A Semantic-aware Representation Framework for Online Log Analysis

Weibin Meng, Ying Liu, Yuheng Huang, Shenglin Zhang
Federico Zaiter, Bingjin Chen, Dan Pei

清華大學 Tsinghua University
北京郵電大學 BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS
南開大學 Nankai University
中山大學 SUN YAT-SEN UNIVERSITY

Outline

Background

Internet Services

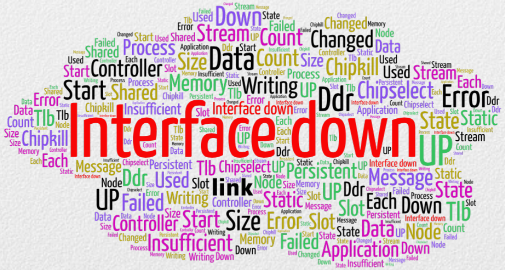
Growing rapidly

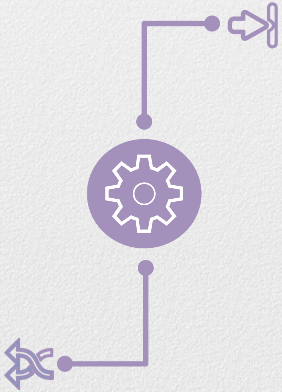
Various types of services

Stability are important

Logs

- Monitoring data:
  - logs, traffic, PV.
- Logs are one of the most valuable data for service management





General

- Every service generates logs

Diverse

- Logs record a vast range of runtime information (7*24)

Logs

- Logs are unstructured text
  - designed by developers
  - printed by logging statements (e.g., printf())

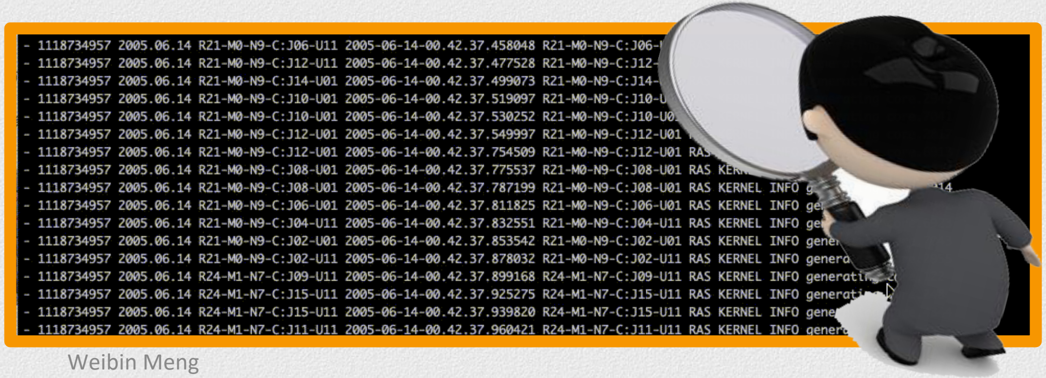
L_1. Interface ae3, changed state to down
L_2. Interface ae3, changed state to up
L_3. Interface ae1, changed status to down
L_4. Interface ae1, changed status to up
L_5. Vlan-interface vlan22, changed state to down
L_6. Vlan-interface vlan22, changed state to up

Logs are similar to nature language

Manual inspection of logs

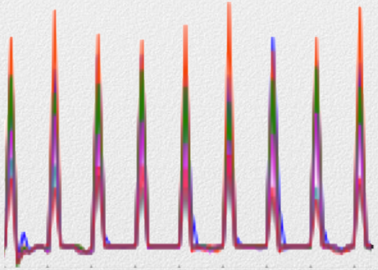
- Manual inspection of logs is impossible
  - A large-scale service is often implemented/maintained by hundreds of developers/operators.
  - The volume of logs is growing rapidly.
  - Traditional way: labor-intensive and time consuming

