

# Graph Machine Learning for Entity Linking

Rhicheek Patra

Oracle Labs Zürich

*[rhicheek.patra@oracle.com](mailto:rhicheek.patra@oracle.com)*

# Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

# Agenda

- 1 ➤ Graph Data Model
- 2 ➤ Entity Linking Problem
- 3 ➤ Graph Machine Learning
- 4 ➤ Entity Linking with GraphML
- 5 ➤ Other Applications of GraphML
- 6 ➤ Summary

# Agenda

- 1 ➤ Graph Data Model
- 2 ➤ Entity Linking Problem
- 3 ➤ Graph Machine Learning
- 4 ➤ Entity Linking with GraphML
- 5 ➤ Other Applications of GraphML
- 6 ➤ Summary

# Benefits of Graph Models

- Some of graph benefits
  - Intuitive data model
  - Aggregate information over heterogeneous data sources
  - Fast query over multi-hop relationships
  - Data visualization and interactive exploration

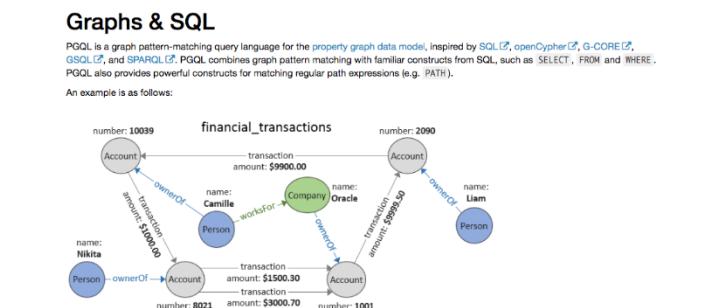
# Parallel Graph AnalytiX (PGX)

- A fast, parallel graph analytics framework
  - Shared memory and distributed version
- Offers 40+ built-in, native graph analytics algorithms
  - PageRank, Centrality, WCC, SCC
- Offers Graph Machine Learning Algorithms
  - DeepWalk, PG2Vec
- Provides a graph-specific query language (PGQL)

```
SELECT v, e, v2
FROM graph
MATCH (v)-[e]->(v2)
WHERE v.first_name = 'Jerald'
AND v.last_name = 'Hilpert'
GROUP BY ... ORDER BY ... LIMIT ...
```

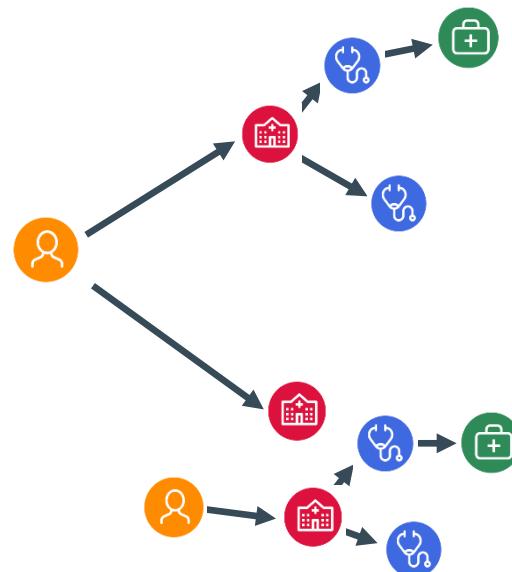
The screenshot shows the Oracle Labs PGX documentation page. The top navigation bar includes links for Welcome, Installation, Tutorials, Use Cases, Reference, and PGX Overview. The main content area is titled "Algorithms" and contains a grid of 12 boxes, each representing a different algorithm: Adamic-Adar, Closeness Centrality and variants, Conductance, Conductance Minimization (Soman and Narang), Cycle Detection Algorithms, Degree Centrality and variants, Eccentricity Algorithms, Eigenvector Centrality, Fattest-Path, Filtered BFS, Hyperlink-Induced Topic Search K-Corr, Label Propagation, Local Clustering Coefficient, Matrix Factorization (Gradient Descent), PageRank and variants, and SVD++.

The screenshot shows the PGQL specification page. It features a red header with the text "PGQL · Property Graph Query Language" and "An SQL-like query language for graphs". Below the header, there are links for 1.2 Specification, 1.1 Specification, 1.0 Specification, Open-Sourced Parser, and Oracle. The main content area contains a brief introduction to PGQL, stating it is a graph pattern-matching query language for the property graph data model, inspired by SQL, openCypher, G-CORE, GSQl, and SPARQL. It combines graph pattern matching with familiar constructs from SQL, such as SELECT, FROM and WHERE. The page also includes a diagram illustrating a graph structure with nodes like Account, Person, Company, and Transaction, and relationships like ownerOf, transaction, and connected.



