Deep Information Retrieval for Malware Searching System

Wonkyung Lee, Uijung Chung, Hyeongjin Byeon, Junyeon Weon

Deep Information Retrieval for Malware Searching System

Wonkyung Lee, Uijung Chung, Hyeongjin Byeon, Junyeon Weon

Index

Part1

- Motivations - Malware retrieval systems
- Properties - Semantics awareness
- Methods - Metric learning

Part2

- Demo - Training and starting the system.
- Demo - Querying tests

Motivations

Q1. Why do we need malware retrieval systems?

Motivations

1. To help security experts
   - Categorize malicious codes.
   - Write reports of malware.
   - If they could search semantically similar malware samples, it’ll be useful for above tasks.

Motivations

2. Beyond a good feature extraction skills

- A lot of researches for getting good feature.
- We focused on how to solve the real problems using good features.

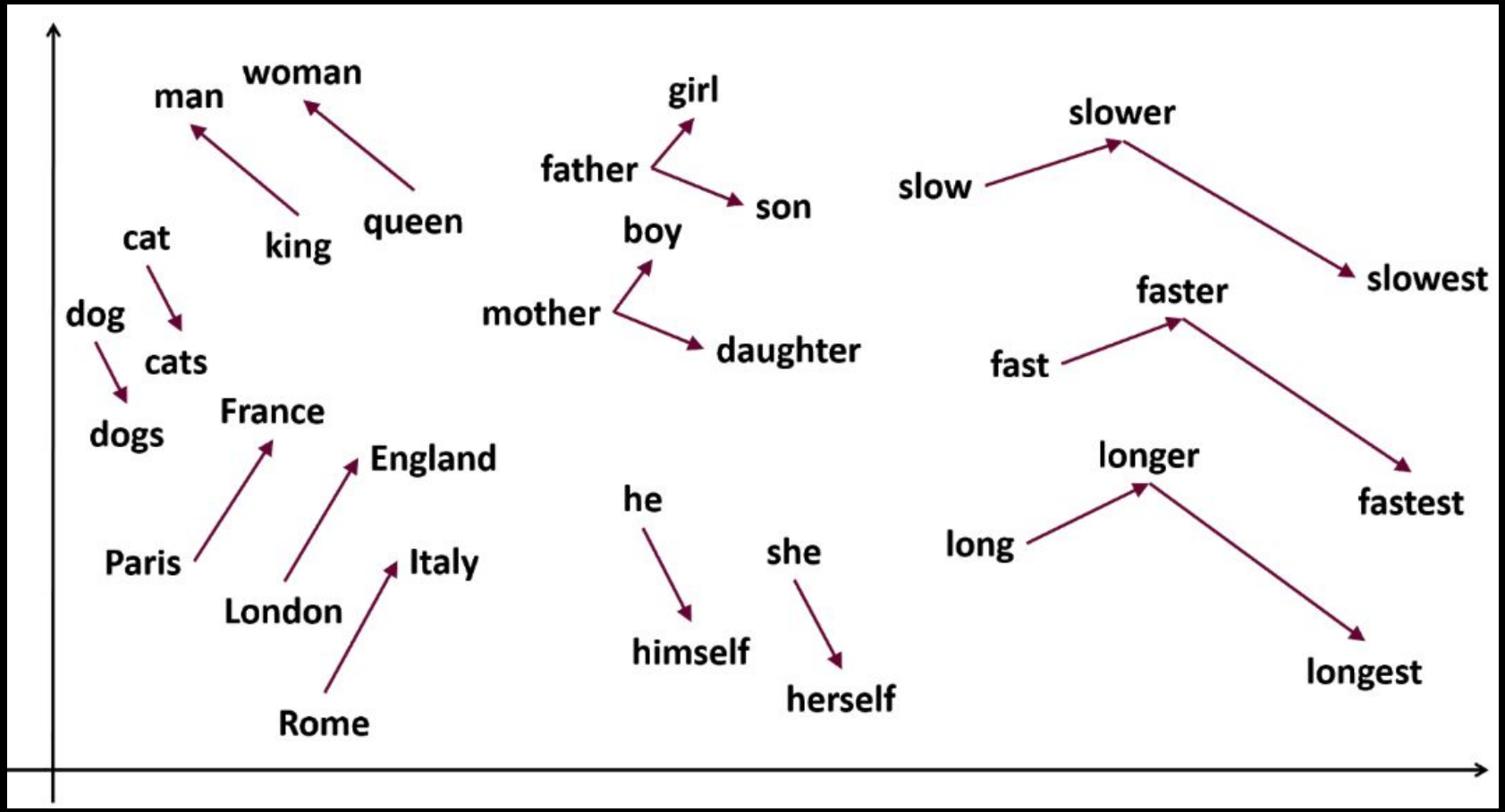
Properties

Q2. Why the malware retrieval system should be semantics-aware?