Automatic log analysis

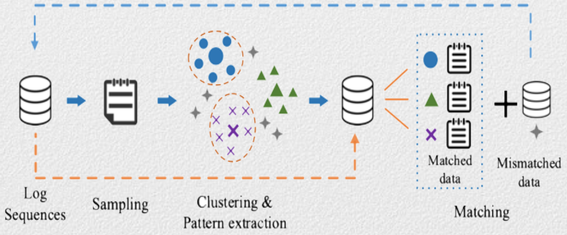


Automatic log analysis

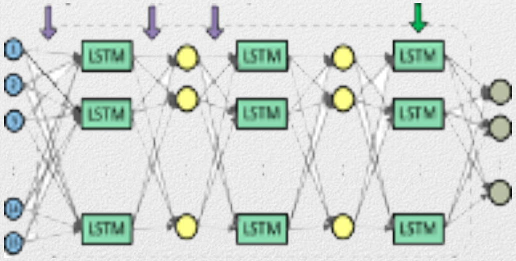
- Automatic log analysis approaches, which are employed for services management, have been widely studied



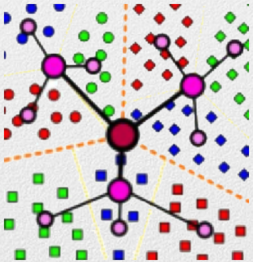
Monitoring
[INFOCOM'19]



Problem Identifying
[FSE'18]



Anomaly detection
[CCS'17]



Failure prediction
[SIGMETRICS'18]

Log representation

- Most of automatic log analysis require structured input
  - Logs are unstructured text
- Log representation serves as the first step of automatic log analysis
  - Template index
  - Template count vector

→ Lost semantic information

Semantic-aware log representation approach

Challenges

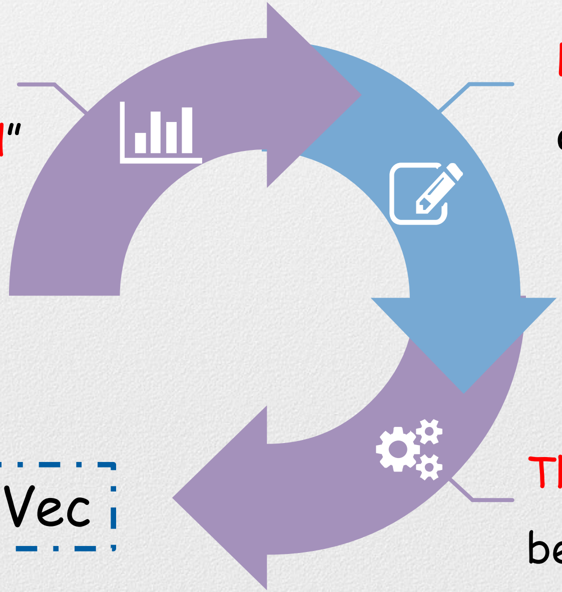
1. Domain-specific semantic information
   - Logs contain logs of domain-specific words

2. Out-of-vocabulary (OOV) words
   - The vocabulary is growing continuously because the service can be upgraded to add new features and fix bugs

Idea

Original goal of logs: "
logs are for users to read"

