

SemParser: A Semantic Parser for Log Analytics

Yintong Huo, Yuxin Su, Cheryl Lee, Michael Lyu
ICSE 2023

発表者: 稲葉 真幸

ログデータの解析

- ログの重要性

ログにはシステムの健全性，パフォーマンス，安全性に関する貴重な情報が含まれる。
特に大規模な分散システムやクラウドベースのアプリケーションではログデータの量が膨大。
その解析はより複雑かつ重要。

- ログのパーズ

ログファイル内のデータを解析し，構造化するプロセスのこと。
テキスト形式のログから情報を抽出して意味のある形式に変換
→ログデータからの重要な情報の識別，エラーの追跡，システムのパフォーマンス監視
などが可能に。

既存研究：ログの自動解析

ログの自動解析には多くの課題があるが、
統計的手法や履歴ベースの手法を活用して進歩を遂げてきた

- SLCT[1], LFA[2]

過去に頻繁に出現した単語の数をカウントすることでログテンプレートを構築

- Logram[3]

頻繁に出現するn-gramパターンを考慮

- LogSig[4], SHISO[5]

それぞれ単語ペアと単語の長さでログを符号化，クラスタリングアルゴリズムによって分割

[1] R. Vaarandi, "A data clustering algorithm for mining patterns from event logs," in Proceedings of the 3rd IEEE Workshop on IP Operations & Management (IEEE Cat. No. 03EX764), Kansas City, MO, USA, Oct 3, 2003. IEEE, 2003, pp. 119–126.

[Online]. Available: <https://ieeexplore.ieee.org/document/1251233>

[2] M. Nagappan and M. A. Vouk, "Abstracting log lines to log event types for mining software system logs," in Proceedings of the 7th International Working Conference on Mining Software Repositories, Cape Town, South Africa, May 2–3, 2010. IEEE Computer Society, 2010, pp. 114–117. [Online]. Available: <https://doi.org/10.1109/MSR.2010.5463281>

[3] H. Dai, H. Li, C. S. Chen, W. Shang, and T.-H. Chen, "Logram: Efficient log parsing using n-gram dictionaries," CoRR, vol. abs/2001.03038, 2020. [Online]. Available: <http://arxiv.org/abs/2001.03038>

[4] L. Tang, T. Li, and C.-S. Perng, "Logsig: Generating system events from raw textual logs," in Proceedings of the 20th Conference on Information and Knowledge Management, UK, October 24–28, 2011. ACM, 2011, pp. 785–794. [Online]. Available: <https://doi.org/10.1145/2063576.2063690>

[5] M. Mizutani, "Incremental mining of system log format," in International Conference on Services Computing, Santa Clara, CA, USA, June 28–July 3, 2013. IEEE Computer Society, 2013, pp. 595–602. [Online]. Available: <https://doi.org/10.1109/SCC.2013.73>

Syntax-base（構文ベース）のパーサー

現行のログパーサーはすべてSyntax-baseである。

表面的な特徴（例えば単語の長さ、ログの長さ、頻度など）から構文解析するため、高レベルで意味を認識することは難しい。具体的に以下3つのような制限がある。

- 個々の情報トークンに十分な注意が払われていない

例) パラメータや技術的単語は前置詞よりも重要

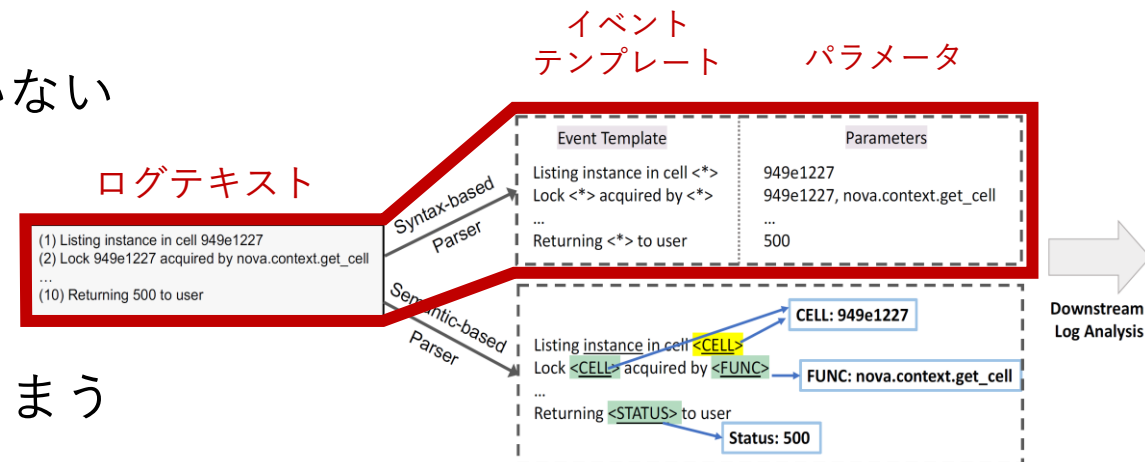
- メッセージ内のセマンティクスの欠如

パラメータを意味のない文字列として扱ってしまう

例) 949e1227が意味のない文字列としてみなされる

- ログ間の関係の見落とし

例) 949e1227がcellを指していることがわからない



目的, Research Question

- 研究の目的

前述の制限に取り組むため, Semantic-base (意味ベース) に基づくログ解析ツール **SemParser** を提案する.