Semantics-awareness

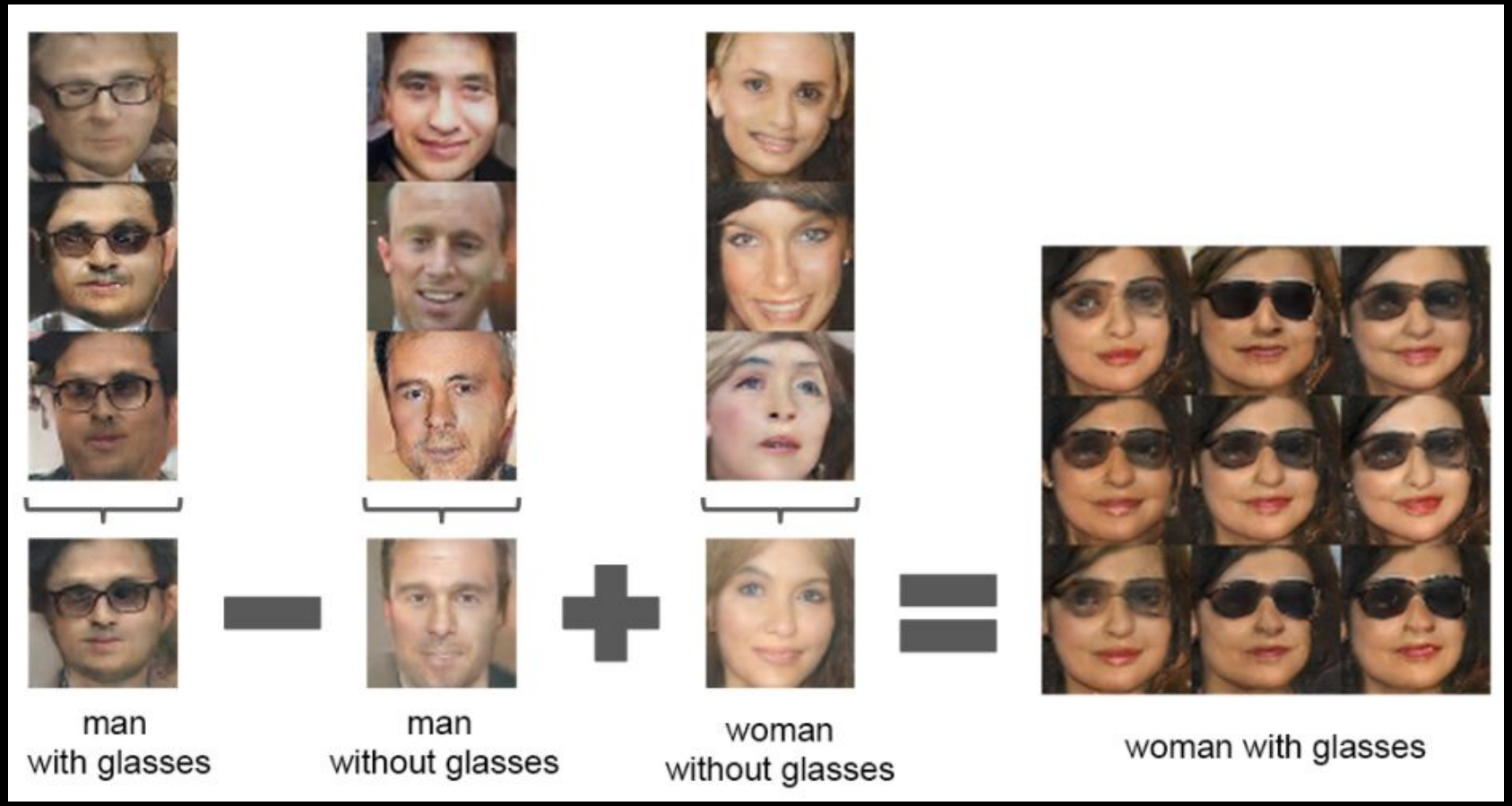
- Meaning
  - Property of system which uses our semantic relationships between objects.
- Examples
  - Word2Vec : Natural Language -> Vector space
  - Deep IR : Image -> Vector space
  - DCGAN, CVAE : Image <-> Vector space

Semantics-awareness



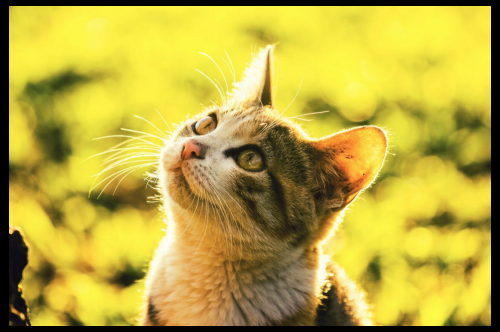
주석을 입력합니다.