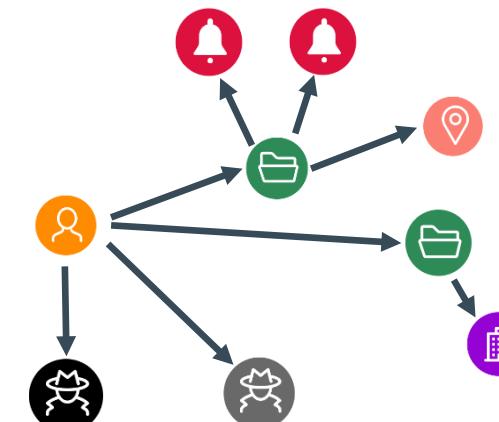
# Graph in Various Domains

## Health Science



- Patient
- Admissions
- Diagnosis
- Procedures

## Financial Services



- Customer
- Account
- Address
- Alert
- Institution
- ICIJ
- Worldcheck

# Agenda

- 1 ➤ Graph Data Model
- 2 ➤ Entity Linking Problem
- 3 ➤ Graph Machine Learning
- 4 ➤ Entity Linking with GraphML
- 5 ➤ Other Applications of GraphML
- 6 ➤ Summary

# Entity Linking (EL)

- Connecting words of interest to unique identities (e.g. Wikipedia Page)

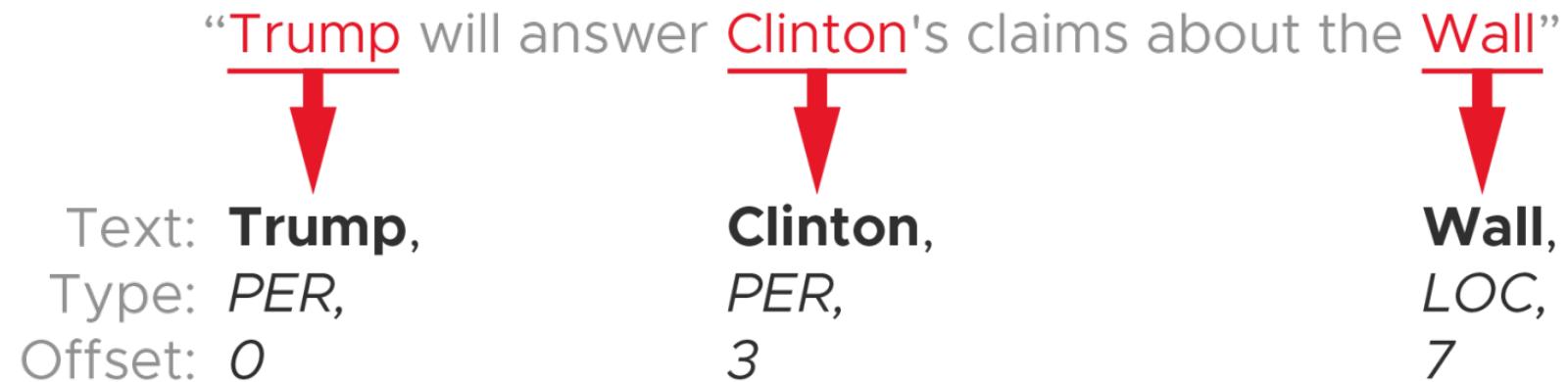


- Use cases
  - **Search Engines**, for semantic search
  - **Recommender Systems**, to retrieve documents similar to each other
  - **Chat bots**, to understand intents and entities

# Standard EL Pipeline

An EL system requires 2 steps:

1. **Named Entity Recognition (NER):** spot **mentions** (a.k.a. Named Entities)
  - ❖ High-accuracy in the state-of-the-art<sup>[1]</sup>

