# Demystifying and Extracting Fault-indicating Information from Logs for Failure Diagnosis

Junjie Huang[1], Zhihan Jiang[1], Jinyang Liu[1], Yintong Huo[1], Jiazhen Gu[1],

Zhuangbin Chen[2], Cong Feng[3], Hui Dong[3], Zengyin Yang[3], Michael R. Lyu[1]

[1]The Chinese University of Hong Kong

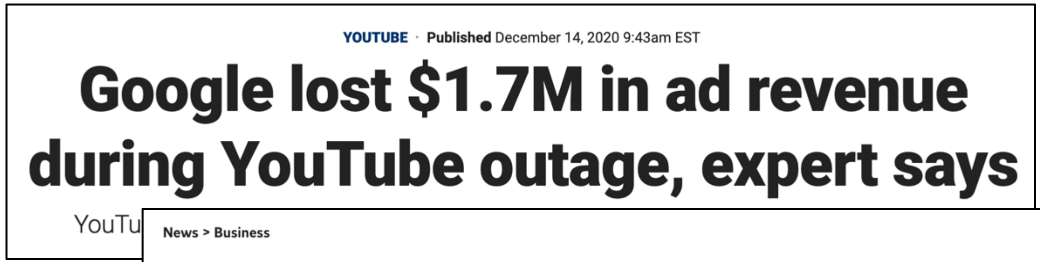
[2]Sun Yat-sen University [3]Huawei Cloud

*Read the paper!*

## Background

- Online service systems suffer from unplanned interruptions and failures

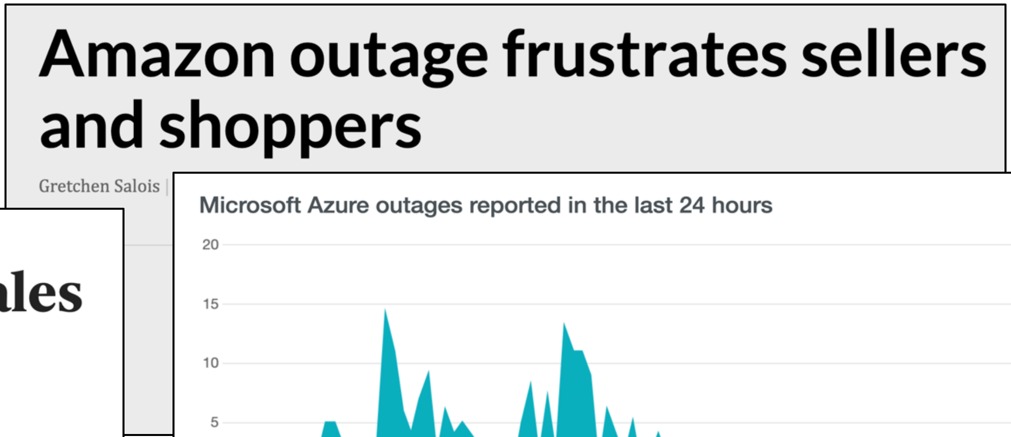
- IT companies must take timely and effective measures to respond to failures



YOUTUBE · Published December 14, 2020 9:43am EST

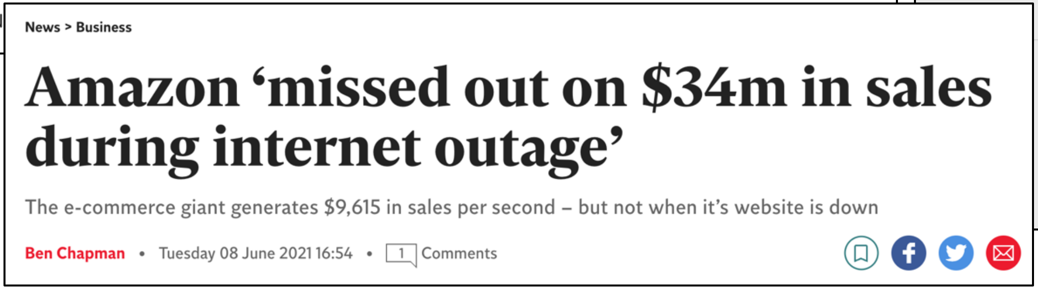
**Google lost $1.7M in ad revenue during YouTube outage, expert says**

YouTu



**Amazon outage frustrates sellers and shoppers**

Gretchen Salois |

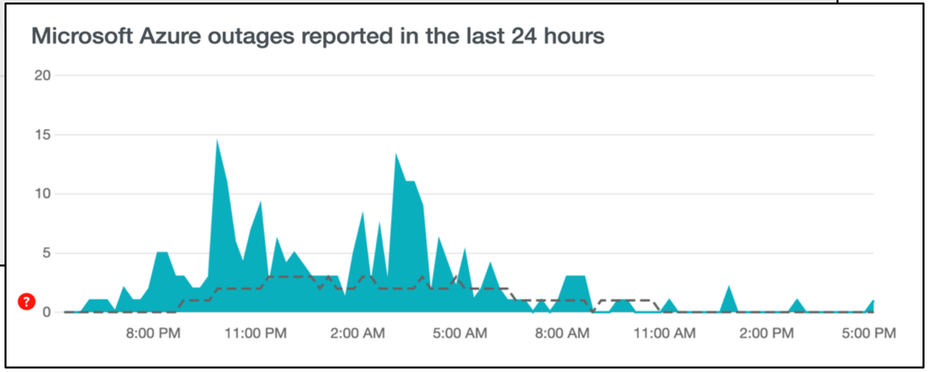


News > Business

**Amazon 'missed out on $34m in sales during internet outage'**

The e-commerce giant generates $9,615 in sales per second – but not when it's website is down

