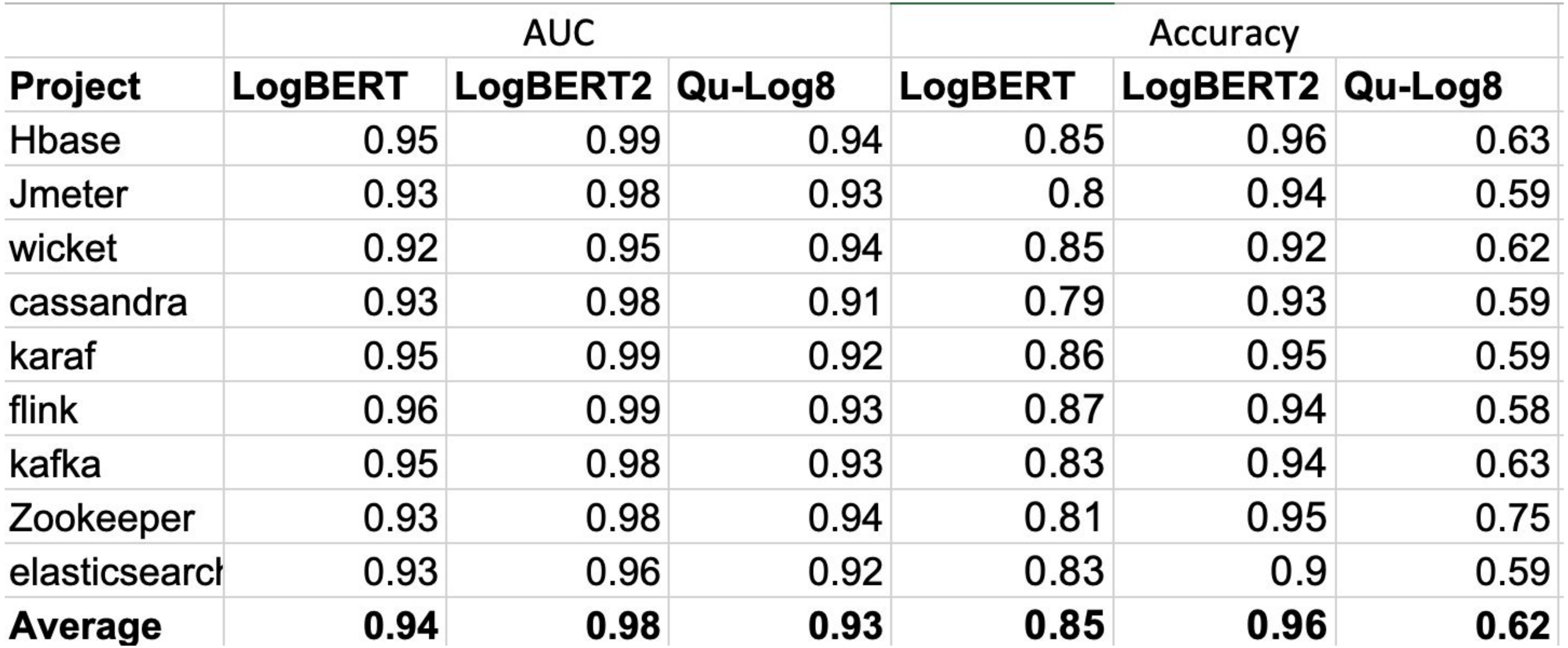


06

Results & Analysis

Log Level Results

- LogBERT has **23%** increase in accuracy compared to QuLog.
- When used for binary classification (info & error), LogBert2 performs **34%** better than QuLog, 11% better than LogBERT's multi-level classification.



	AUC			Accuracy		
Project	**LogBERT**	**LogBERT2**	**Qu-Log8**	**LogBERT**	**LogBERT2**	**Qu-Log8**
Hbase	0.95	0.99	0.94	0.85	0.96	0.63
Jmeter	0.93	0.98	0.93	0.8	0.94	0.59
wicket	0.92	0.95	0.94	0.85	0.92	0.62
cassandra	0.93	0.98	0.91	0.79	0.93	0.59
karaf	0.95	0.99	0.92	0.86	0.95	0.59
flink	0.96	0.99	0.93	0.87	0.94	0.58
kafka	0.95	0.98	0.93	0.83	0.94	0.63
Zookeeper	0.93	0.98	0.94	0.81	0.95	0.75
elasticsearch	0.93	0.96	0.92	0.83	0.9	0.59
Average	**0.94**	**0.98**	**0.93**	**0.85**	**0.96**	**0.62**