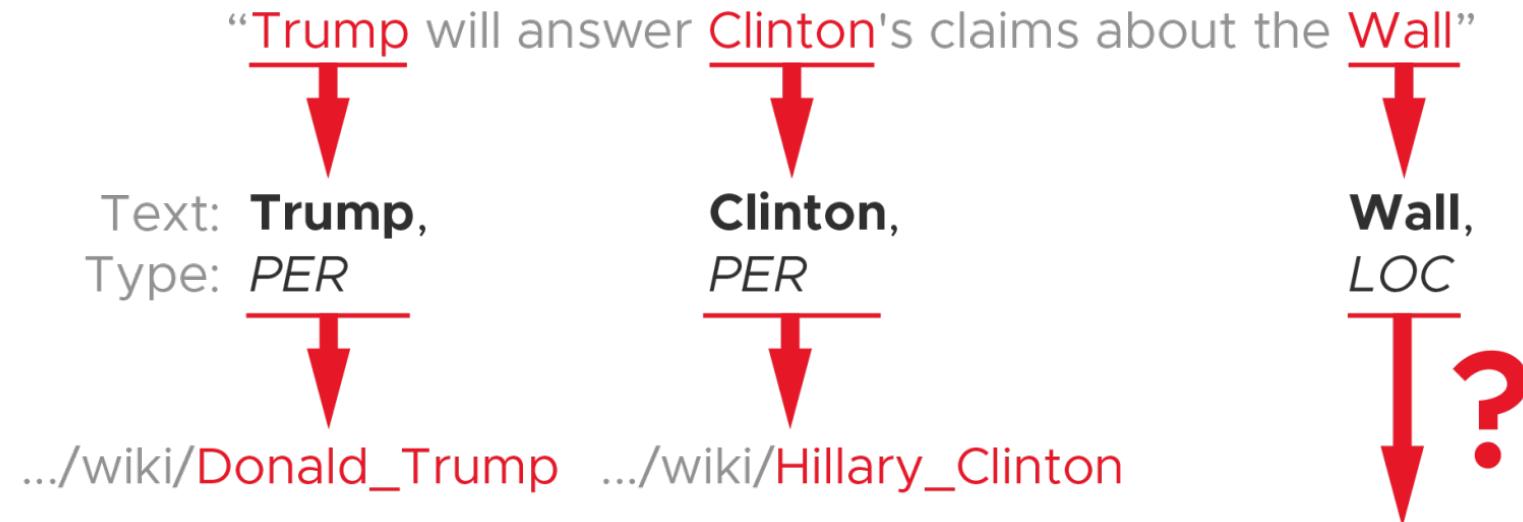


[1] Huang, Zhiheng, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF models for sequence tagging."

# Standard EL Pipeline

An EL system requires 2 steps:

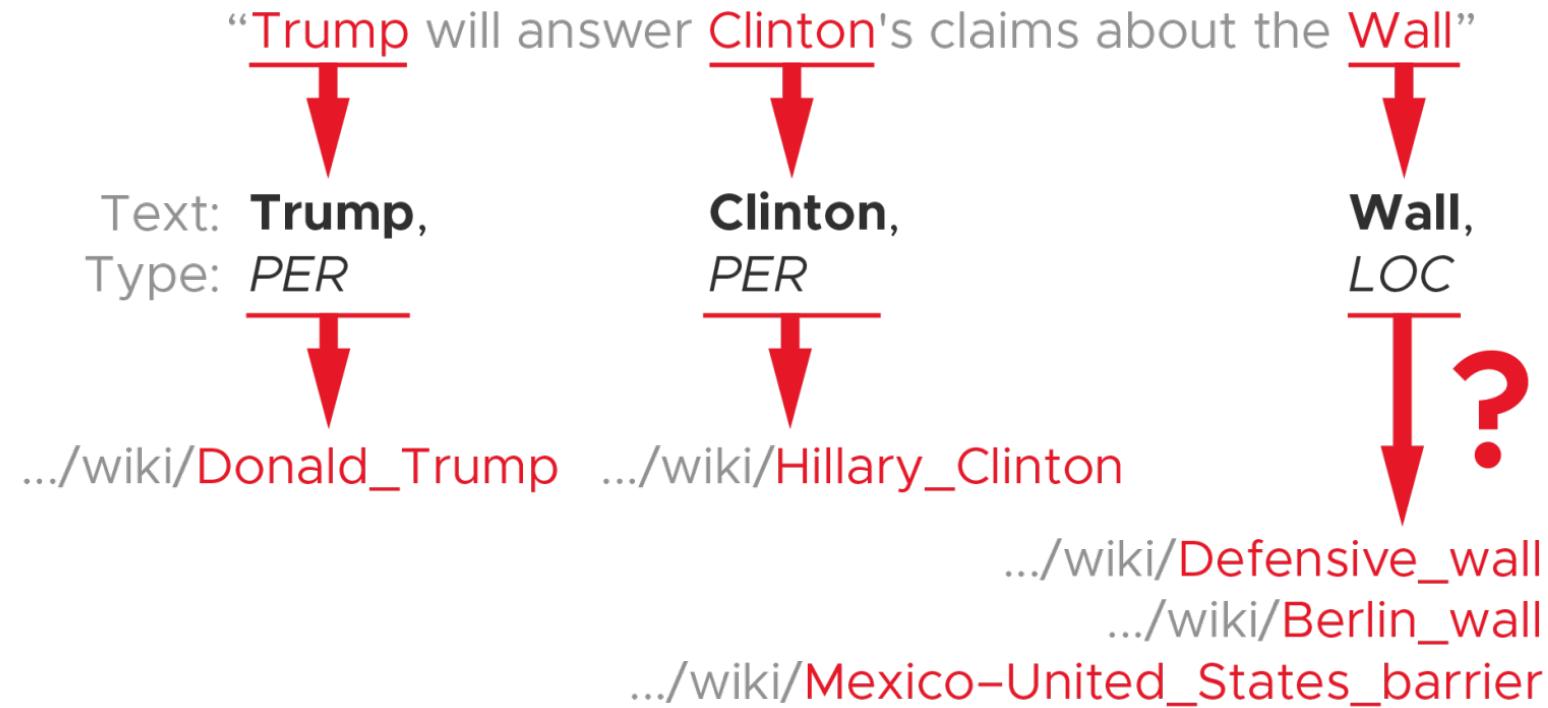
## 2. Entity Linking: connect mentions to entities