- Research Question

RQ1: SemParserは**意味抽出**にどの程度有効か?

RQ2: SemParserは**異常検出**にどの程度有効か?

RQ3: SemParserは**障害検出**にどの程度有効か?

用語・概念

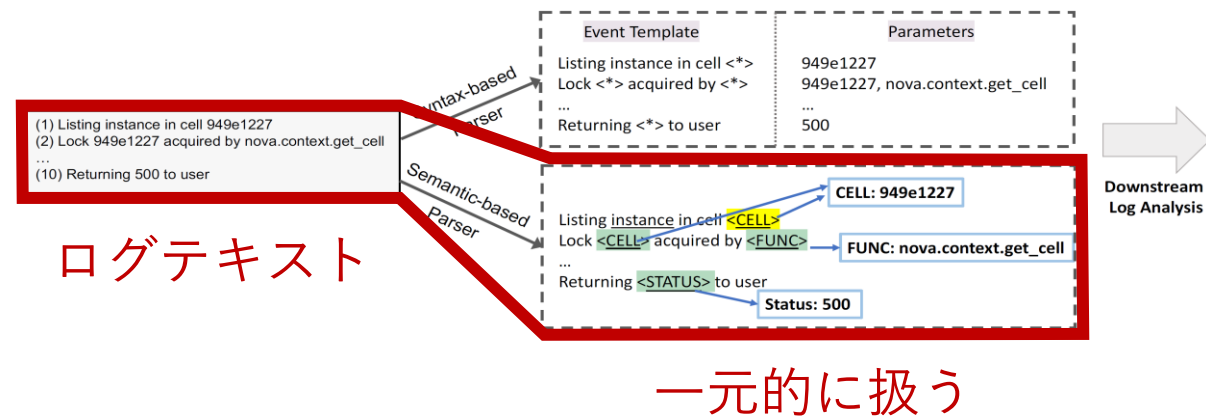
この論文ではログの意味解析に焦点を当てる.

- ここでいう意味解析とは

ログに現れるテキストは以下の2つに分けられる.

- メッセージレベルの意味 (Concept) : 用語や概念. 例) cell
- インスタンスレベルの意味 (Instance) : 変数. 例) 949e1227

2つのペアを「CI pairs」ということにする.