Logs are designed by developers
and "printf"-ed by services

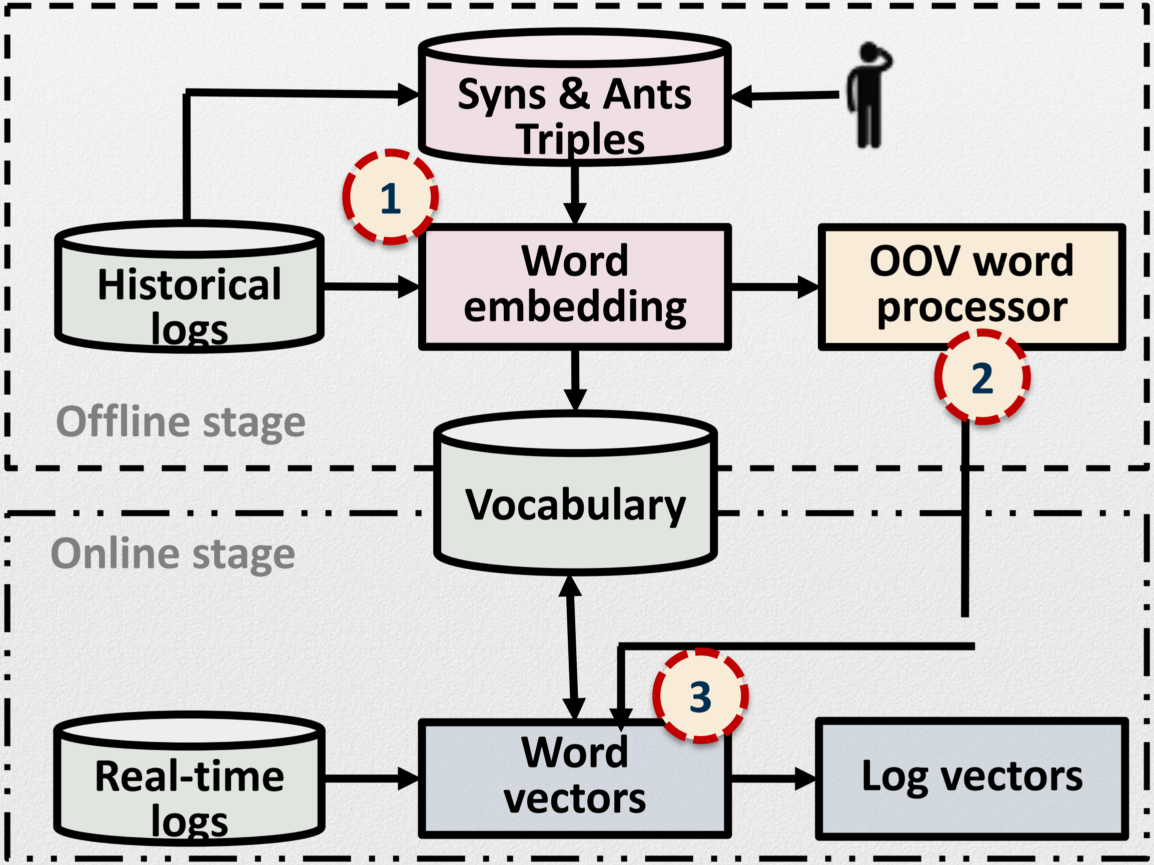


The intuition and methods in NLP can
be applied for log representation

Log2Vec

Design

Overview of Log2Vec



1. Log-specific word embedding

2. Out-of-vocabulary word processor

3. Log vector generation

Open source toolkit: https://github.com/WeibinMeng/Log2Vec

Log-specific semantics

- When embed words of logs, we should consider many information:
  - Antonyms
  - Synonyms
  - Relation triples
  - Others (future work)

Historical logs:

L_1. Interface ae3, changed state to down
L_2. Interface ae3, changed state to up
L_3. Interface ae1, changed status to down
L_4. Interface ae1, changed status to up

Real-time logs:

L_5. Vlan-interface vlan22, changed state to down
L_6. Vlan-interface vlan22, changed state to up

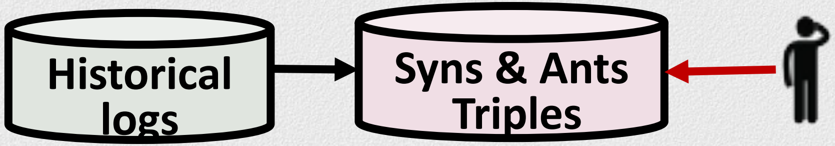
Out-of-vocabulary	Vlan-interface
Relation triples	(Interface, changed, state)
Antonym pairs	(down, up)
Synonym pairs	(state, status)