# Standard EL Pipeline

An EL system requires 2 steps:

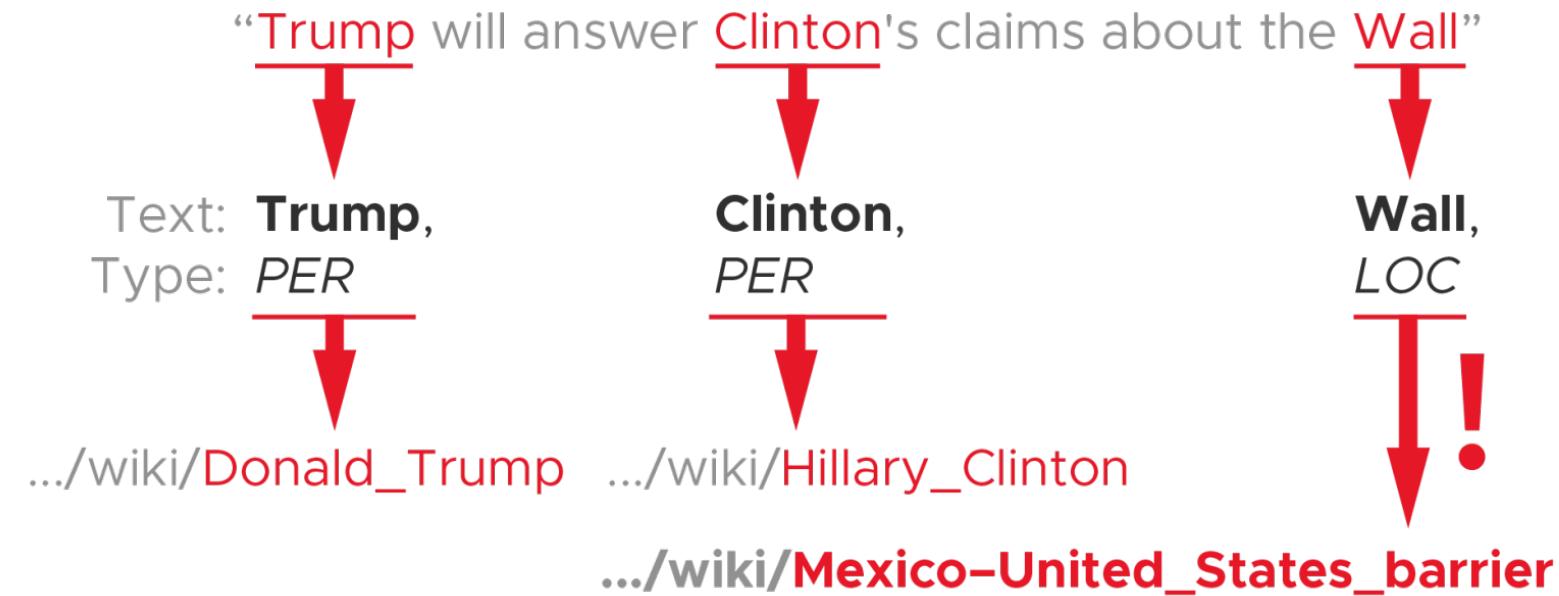
## 2. Entity Linking: connect mentions to entities