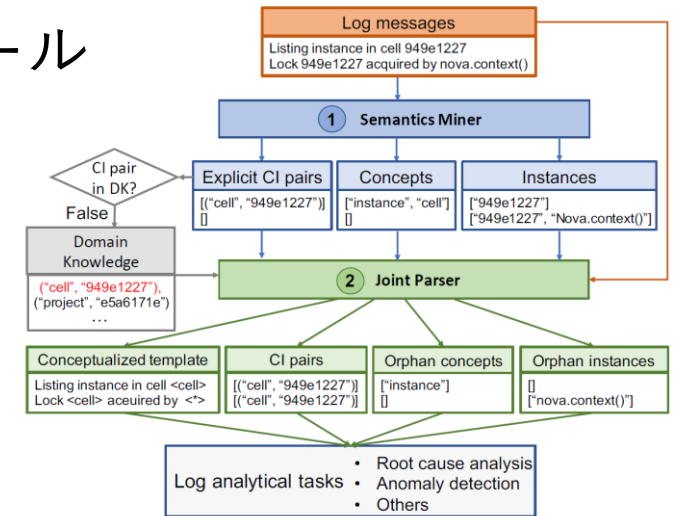


Semantic-base（意味ベース）のパーサー

前述の制限に取り組むため、意味論に基づくログ解析ツール
SemParserを提案する。

• SemParserの概要

2つの主要部分，Semantics MinerとJoint Parserを含む。



① Semantics Miner

ログメッセージから「意味的」情報を抽出する部分。CIペアの探索と各トークンの型の分類を行う。

② Joint Parser

①から得られた情報を基により高度な意味解析を行う。単に抽出された情報を組み合わせるだけでなく、それらがどのように関連しているかの理解を試みる

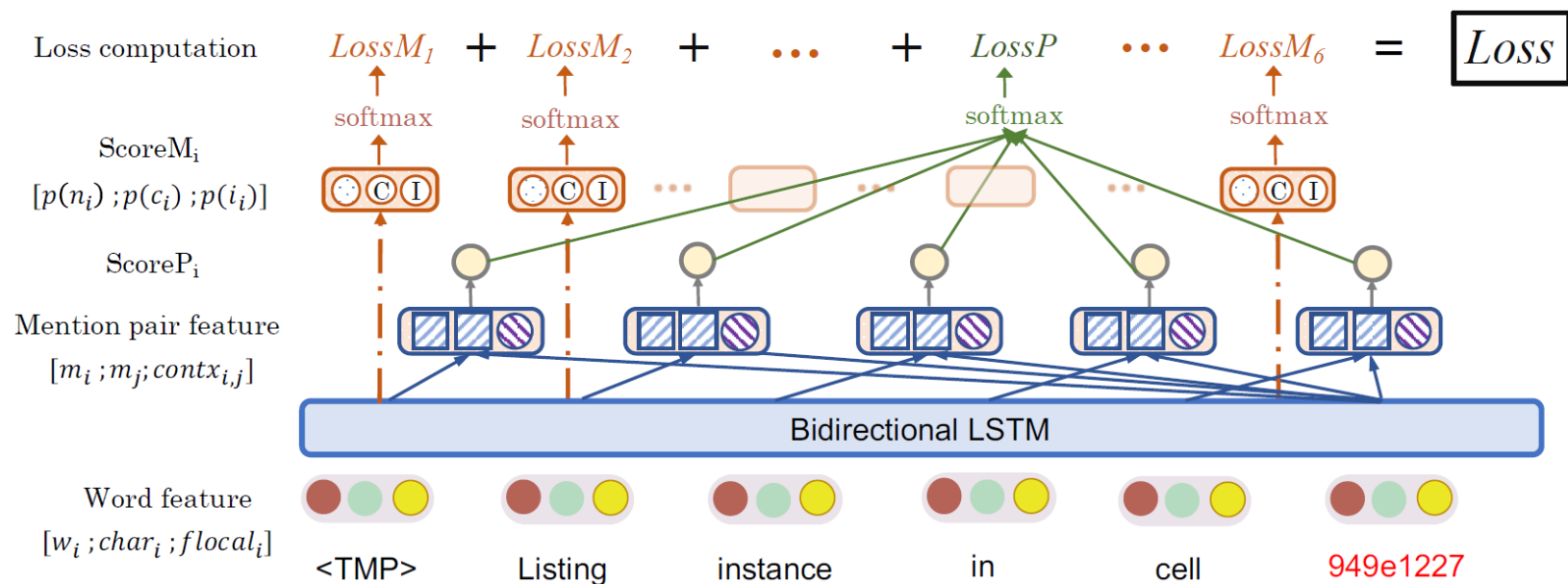
Semantic Minerの構造

文脈エンコーダ：ログメッセージの単語からコンテキストに基づいた表現を取得。Bi-LSTMを利用

ペアマッチャー：ペア候補の信頼スコアをランク付けする。

ワードスコアラ：各トークンが概念、インスタンス、またはそのどちらでもないかを判断。
これによりメッセージレベルの意味情報を把握。

損失関数：損失を合算し、モデルが両方のサブタスクから同時に学習するよう設計。



データセット

- SemParserは、OpenStackから収集されたログメッセージを含む公開データセットで実装
- 200件のログをランダムにサンプリング
- 単語が概念，インスタンス，またはそのどちらでもないか，および明示的なCIペアを独立してラベリング

TABLE I: Statistics of dataset for semantic mining.

System type	System	#Logs	#Pairs	#Temp.	Unseen
Mobile system	Android	2,000	6,478	166	82.8%
Operating system	Linux	2,000	2,905	118	86.8%
Distributed system	Hadoop	2,000	2,592	14	84.6%
	HDFS	2,000	3,105	30	47.0%
	OpenStack	2,000	4,367	43	52.3%
	Zookeeper	2,000	1,189	50	75.9%

TABLE II: Statistics of anomaly detection datasets.