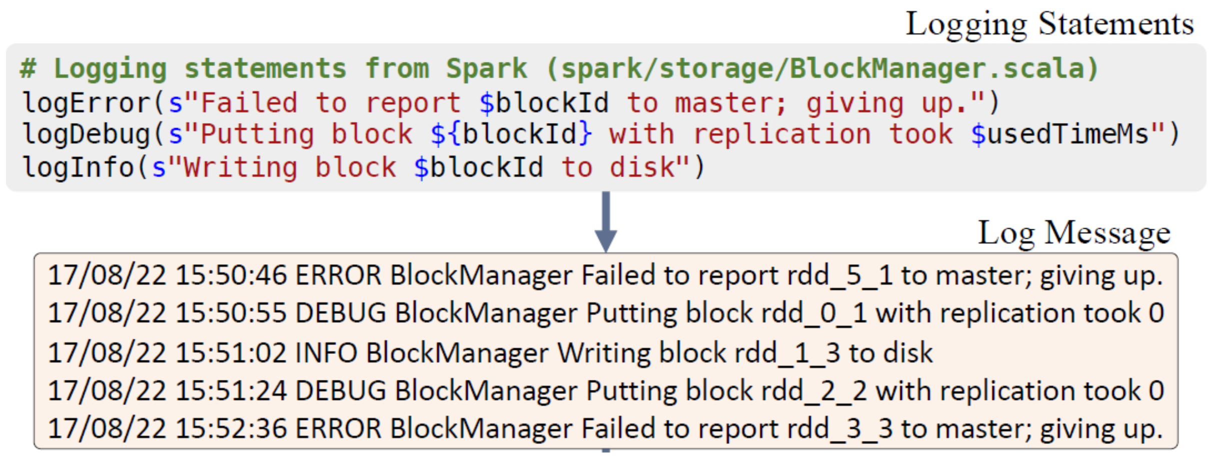
Ben Chapman • Tuesday 08 June 2021 16:54 • 1 Comments



## Background

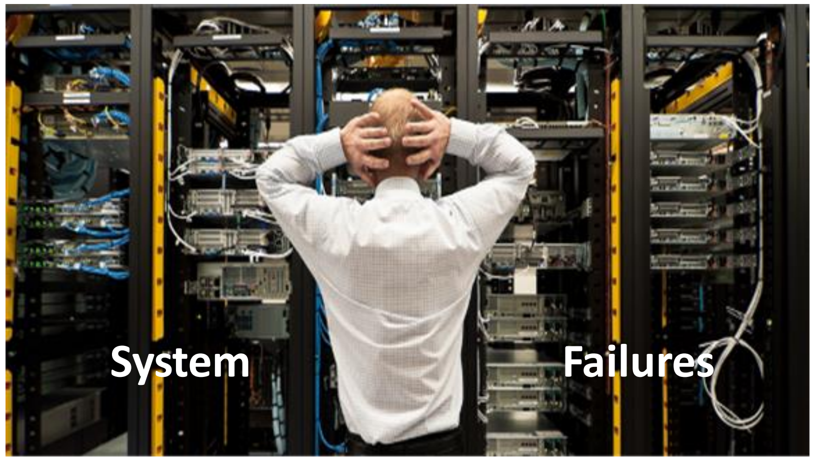
- Log is an important source for failure diagnosis
  - Logs record a vast range of software runtime events
  - At least 31% failure diagnosis practices rely on logs in a commercial bank service system [1]

- Examining logs and gaining insights about faults is time-consuming and labor-intensive
  - Software systems produce a large volume of logs due to the scale and complexity



```
# Logging statements from Spark (spark/storage/BlockManager.scala)
logError(s"Failed to report $blockId to master; giving up.")
logDebug(s"Putting block ${blockId} with replication took $usedTimeMs")
logInfo(s"Writing block $blockId to disk")
```

```
17/08/22 15:50:46 ERROR BlockManager Failed to report rdd_5_1 to master; giving up.
17/08/22 15:50:55 DEBUG BlockManager Putting block rdd_0_1 with replication took 0
17/08/22 15:51:02 INFO BlockManager Writing block rdd_1_3 to disk
17/08/22 15:51:24 DEBUG BlockManager Putting block rdd_2_2 with replication took 0
17/08/22 15:52:36 ERROR BlockManager Failed to report rdd_3_3 to master; giving up.
```

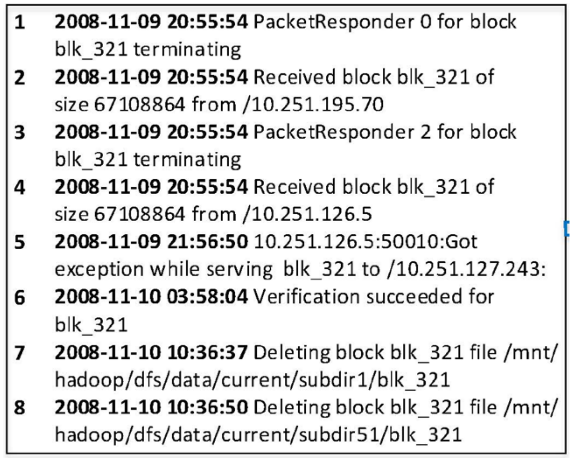


[1] Zhao et al. An empirical investigation of practical log anomaly detection for online service systems," in ESEC/FSE 2021