Semantics-awareness



Semantics-awareness

- If visual retrieval system is semantics-aware, then
  - It can find semantically similar images.



Semantics-awareness

Semantics-awareness

For the malwares,

- We can find samples which are semantically similar with a queried sample.
- We can find samples which have properties like “ransom, smspay, dropper”.
- We can find samples which have properties that “dropper” is removed from a queried sample.

Methods

Q3. Why do we use Metric learning?

Metric Learning

- Metric is a distance function.
  - Examples
    - Euclidean distance
    - Cosine similarity
- Learn how similar two objects are.

Similarities between Malwares

- Structural similarity
  - Static features (size, libraries, permissions, entropy, …)
- Semantic similarity
  - Static + Dynamic features (behaviors, logs, packets…)

Metric Learning

- Objectives
  - Embed malware samples on vector space.
  - Each embedded vector is made up of property vectors and some noise that includes the other informations.
  - Semantically similar samples should be embedded nearly in the vector space.

Metric Learning

- Parameterized distance function
  - $D(s_1, s_2; \theta) = d$
- Deep neural network
  - Universal function approximator
  - Examples of metric learning by deep learning
    - Siamese network, triplet loss
    - Center loss

Objective Function

- Objective Function
  - Multi-label Center Loss

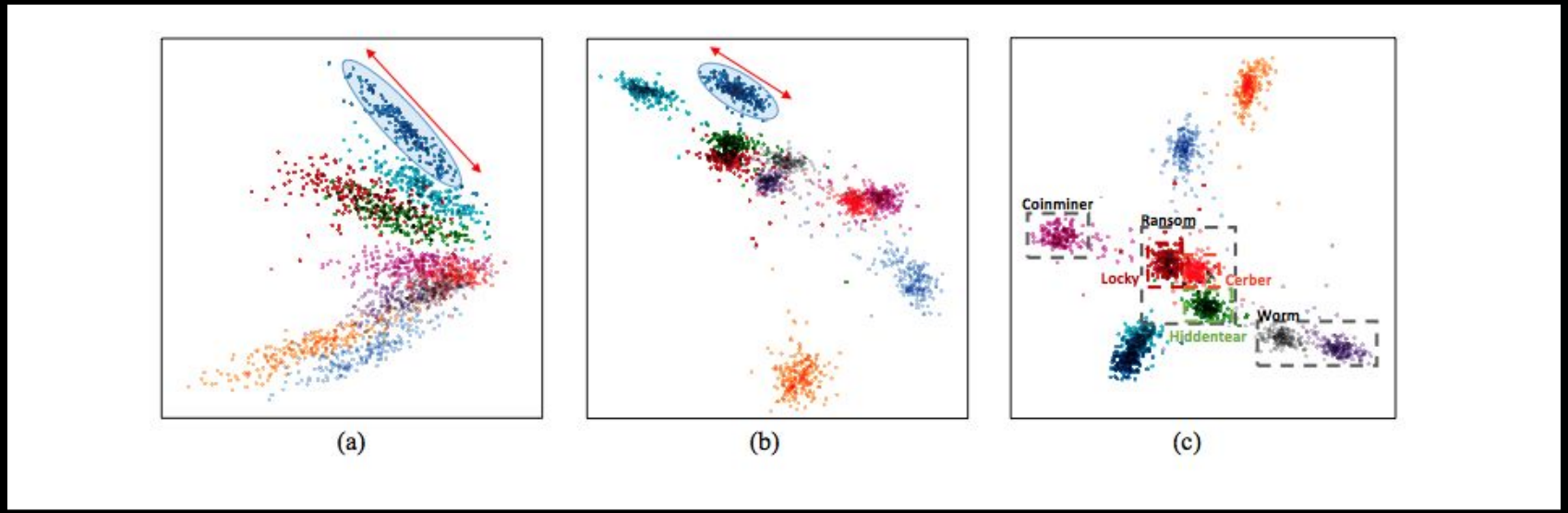
$$L = L_s + \lambda L_c$$

$$= -\frac{1}{N}\sum_i \sum_j y_{mij} \log \hat{y_{ij}} + \lambda \frac{1}{N} \sum_i (\mathbf{s}_{\text{target}} - h(v_i; \theta))^2$$

where

$$\hat{y_{ij}} = \frac{\exp(Wh(v_i; \theta) + \mathbf{b})}{1 + \exp(Wh(v_i; \theta) + \mathbf{b})}$$