- Traditional word embedding methods (e.g., word2vec) assumes that words with a similar context tend to have a similar meaning

fail to capture the log-specific meaning

Prepare log-specific information

- Automatically extract
  - Antonyms & Synonyms
    - Search from WordNet[1], a lexical database for English
  - Triples
    - Dependency tree[2]
- Manually modify



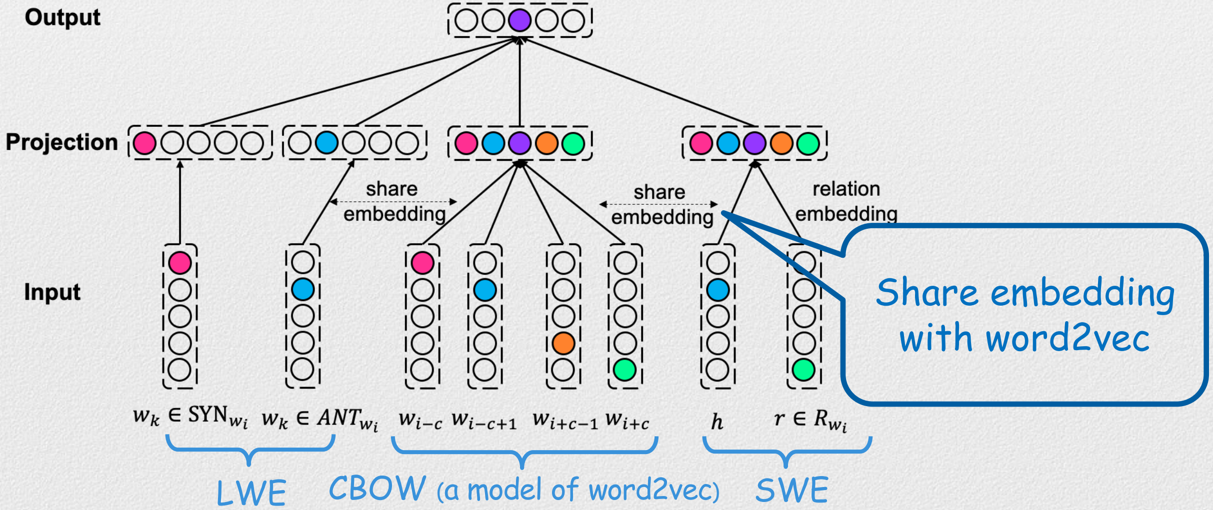
Relations	Word pairs		Adding methods
Synonyms	Interface	port	Operators
Antonyms	DOWN	UP	WordNet
	powerDown	powerUp	Operators
Relations	(interface, changed, state)		Dependency tree

[1]Fellbaum C. WordNet[J]. The encyclopedia of applied linguistics, 2012.

[2]Culotta A, Sorensen J. Dependency tree kernels for relation extraction[C]//Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04). 2004: 423-429.

Log-specific word embedding

- Log-specific word embedding combines two existing methods:
  - Lexical Information word embedding (LWE)[1] -> ants & syns
  - Semantic Word embedding (SWE)[2] -> relation triples



[1]Luchen Tan, Haotian Zhang, Charles Clarke, and Mark Smucker. Lexical comparison between wikipedia and twitter corpora by using word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing* , pages 657–661, 2015.

[2]/Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. Learning semantic word embeddings based on ordinal knowledge constraints. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1501– 1511, 2015.

OOV processor

- We adopt MIMICK[3] to handle OOV words at runtime.
  - Learn a function from spelling to distributional embeddings.