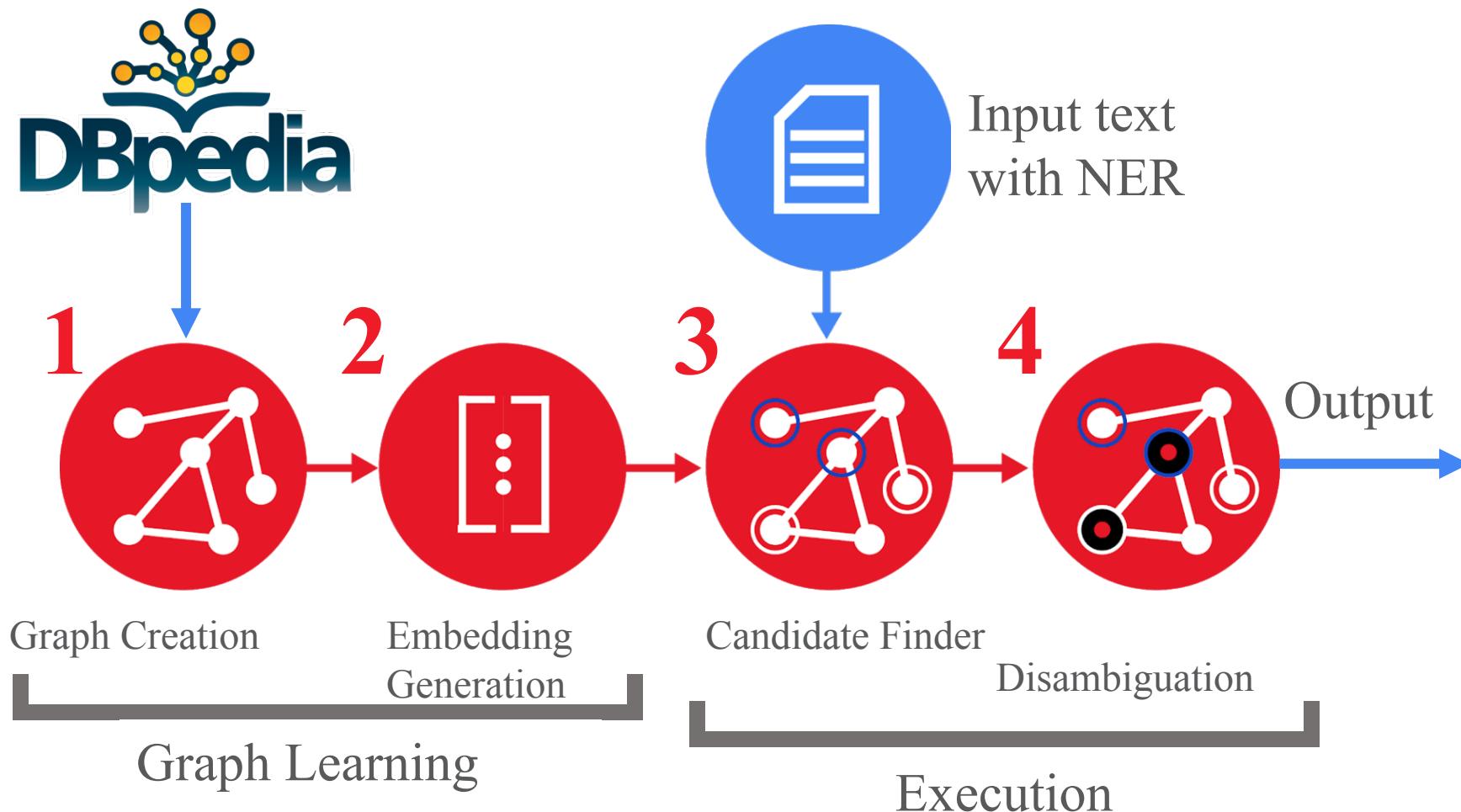
# Standard EL Pipeline

An EL system requires 2 steps:

## 2. Entity Linking: connect mentions to entities



# Our Graph-based EL Pipeline

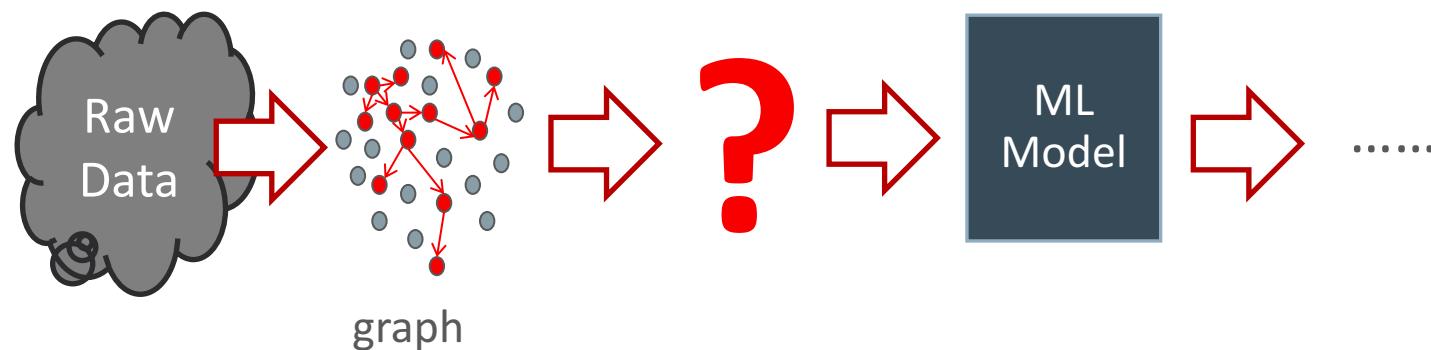


# Agenda

- 1 ➤ Graph Data Model
- 2 ➤ Entity Linking Problem
- 3 ➤ Graph Machine Learning
- 4 ➤ Entity Linking with GraphML
- 5 ➤ Other Applications of GraphML
- 6 ➤ Summary