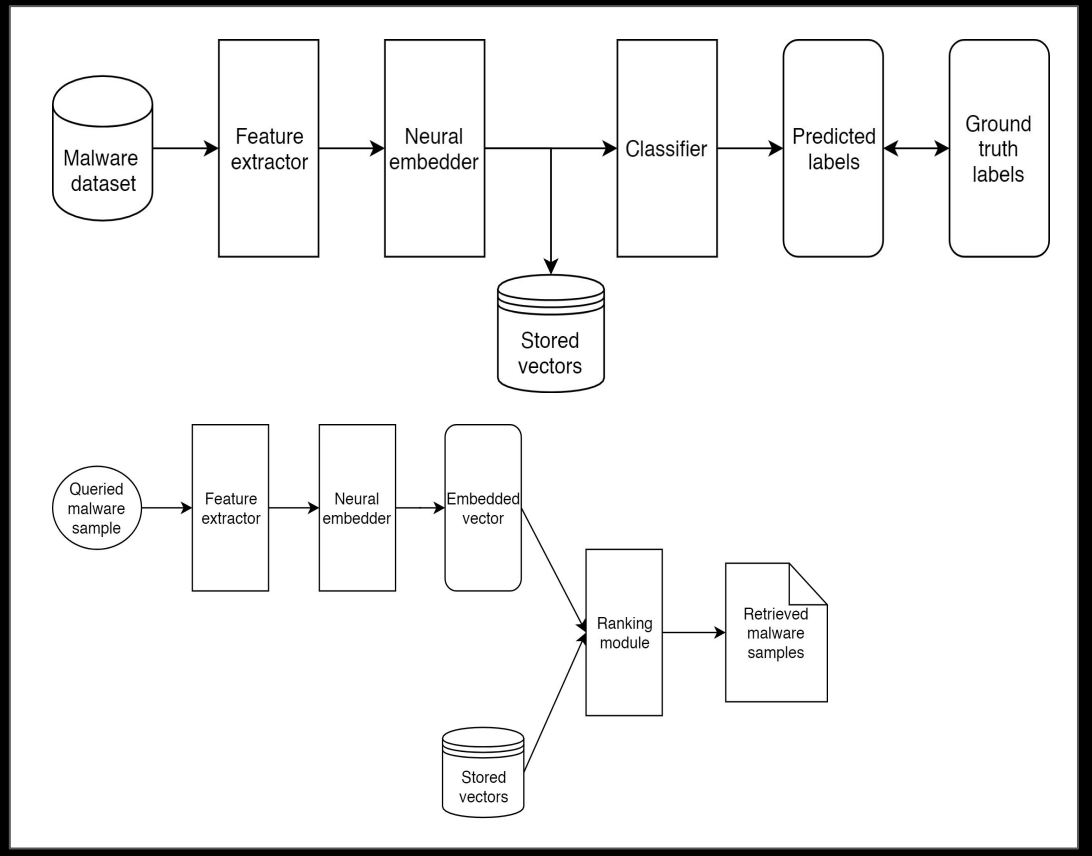
Objective Function



- Decrease inner class variance and increase inter class variance.
- If samples share properties, then they are located nearly.

Structure

- Structure of our proposed system.
  - Feature extractor
  - Neural embedder
  - Classifier
  - Ranking module



More Specifically

- Preparing dataset
  - Multi-tagged malware samples
  - Attach multi-tags to malware samples based on their properties.

Qualitative Results

Dataset	Querying trainset						Querying validset					
Metric	Precision			Weighted precision			Precision			Weighted precision		
k	top1	top10	top100	top1	top10	top100	top1	top10	top100	top1	top10	top100
Weighted centerloss (sum)	0.9846	0.9743	0.9374	0.9869	**0.9802**	**0.955**	0.8717	0.862	0.8272	0.8808	0.8746	0.8534
Weighted centerloss (avg)	**0.9856**	0.9772	**0.9481**	0.9855	0.9779	0.9528	**0.8929**	**0.8837**	0.8559	**0.8893**	0.8815	0.8591
Centerloss (sum)	0.9843	0.9737	0.9358	0.984	0.9733	0.9388	0.8736	0.8676	0.8373	0.8752	0.872	0.8462
Centerloss (avg)	0.987	**0.9776**	0.9474	0.9872	**0.9776**	0.9472	0.8845	0.8779	**0.8562**	0.8892	**0.8829**	**0.8631**
Multi-label (baseline2)	0.9764	0.9617	0.9115	0.9752	0.9593	0.9094	0.8871	0.8713	0.8184	0.8869	0.8704	0.8236
Single-label (baseline1)	0.9035	0.8727	0.8061	0.9004	0.8667	0.7946	0.8489	0.8231	0.7446	0.8434	0.8151	0.733