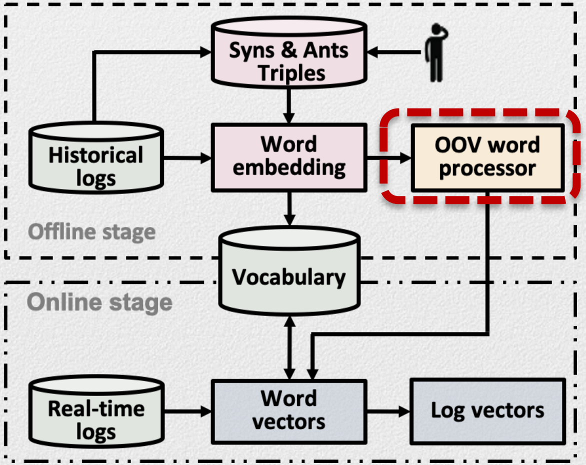
Historical logs:

L_1. Interface ae3, changed state to down
L_2. Interface ae3, changed state to up
L_3. Interface ae1, changed status to down
L_4. Interface ae1, changed status to up

Real-time logs:

L_5. Vlan-interface vlan22, changed state to down
L_6. Vlan-interface vlan22, changed state to up

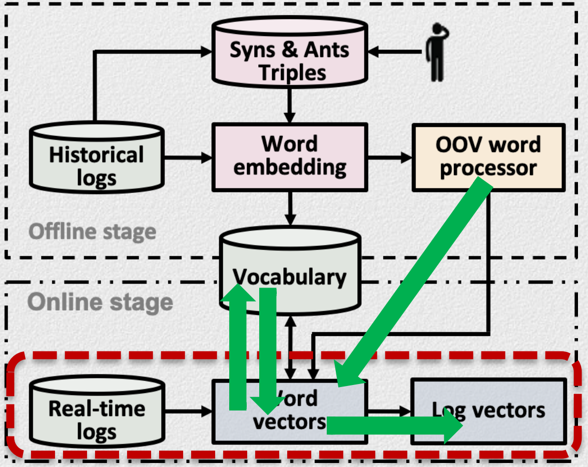
Out-of-vocabulary	Vlan-interface
Relation triples	(interface, changed, state)
Antonym pairs	(down, up)
Synonym pairs	(state, status)



[3].Yuval Pinter, Robert Guthrie, and Jacob Eisenstein. Mimicking word embeddings using subword rnns. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 102–112, 2017.

Log vector generation (Online stage)

1. Determine whether each word in logs is in vocabulary
2. Convert existing words to word vectors
3. Assign a new embedding vector to the OOV word
4. Calculate the log vector by averaging of its word vectors.



Evaluation

Experimental setting

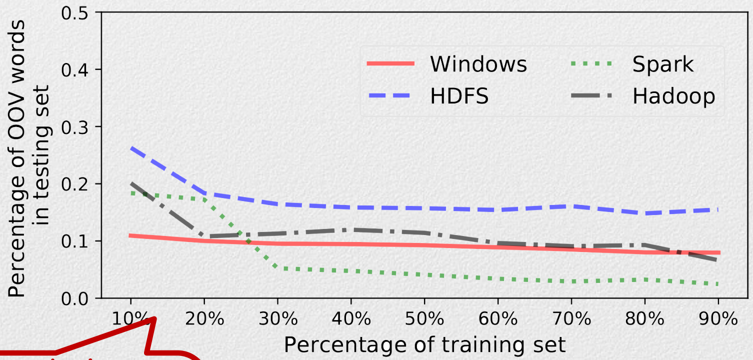
- Datasets:

Datasets	Description	# of logs
HPC	High performance cluster	433,489
HDFS	Hadoop distributed file system	11,175,629
ZooKeeper	ZooKeeper service	74,380
Hadoop	Hadoop MapReduce job	394,308

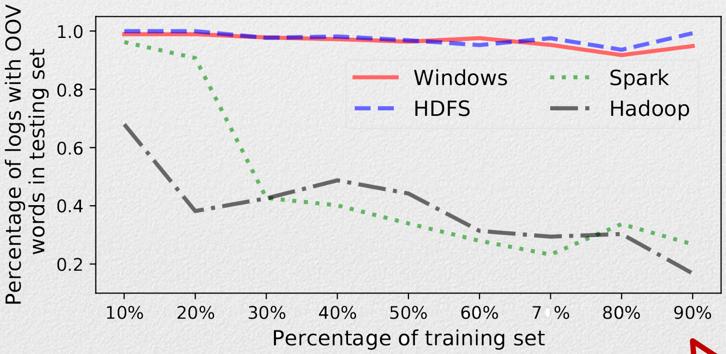
- Experimental setup:
  - Linux server with Intel Xeon 2.40 GHz CPU