# Background

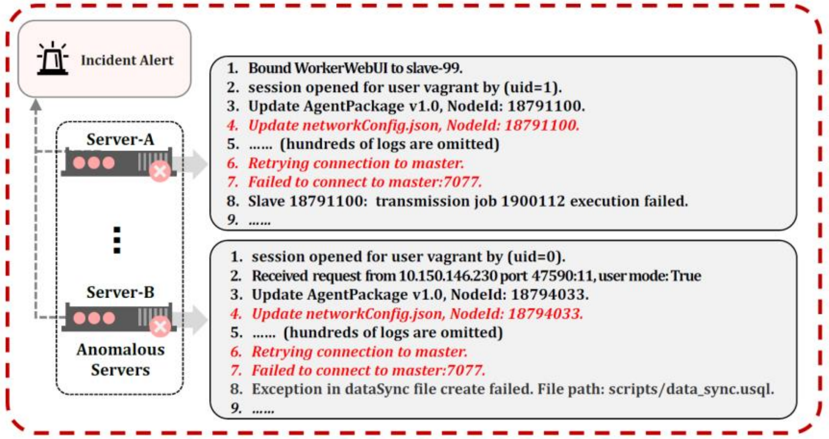
- Current log analysis approaches cannot find direct information for fault diagnosis

| | Log Analysis Approach | Identified information | Problem |
|---|---|---|---|
| ❶ | Log anomaly detection | Anomalous sessions | The sessions can be large (e.g., >100 logs) |
| ❷ | Log clustering | Incident-related logs | The logs can be long |
| ❸ | Log parsing | Templates and variables | Events and variables may not relate to fault components (e.g., devices and VMs) |



```
1 2008-11-09 20:55:54 PacketResponder 0 for block
  blk_321 terminating
2 2008-11-09 20:55:54 Received block blk_321 of
  size 67108864 from /10.251.195.70
3 2008-11-09 20:55:54 PacketResponder 2 for block
  blk_321 terminating
4 2008-11-09 20:55:54 Received block blk_321 of
  size 67108864 from /10.251.126.5
5 2008-11-09 21:56:50 10.251.126.5:50010:Got
  exception while serving blk_321 to /10.251.127.243:
6 2008-11-10 03:58:04 Verification succeeded for
  blk_321
7 2008-11-10 10:36:37 Deleting block blk_321 file /mnt/
  hadoop/dfs/data/current/subdir1/blk_321
8 2008-11-10 10:36:50 Deleting block blk_321 file /mnt/
  hadoop/dfs/data/current/subdir51/blk_321
```

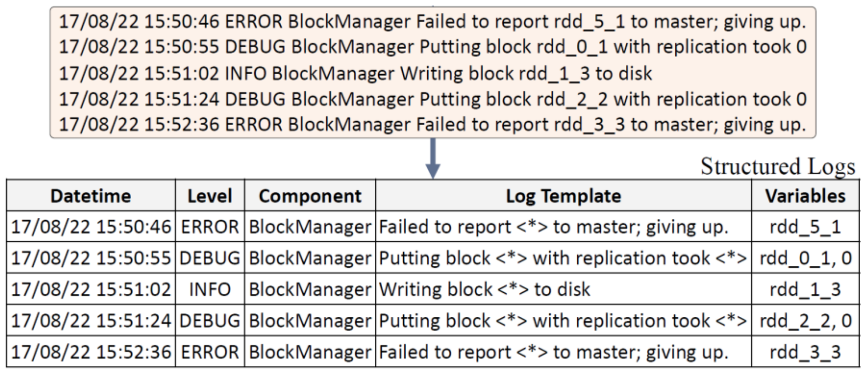
❶ Anomalous log sessions



```
1. Bound WorkerWebUI to slave-99.
2. session opened for user vagrant by (uid=1).
3. Update AgentPackage v1.0, NodeId: 18791100.
4. Update networkConfig.json, NodeId: 18791100.
5. ...... (hundreds of logs are omitted)
6. Retrying connection to master.
7. Failed to connect to master:7077.
8. Slave 18791100:  transmission job 1900112 execution failed.
9. ......
```

```
1. session opened for user vagrant by (uid=0).
2. Received request from 10.150.146.230 port 47590:11, user mode: True
3. Update AgentPackage v1.0, NodeId: 18794033.
4. Update networkConfig.json, NodeId: 18794033.
5. ...... (hundreds of logs are omitted)
6. Retrying connection to master.
7. Failed to connect to master:7077.
8. Exception in dataSync file create failed. File path: scripts/data_sync.usql.
9. ......
```

❷ Incident-related logs (in red)



```
17/08/22 15:50:46 ERROR BlockManager Failed to report rdd_5_1 to master; giving up.
17/08/22 15:50:55 DEBUG BlockManager Putting block rdd_0_1 with replication took 0
17/08/22 15:51:02 INFO BlockManager Writing block rdd_1_3 to disk
17/08/22 15:51:24 DEBUG BlockManager Putting block rdd_2_2 with replication took 0
17/08/22 15:52:36 ERROR BlockManager Failed to report rdd_3_3 to master; giving up.
```