Dataset	APK19000			PE1300		
Variance	Intra-class	Inter-class	Inter/Intra	Intra-class	Inter-class	Inter/Intra
Weighted centerloss (sum)	0.0473	2.793	59.0486	0.129	2.169	16.8140
Weighted centerloss (avg)	**0.0101**	**1.1906**	**117.8812**	**0.0361**	**1.2698**	**35.1745**
Centerloss (sum)	0.2268	5.2967	23.3541	0.1748	1.8299	10.4685
Centerloss (mean)	0.0258	1.2785	49.5543	0.1332	1.6826	12.6321
Multi-label (baseline2)	2.4956	21.8435	8.7528	25833.582	840.54	0.0325
Single-label (baseline1)	1.2103	15.18	12.5423	2310.7649	267.683	0.1158

Demo - Training MR System

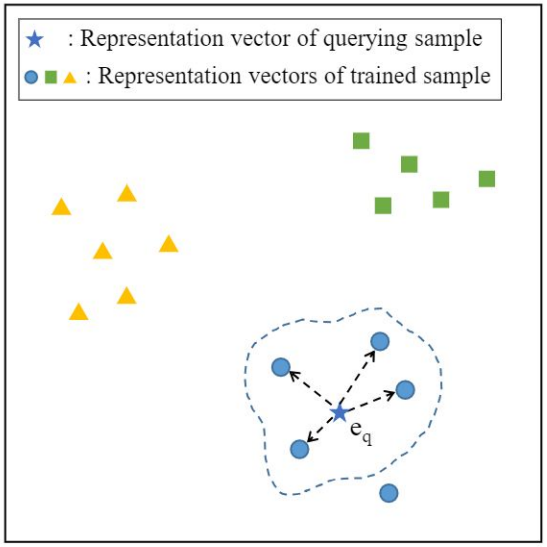
https://github.com/est-ai/malware-retrieval

Demo - Querying test

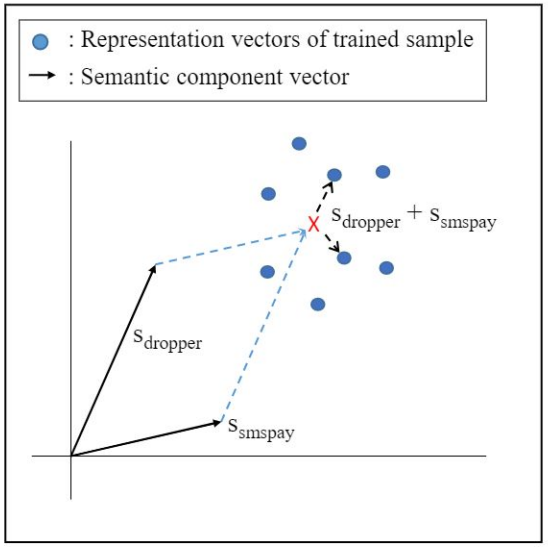
Sample Analysis

- Statistical Analysis - Probabilities of tags

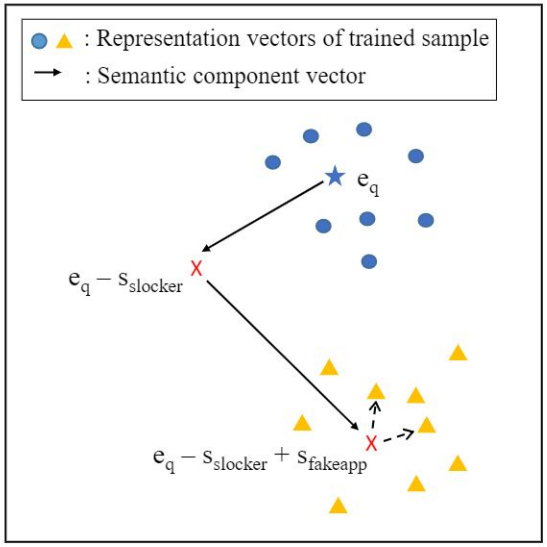
Verify Semantics-awareness



(a)



(b)



(c)

Querying by sample

Querying by tags

Querying by sample + tags

Issues

- Generalization
- Tagging

Demonstration

Thank you