Dataset	#Message	Anomaly rate
HDFS dataset	11,175,629	3%
F-Dataset	1,318,860	0.22%

RQ1：SemParserは意味抽出にどの程度有効か？

- CI pairsの抽出精度に焦点をあてて実験。
すべての結果の**F1値の平均が0.985**と高い
- SemParserはログからインスタンスレベルの意味情報を抽出できている，といえる

Framework	System																			
	Andriod					Hadoop					HDFS					Linux				
	P	—	R	—	F1	P	—	R	—	F1	P	—	R	—	F1	P	—	R	—	F1
SemParser	0.951	0.935	0.943			0.993	0.978	0.985			1.000	1.000	1.000			0.998	0.977	0.987		
- w/o F_{char}	0.981	0.909	0.943			0.988	0.953	0.970			1.000	0.998	0.999			0.995	0.957	0.976		
- w/o F_{local}	0.979	0.858	0.915			0.993	0.880	0.933			1.000	0.999	0.999			0.992	0.947	0.969		
- w/o $LSTM$	0.979	0.858	0.915			0.993	0.879	0.932			1.000	0.999	0.999			0.995	0.909	0.951		
- w/o F_{contx}	0.977	0.060	0.113			0.984	0.253	0.403			0.999	0.289	0.449			0.999	0.242	0.389		

RQ2：SemParserは異常検出にどの程度有効か？

- ログメッセージの短期間内に異常が存在するかを予測。

SemParserと他の4つのパーサーを比較，それぞれ4つのモデルで評価。

(パーサー：構造化，モデル：実行)

- 平均F1スコアで以下の改善が見られた

HDFS Dataset：1.22%向上

F-Dataset：10%以上の向上

- インスタンスレベルでの意味抽出を行っていることがこの結果の改善にも寄与

(a) HDFS Dataset.

Baseline	Technique											
	DeepLog			LogRobust			CNN			Transformer		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LenMa	.897	.994	.943	.914	.995	.953	.924	.995	.958	.872	.908	.890
AEL	.896	.994	.943	.935	.996	.964	.922	.995	.958	.893	.904	.898
Drain	.908	.994	.949	.934	.994	.963	.925	.995	.959	.886	.871	.878
IPLoM	.898	.994	.944	.940	.994	.966	.926	.996	.960	.889	.904	.896
SemParser	.940	.995	.967	.954	.995	.974	.931	.995	.962	.881	.954	.916
$\Delta\%$	+1.86%			+0.82%			+0.21%			+2.00%		

(b) F-Dataset

Baseline	Technique											
	DeepLog			LogRobust			CNN			Transformer		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LenMa	.717	.938	.813	.714	.924	.806	.793	.815	.804	.685	.896	.776
AEL	.738	.934	.824	.791	.877	.832	.747	.924	.826	.503	.962	.660
Drain	.824	.867	.845	.810	.886	.846	.737	.943	.827	.693	.919	.790
IPLoM	.863	.833	.848	.808	.877	.841	.834	.834	.834	.929	.683	.787
SemParser	.971	.927	.948	.952	.913	.932	.907	.899	.903	.938	.904	.921
$\Delta\%$	+11.80%			+10.17%			+8.27%			+16.58%		

RQ3：SemParserは障害検出にどの程度有効か？

- 故障識別タスクは，特定のAPIエラーを特定する。

RQ2と同様に比較

- 4つのベースラインパーサーを大幅に上回った
- Recall@1スコアでは，全てのモデルにおいてそれぞれ

12.5%，10%，7.75%，3.81%の改善が見られた。

LSTM Atten-biLSTM CNN Transformer

(a) A case for instance-level semantics.

API error	server add volume
Log message	... Cannot 'attach_volume' instance 853cfe1b ...
C-Template	... Cannot 'attach_volume' instance <*server*> ...
CI Pairs	[(server, 853cfe1b)]

(b) A case for message-level semantics.

API error	network create
Log message	... POST /v2.0/networks ...
C-Template	... POST /<*>/networks ...
Concepts	[POST, networks]

	Model											
Baseline	Rec@1	LSTM Rec@2	Rec@3	Atten-biLSTM			CNN			Transformer		
				Rec@1	Rec@2	Rec@3	Rec@1	Rec@2	Rec@3	Rec@1	Rec@2	Rec@3
LenMa	0.839	0.924	0.953	0.858	0.943	0.957	0.877	0.962	0.967	0.919	0.934	0.948
AEL	0.844	0.919	0.953	0.853	0.915	0.962	0.810	0.905	0.929	0.858	0.929	0.953
Drain	0.844	0.919	0.972	0.863	0.938	0.953	0.867	0.948	0.967	0.853	0.919	0.943
IPLoM	0.848	0.943	0.957	0.863	0.948	0.962	0.867	0.967	0.986	0.839	0.910	0.948
SemParser	0.954	0.968	0.968	0.954	0.968	0.972	0.945	0.963	0.972	0.954	0.958	0.968
$\Delta\%$	+12.50%	+2.65%	-0.41%	+10.54%	+2.11%	+1.04%	+7.75%	-0.42%	-1.44%	+3.81%	+2.46%	+2.11%

まとめ

構文ベースから意味ベースにすることで
ログデータの様々な解析精度が向上する

- 所感

意味抽出において、処理（今回の場合ペア作りと分類，関連づけ）を行うことで精度が向上する可能性があるというのが面白い。

自分の卒論テーマにも関連する