Measurement of OOV

- To highlight the challenge in processing OOV words
  - Generate training sets with the percentage of original logs ranging from 10% to 90% and regard the remaining logs as the testing set



Measurements of OOV words



Measurement of logs with OOV words

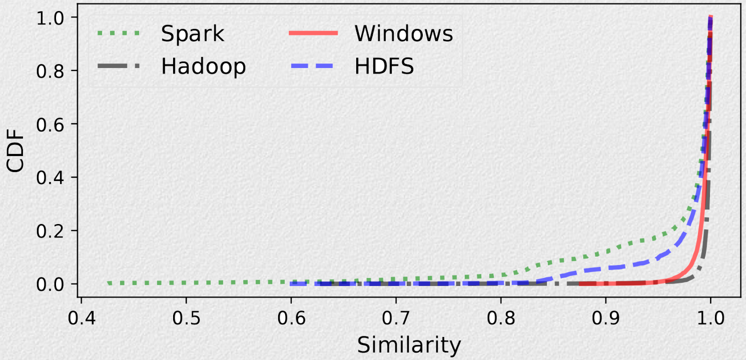
OOV words has a big percentage when trained on a smaller sample

It's important to handle OOV words

Always more than 90% logs contain OOV words in Spark/Windows

Evaluation of OOV processor

- Randomly select a word in each log
- Changed one of the letters to make the word as an OOV
- Test the similarity between the changed log and the original log



Distribution of Logs' Similarity

Dataset	Spark	HDFS	Windows	Hadoop
Similarity	0.964	0.984	0.993	0.996

Average similarity when Log2Vec processes logs with OOV words

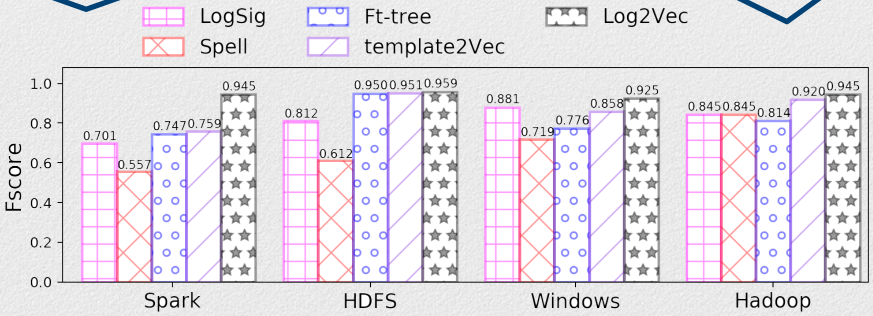
Log-based service management task

- Online log classification
  - Baselines: LogSig, FT-tree, Spell, template2Vec
  - Divide: 50% training set and 50% testing set

Average Fscore of Log2Vec is 0.944

Average Fscore of baselines 0.745

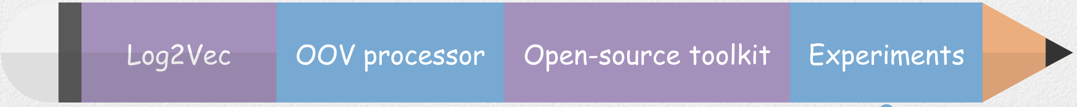
Log2Vec is stable



Comparison of log classification when use 50% training logs

Summary

Summary



Semantic-aware representation framework for online log analysis

A mechanism for generating OOV word embeddings when new types of logs appear

We have open-sourced Log2Vec,

The results are excellent

Thanks

mwb16@mails.tsinghua.edu.cn

Open source toolkit: https://github.com/WeibinMeng/Log2Vec