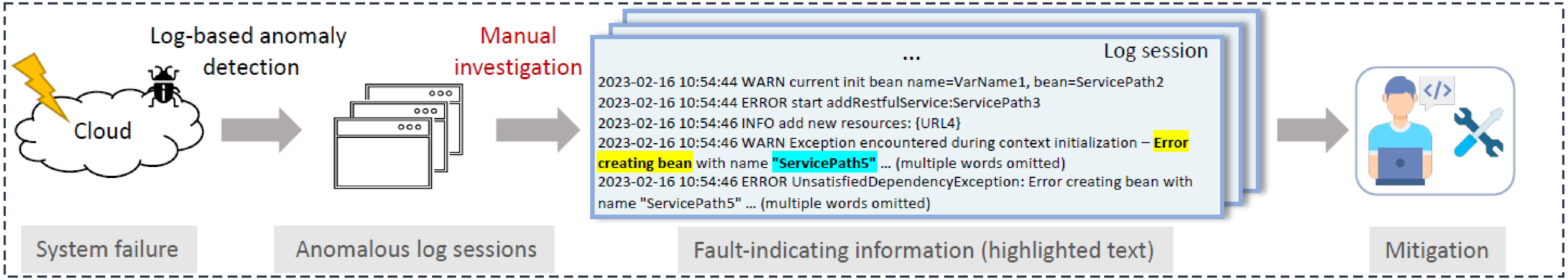
Structured Logs

| Datetime | Level | Component | Log Template | Variables |
|---|---|---|---|---|
| 17/08/22 15:50:46 | ERROR | BlockManager | Failed to report <*> to master; giving up. | rdd_5_1 |
| 17/08/22 15:50:55 | DEBUG | BlockManager | Putting block <*> with replication took <*> | rdd_0_1, 0 |
| 17/08/22 15:51:02 | INFO | BlockManager | Writing block <*> to disk | rdd_1_3 |
| 17/08/22 15:51:24 | DEBUG | BlockManager | Putting block <*> with replication took <*> | rdd_2_2, 0 |
| 17/08/22 15:52:36 | ERROR | BlockManager | Failed to report <*> to master; giving up. | rdd_3_3 |

❸ Parsed log templates and variables

## Our works

- ***Fault-indicating information***: the fine-grained log segments that
  - indicate fault-related information
  - provide actionable insights to guide fault diagnosis



```
2023-02-16 10:54:44 WARN current init bean name=VarName1, bean=ServicePath2
2023-02-16 10:54:44 ERROR start addRestfulService:ServicePath3
2023-02-16 10:54:46 INFO add new resources: {URL4}
2023-02-16 10:54:46 WARN Exception encountered during context initialization – Error
creating bean with name "ServicePath5" ... (multiple words omitted)
2023-02-16 10:54:46 ERROR UnsatisfiedDependencyException: Error creating bean with
name "ServicePath5" ... (multiple words omitted)
```

## A preliminary study on fault-indicating information

- Manually investigate how logs are leveraged by engineers through history reports
  - Fault-indicating description (FID) describes the symptoms of a fault
  - Fault-indicating parameter (FIP) denotes the precise position of the fault that needs action

CATEGORIES OF FAULT-INDICATING INFORMATION IN LOGS.

| Category | Subtype | Example | Number |
|---|---|---|---|
| Description | Error Message | . . . **url detection error**!agent taskId:f292c7e596d5435d9b9e9b9f47e1f872, retCode is empty | 34/96 |
| | Missing Component | . . . execute template error, reason is **Host name must not be empty** | 23/96 |
| | Abnormal Behavior | . . . reader request line for 192.168.132.245:8080(https) failed, **read line timed-out** | 26/96 |
| | Wrong Status | . . . **httpCode is 404. requestEntity’s type is GET**. requestEntity’s url is /users/orders/task. please check! | 13/96 |
| Parameter | Address | . . . **httpCode is 404. requestEntity’s type is GET**. requestEntity’s url is /users/orders/task. please check! | 18/71 |
| | Component ID | . . . **query** . . . **failed**. historyid=51890bae-57c6-47a3-b37d-62df9d2f3c87 | 28/71 |
| | Parameter Name | . . . **cannot get topicInfo for consumer by topic**: alarm_and_event_data | 25/71 |

[1] Descriptions is marked in **BLOD** and parameters is in UNDERLINE.

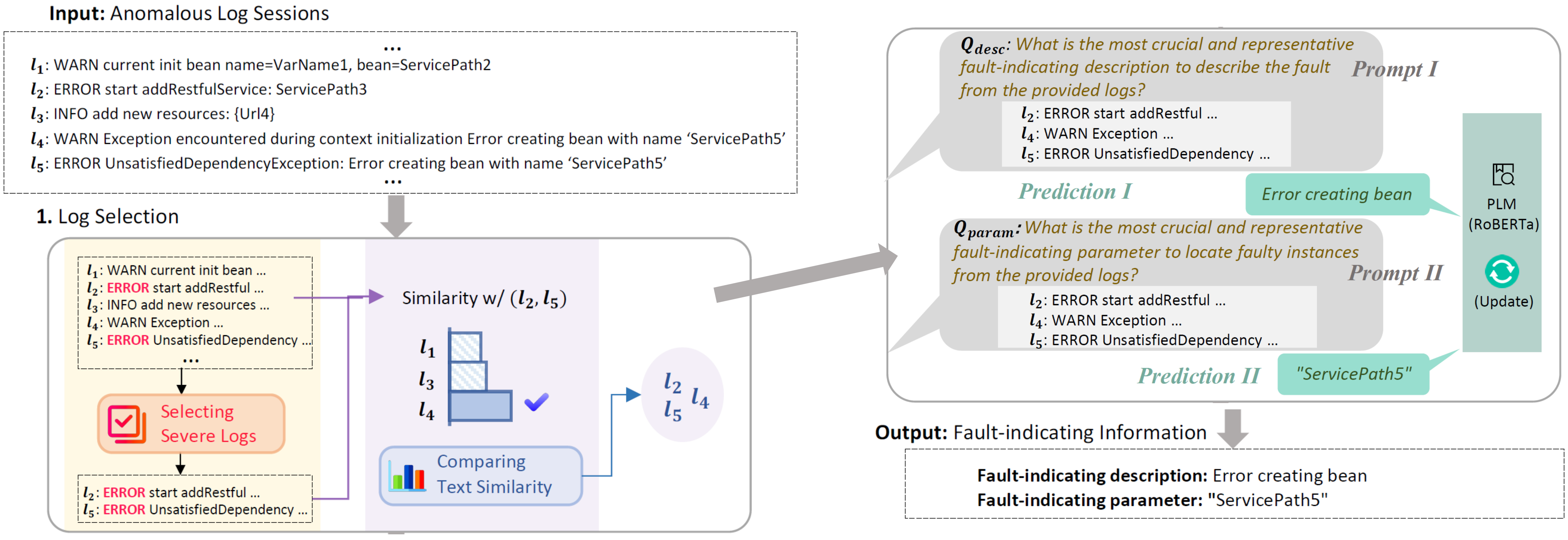
# A preliminary study on fault-indicating information