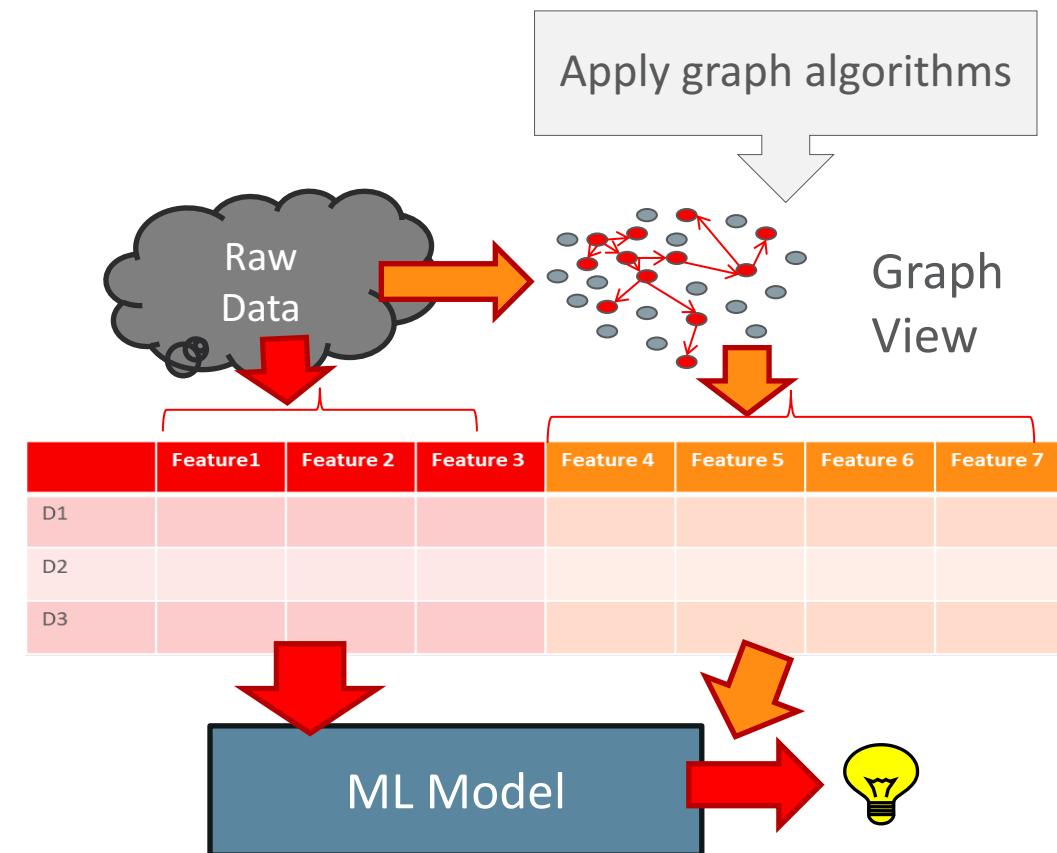
# Feeding graph data into ML pipeline

- Goal: want to apply Machine Learning techniques using graph signals
  - Need some form that are suitable for feeding into conventional ML pipeline
  - but still carries the information in the graph
- ... How can this be done?



# Feature Generation via Graph Algorithm

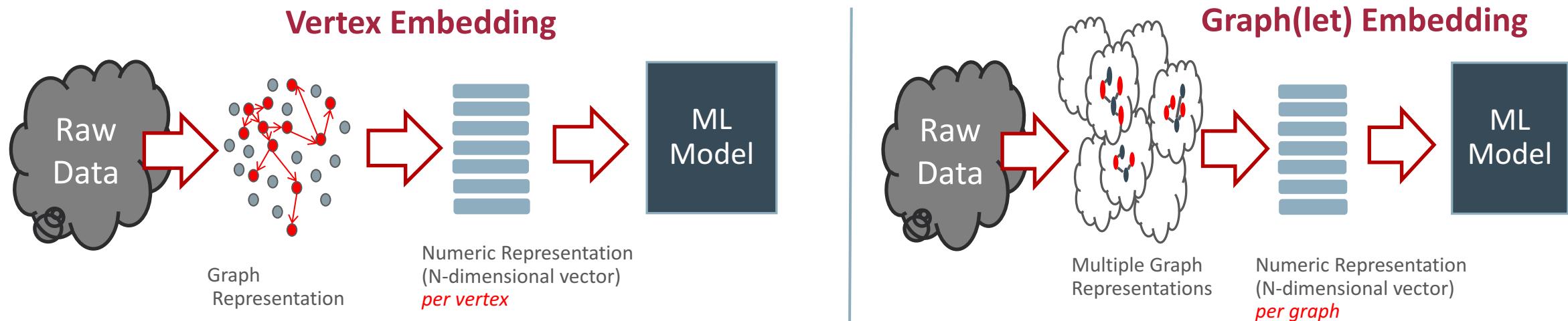
- Approach
  - Compute (various) graph algorithms to generate some numeric values
    - In/out degree
    - Eigenvector centrality
    - Pagerank
    - Between-ness centrality
  - Feed the output of graph algorithms into ML model
- Rationale
  - Each graph algorithm result contains certain characteristics of the graph data
  - Combination of those result would keep information about the graph structure



# Machine Learning and Graphs

- Still there are issues
  - Applied seemingly arbitrary set of algorithms for extracting features
  - Would it work for other applications?