Linguistic Structure Results

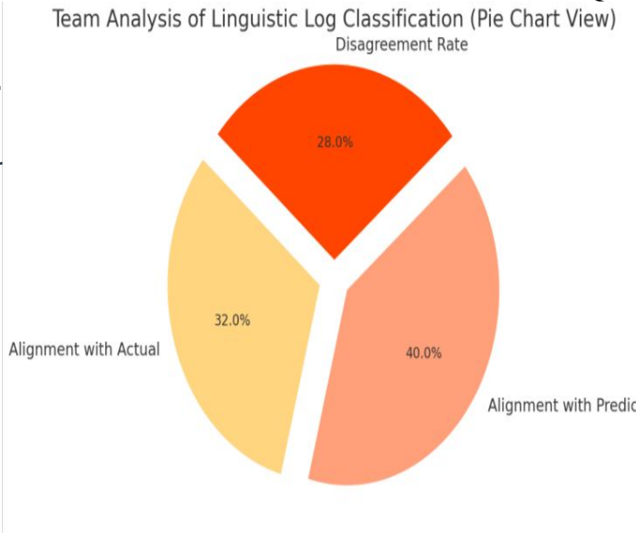
- For linguistic structure analysis, we used two approaches: using part of speech tags as features, and using static text as features.

Metrics	**Using POS features**	**Using static text features**	**QuLog**
F1	1	0.94	0.98
Specificity	1	0.94	0.99

Analysis:

For Linguistic Sufficiency :

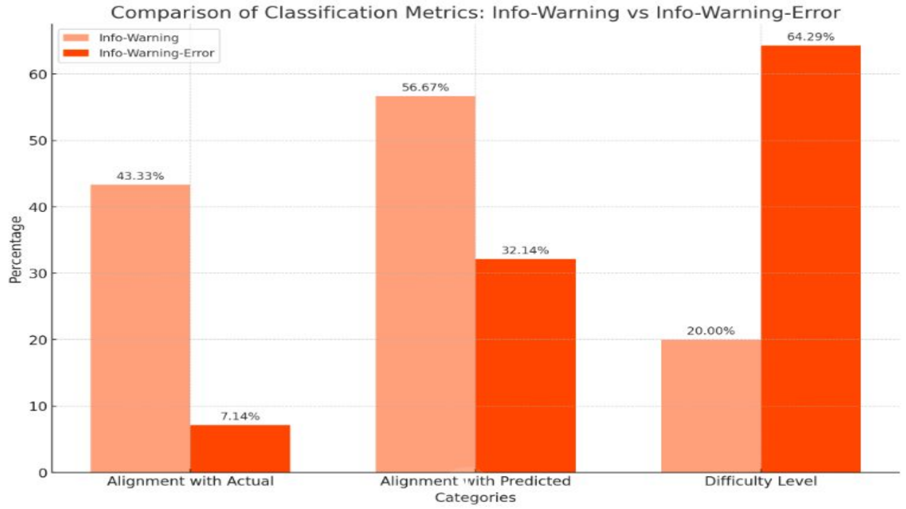
- We propose incorporating character length as an additional feature for extraction, considering its direct correlation with sufficiency.
- **Sufficient - Example: "Failed to retrieve data from API"**
- **Insufficient - Example: “Failed”**
- Dynamic variables are deemed beneficial for analysis, particularly due to the natural language association with variable names.
  Example: “Waiting for”
- Dataset mislabeling was encountered during sampling, where independently created labels did not align with combined responses. This resulted in disagreements or challenges in classification.
- The entire dataset was converted to lowercase, posing difficulties for BERT in accurately labeling the data.
  Example: stopping defaultleaderretrievalservice



Analysis:

For Log Levels :

- Similar data sampling and analysis - we agreed more with the results of the predicted model than the actual label. So, data mislabeling could be a problem here.
- Sometimes multiple log levels could be applicable.
- Humans also had trouble distinguishing between info, warning and error. Maybe including more information such as the lines around it might help.



Future Works:

- Analyse the Linguistic sufficiency with Camel-cased data.
- Include character length as a feature.
- Creating new datasets and validating them to avoid mislabeled data.
- Including dynamic text in LogBERT training data to determine if it increases linguistic accuracy.
- Extracting context by using lines around the logs could be a way to improve accuracy of LogBERT.

THANKS!

Do you have any questions?