- Manually investigate how logs are leveraged by engineers through history reports
  - Fault-indicating description (FID) describes the symptoms of a fault
  - Fault-indicating parameter (FIP) denotes the precise position of the fault that needs action

- **Challenges:**
  - **Large volume of logs:** Only 1.7% of log lines contain either FID or FIP
  - **Noisy semantics in logs:** Only 14.1% of words are FID or FIP

# LoFI: A method to extract log fault-indicating information

- Two stages in LoFI: Log selection and prompt-based extraction



**Input:** Anomalous Log Sessions

...
$l_1$: WARN current init bean name=VarName1, bean=ServicePath2
$l_2$: ERROR start addRestfulService: ServicePath3
$l_3$: INFO add new resources: {Url4}
$l_4$: WARN Exception encountered during context initialization Error creating bean with name 'ServicePath5'
$l_5$: ERROR UnsatisfiedDependencyException: Error creating bean with name 'ServicePath5'
...

**1.** Log Selection

$l_1$: WARN current init bean ...
$l_2$: ERROR start addRestful ...
$l_3$: INFO add new resources ...
$l_4$: WARN Exception ...
$l_5$: ERROR UnsatisfiedDependency ...
...

$l_2$: ERROR start addRestful ...
$l_5$: ERROR UnsatisfiedDependency ...

$Q_{desc}$: *What is the most crucial and representative fault-indicating description to describe the fault from the provided logs?*

$l_2$: ERROR start addRestful ...
$l_4$: WARN Exception ...
$l_5$: ERROR UnsatisfiedDependency ...

$Q_{param}$: *What is the most crucial and representative fault-indicating parameter to locate faulty instances from the provided logs?*

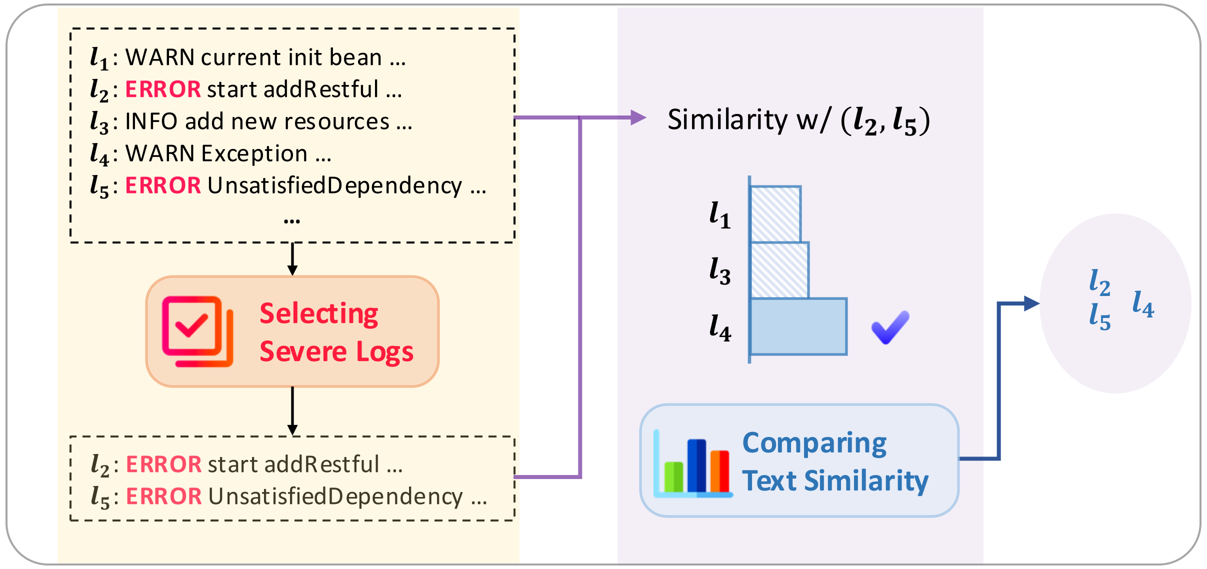
$l_2$: ERROR start addRestful ...
$l_4$: WARN Exception ...
$l_5$: ERROR UnsatisfiedDependency ...

**Output:** Fault-indicating Information

**Fault-indicating description:** Error creating bean
**Fault-indicating parameter:** "ServicePath5"

## Stage-1: log selection

- Problem: abnormal log sessions contain a large volume of logs
  - Irrelevant information can hinder extraction
  - Language models struggle to understand long context
- Method:
  - 1st step: Select logs with severe logging levels
    - i.e., FATAL > ERROR > WARN > INFO > DEBUG > TRACE > Others
  - 2nd step: Identify semantically similar logs
    - RoBERTa-base encoder yields log embeddings
    - Rank the textual similarity of embeddings

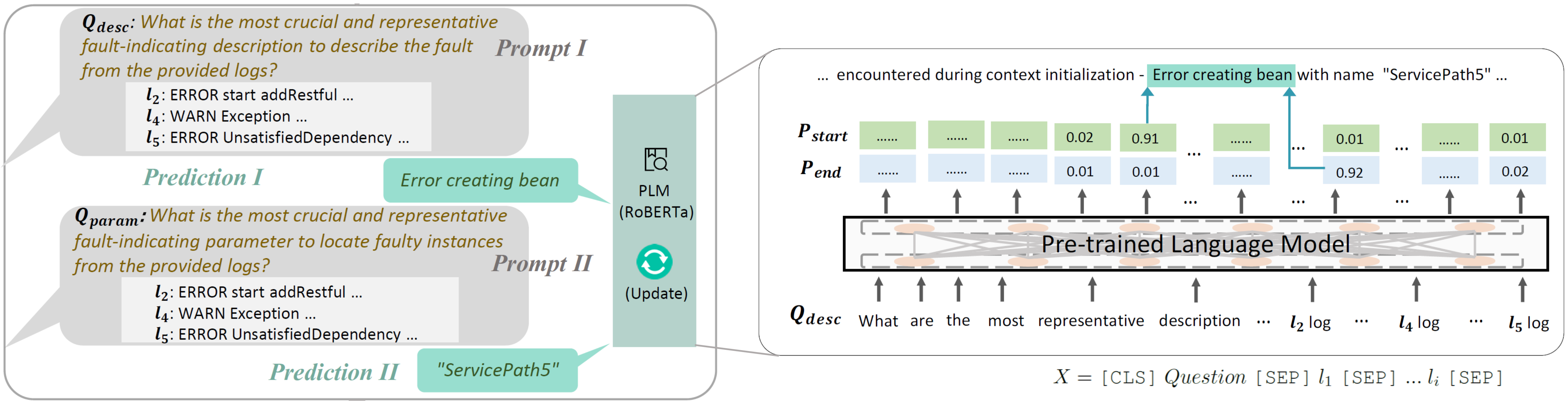


```
l1: WARN current init bean ...
l2: ERROR start addRestful ...
l3: INFO add new resources ...
l4: WARN Exception ...
l5: ERROR UnsatisfiedDependency ...
...
```

```
l2: ERROR start addRestful ...
l5: ERROR UnsatisfiedDependency ...
```

## Stage-2: prompt-based extraction

- Extraction using pretrained language models (PLM) with prompt-based tuning
  - We use two prompt questions for FID and FIP
  - PLM predicts start and end position of fault-indicating information
  - Fine-tune the PLM with logs and ground truth fault-indicating information
  - We use UniXCoder as the PLM

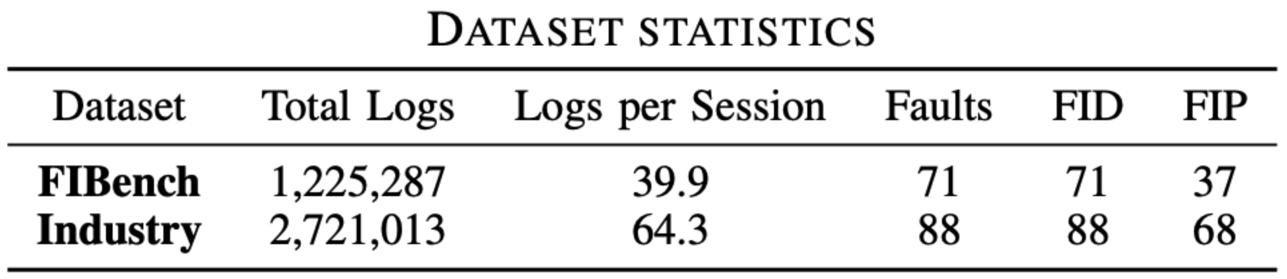

$Q_{desc}$: What is the most crucial and representative fault-indicating description to describe the fault from the provided logs?
$l_2$: ERROR start addRestful ...
$l_4$: WARN Exception ...
$l_5$: ERROR UnsatisfiedDependency ...
Prompt I
Prediction I
Error creating bean
$Q_{param}$: What is the most crucial and representative fault-indicating parameter to locate faulty instances from the provided logs?
$l_2$: ERROR start addRestful ...
$l_4$: WARN Exception ...
$l_5$: ERROR UnsatisfiedDependency ...
Prompt II
Prediction II
"ServicePath5"
PLM (RoBERTa)
(Update)
... encountered during context initialization - Error creating bean with name "ServicePath5" ...
$P_{start}$
$P_{end}$
0.02 0.91 0.01 0.01
0.01 0.01 0.92 0.02
Pre-trained Language Model
$Q_{desc}$ What are the most representative description ... $l_2$ log ... $l_4$ log ... $l_5$ log

$X = \texttt{[CLS]}\ Question\ \texttt{[SEP]}\ l_1\ \texttt{[SEP]} \ldots l_i\ \texttt{[SEP]}$

# Evaluation

- Research questions
  - RQ1: How effective is LoFI in the *offline* setting?
  - RQ2: How log selection affects the results of LoFI?
  - RQ3: How prompt-based tuning affects the results of LoFI?
  - RQ4: How LoFI helps SREs to diagnose in *online* setting?

- Datasets
  - **FIBench**: 71 examples collected from Apache Spark
  - **Industry**: 88 examples collected from CloudA

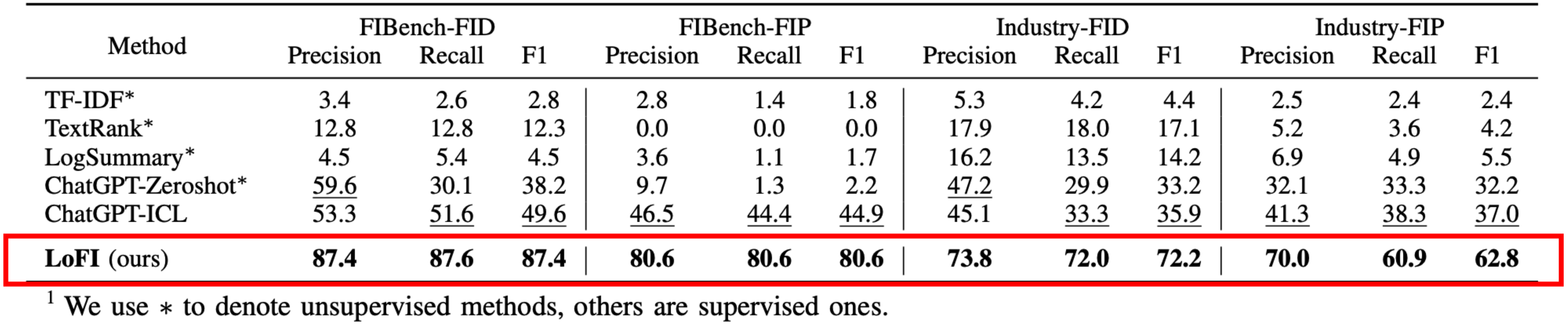


DATASET STATISTICS

| Dataset | Total Logs | Logs per Session | Faults | FID | FIP |
|---|---|---|---|---|---|
| **FIBench** | 1,225,287 | 39.9 | 71 | 71 | 37 |
| **Industry** | 2,721,013 | 64.3 | 88 | 88 | 68 |

## RQ1: Effectiveness of LoFI

- LoFI performs fault-indicating information extraction well!
  - LoFI achieves high accuracy in recognizing FID and FIP
  - LoFI outperforms all baselines by a large margin
- FID prediction is more accurate than FIP prediction

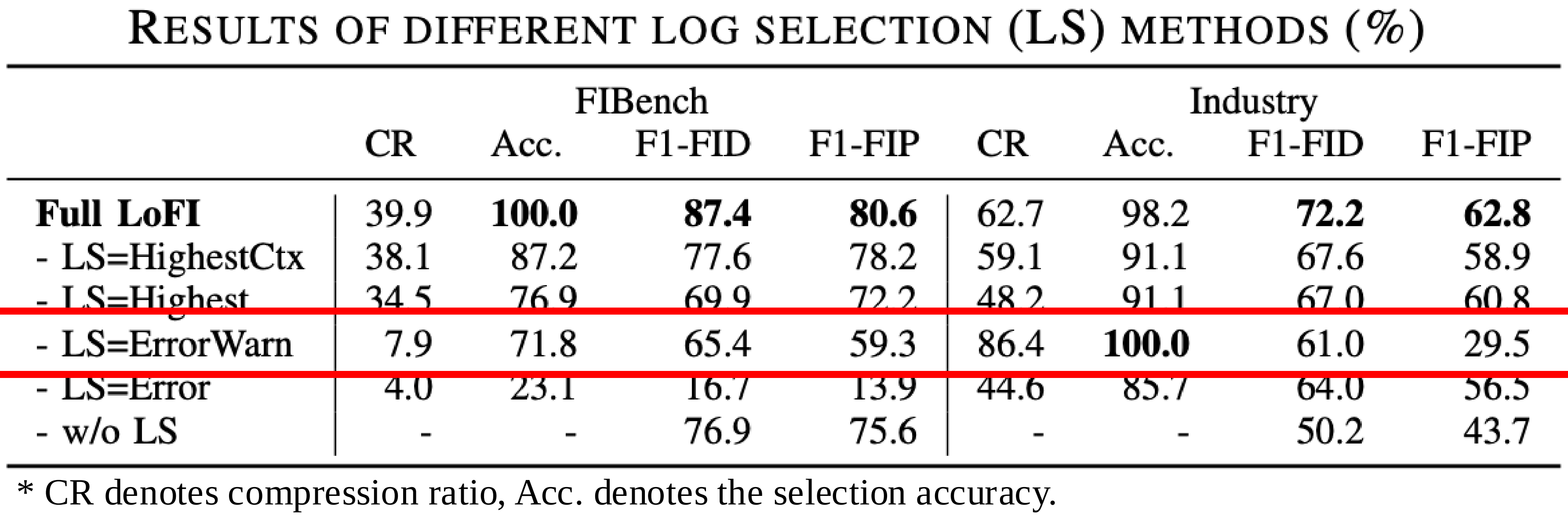


| Method | FIBench-FID | | | FIBench-FIP | | | Industry-FID | | | Industry-FIP | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Precision | Recall | F1 | Precision | Recall | F1 | Precision | Recall | F1 | Precision | Recall | F1 |
| TF-IDF* | 3.4 | 2.6 | 2.8 | 2.8 | 1.4 | 1.8 | 5.3 | 4.2 | 4.4 | 2.5 | 2.4 | 2.4 |
| TextRank* | 12.8 | 12.8 | 12.3 | 0.0 | 0.0 | 0.0 | 17.9 | 18.0 | 17.1 | 5.2 | 3.6 | 4.2 |
| LogSummary* | 4.5 | 5.4 | 4.5 | 3.6 | 1.1 | 1.7 | 16.2 | 13.5 | 14.2 | 6.9 | 4.9 | 5.5 |
| ChatGPT-Zeroshot* | 59.6 | 30.1 | 38.2 | 9.7 | 1.3 | 2.2 | 47.2 | 29.9 | 33.2 | 32.1 | 33.3 | 32.2 |
| ChatGPT-ICL | 53.3 | 51.6 | 49.6 | 46.5 | 44.4 | 44.9 | 45.1 | 33.3 | 35.9 | 41.3 | 38.3 | 37.0 |
| **LoFI** (ours) | **87.4** | **87.6** | **87.4** | **80.6** | **80.6** | **80.6** | **73.8** | **72.0** | **72.2** | **70.0** | **60.9** | **62.8** |

[1] We use $*$ to denote unsupervised methods, others are supervised ones.

# RQ2: How log selection (LS) affects LoFI?

- The two-stage LS in full LoFI is better than other LS methods
- Diagnosing only with logs in error and warning levels is not enough
  - Compression ratio (CR) is high, indicating noisy logs are kept



RESULTS OF DIFFERENT LOG SELECTION (LS) METHODS (%)

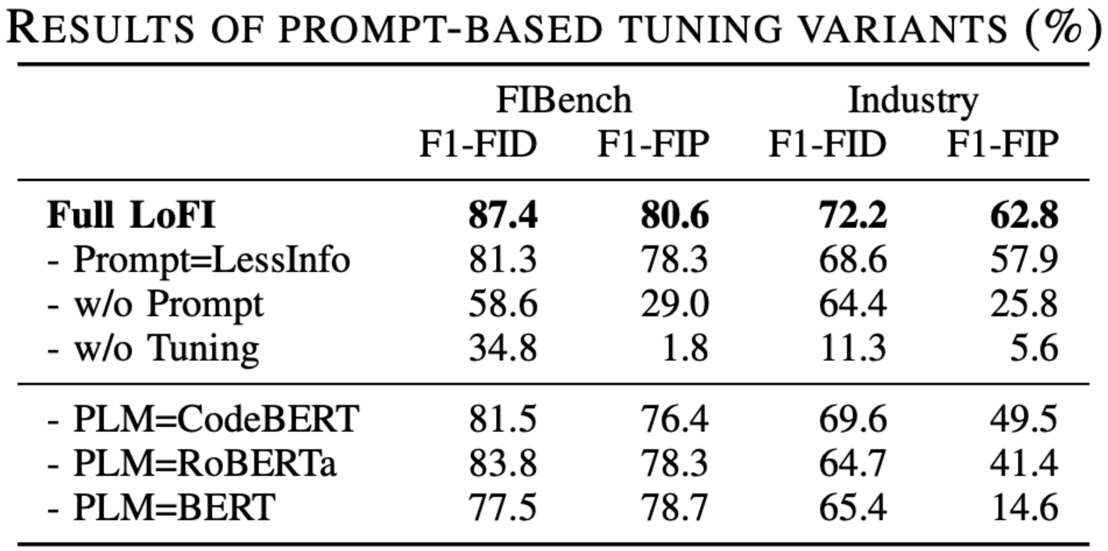
| | FIBench | | | | Industry | | | |
|---|---|---|---|---|---|---|---|---|
| | CR | Acc. | F1-FID | F1-FIP | CR | Acc. | F1-FID | F1-FIP |
| **Full LoFI** | 39.9 | **100.0** | **87.4** | **80.6** | 62.7 | 98.2 | **72.2** | **62.8** |
| - LS=HighestCtx | 38.1 | 87.2 | 77.6 | 78.2 | 59.1 | 91.1 | 67.6 | 58.9 |
| - LS=Highest | 34.5 | 76.9 | 69.9 | 72.2 | 48.2 | 91.1 | 67.0 | 60.8 |
| - LS=ErrorWarn | 7.9 | 71.8 | 65.4 | 59.3 | 86.4 | **100.0** | 61.0 | 29.5 |
| - LS=Error | 4.0 | 23.1 | 16.7 | 13.9 | 44.6 | 85.7 | 64.0 | 56.5 |
| - w/o LS | - | - | 76.9 | 75.6 | - | - | 50.2 | 43.7 |

* CR denotes compression ratio, Acc. denotes the selection accuracy.

- LS=Highest (highest level logs)
- LS=HighestCtx (highest level and context logs)
- LS=ErrorWarn (error and warning level logs)
- LS=Error (error level logs)

## RQ3: How prompt-based tuning affects the results?

- Prompt variants:
  - Using prompt question is helpful for extracting fault-indicating information
  - Prompt question with explicit instructions can improve performance
  - Fine-tuning can improve the performance even with a small amount of training data
- Pretraining Language Models:
  - Using UniXCoder performs better than using other models



RESULTS OF PROMPT-BASED TUNING VARIANTS (%)

| | FIBench | | Industry | |
|---|---|---|---|---|
| | F1-FID | F1-FIP | F1-FID | F1-FIP |
| **Full LoFI** | **87.4** | **80.6** | **72.2** | **62.8** |
| - Prompt=LessInfo | 81.3 | 78.3 | 68.6 | 57.9 |
| - w/o Prompt | 58.6 | 29.0 | 64.4 | 25.8 |
| - w/o Tuning | 34.8 | 1.8 | 11.3 | 5.6 |
| - PLM=CodeBERT | 81.5 | 76.4 | 69.6 | 49.5 |
| - PLM=RoBERTa | 83.8 | 78.3 | 64.7 | 41.4 |
| - PLM=BERT | 77.5 | 78.7 | 65.4 | 14.6 |

- Prompt=LessInfo (prompt with less information)
  + "Fault-indicating descriptions in the following logs:" + logs
  + "Fault-indicating parameters in the following logs:" + logs

- w/o Prompt (without prompt question)
- w/o Tuning (without fine-tuning on the training set)

## RQ4: How LoFI assists in online diagnosis?

- User study after one-month deployment:
  - 10 engineers rate the 50 fault examples during the deployment period
- Q1: Does FID accurately represent anomalous events?
  - Average rating: 4.34 (5-point Likert scale)
- Q2: Does FIP accurately identify anomalous components?
  - Average rating: 4.02 (5-point Likert scale)
- Q3: Would extracting fault-indicating information aid in fault diagnosis?
  - All participants agree!

## Conclusions

- Fault-indicating information are useful
  - Fault-indicating descriptions describe the symptoms of a fault
  - Fault-indicating parameters denote the precise position of the fault that needs action

- LoFI can successfully extracts fault-indicating information
  - Log selection and prompt-based tuning are effective in improving the performance

Check this work in:

Paper Github

Check our repo:

Awesome-LLM-AIOps

**Q & A**