- Need a systematic methodology that turns graph information into n-dimensional numeric representation, i.e. embedding
- Embedding techniques: vertex embedding and graph embedding

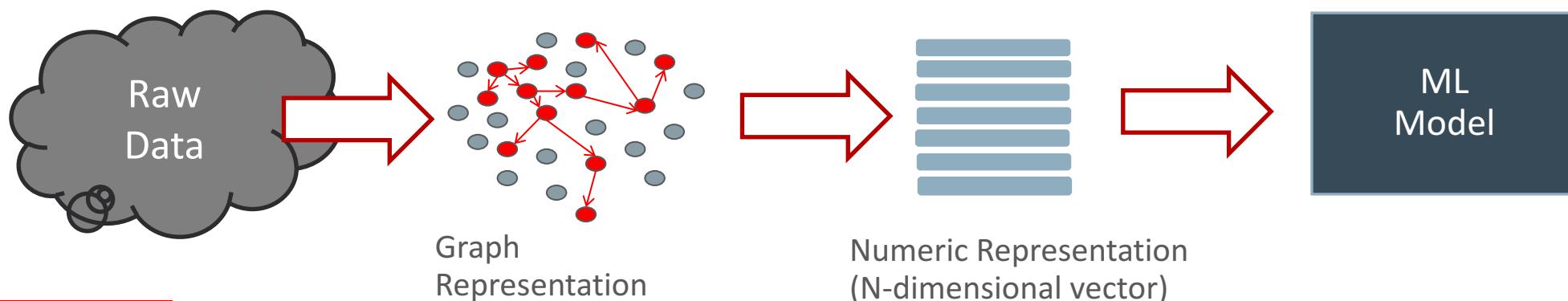


# Vertex Embedding

- Goal: to turn graph into n-dimensional vector
- Want to keep graph (topology) information
  - i.e. Entity distance in distance

$x, y$ : data entity (represented as vertex in graph)  
 $v(x), v(y)$ : n- dimensional vector representation of  $x$  and  $y$

$x, y$  close in graph  $\longleftrightarrow v(x) - v(y)$  close in vector space



# How to achieve this?

- There are several approaches now
  - Academia and Industry
- DeepWalk
  - An early approach that exploits techniques from modern NLP
  - Word2Vec : a ML technique that learns closeness between words from large number of sentences
  - Perform many random walks on the graph and generate traces.
  - Apply W2V technique on them; treating vertices as words.

KDD'14

## DeepWalk: Online Learning of Social Representations

Bryan Perozzi  
Stony Brook University  
Department of Computer Science

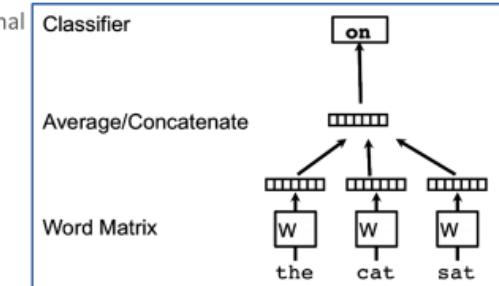
Rami Al-Rfou  
Stony Brook University  
Department of Computer Science

Steven Skiena  
Stony Brook University  
Department of Computer Science

{bperozzi, ralrfou, skiena}@cs.stonybrook.edu

## Word2vec: Word-to-vector model

- Represent each word as a low-dimensional word
- Word similarity = vector similarity
- Key idea: *Predict surrounding words of every word in the context*
- Models:
  - Continuous Bag of Words (CBOW)
  - Skip-gram



Paper: Distributed Representations of Words and Phrases and their Compositionalities, NIPS'13

ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

22

ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

20

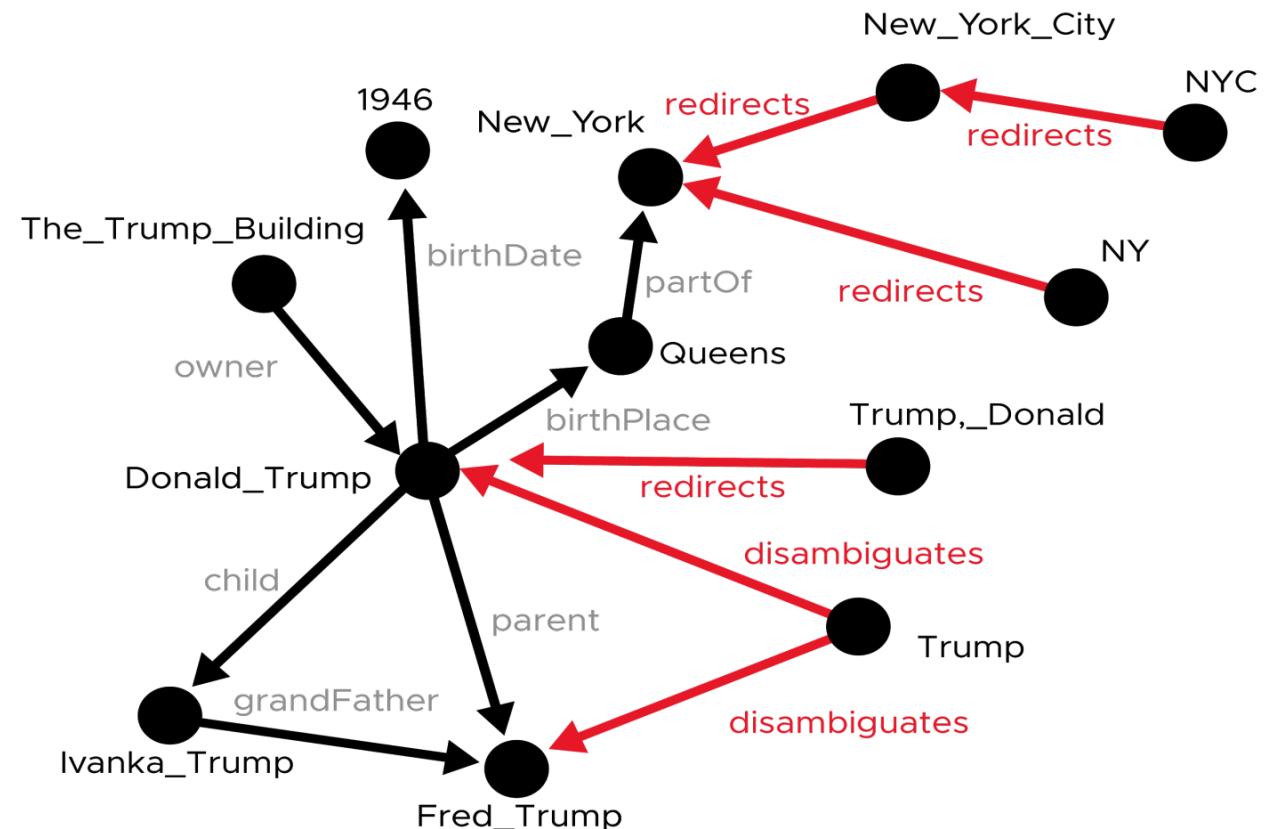
# Agenda

- 1 ➤ Graph Data Model
- 2 ➤ Entity Linking Problem
- 3 ➤ Graph Machine Learning
- 4 ➤ Entity Linking with GraphML
- 5 ➤ Other Applications of GraphML
- 6 ➤ Summary

# Graph Creation

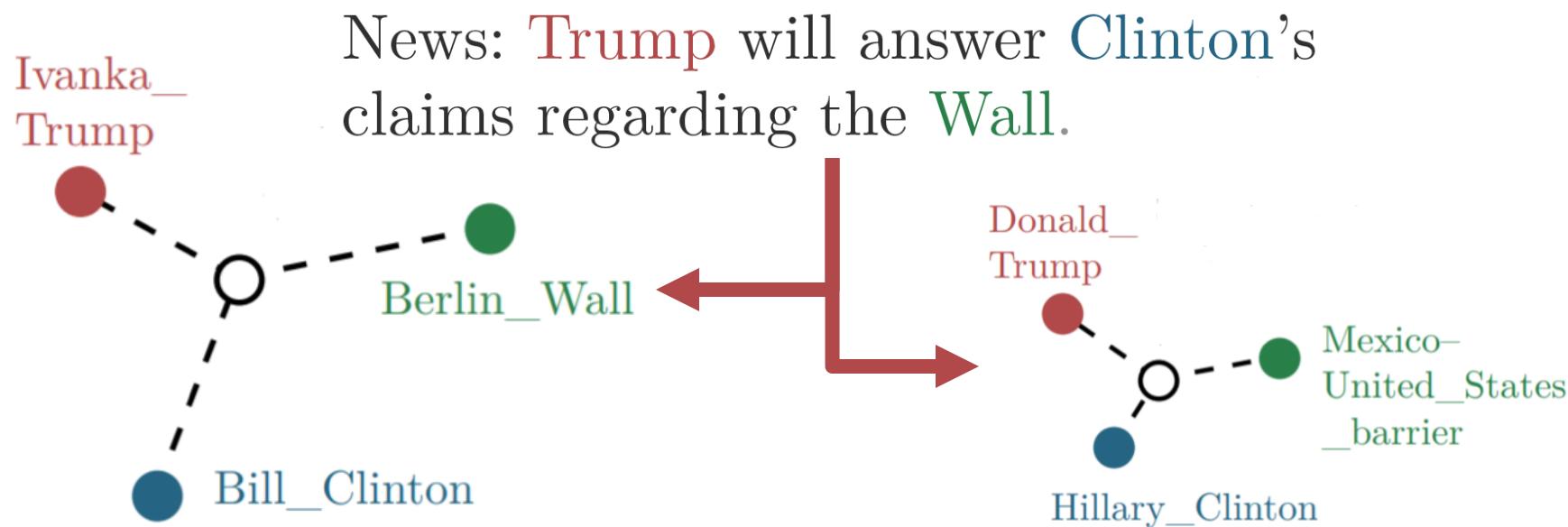
We obtain a large graph from DBpedia

- All the information of Wikipedia
- Stored as RDF triples
- 12M entities
- 170M links



# Graph Embeddings

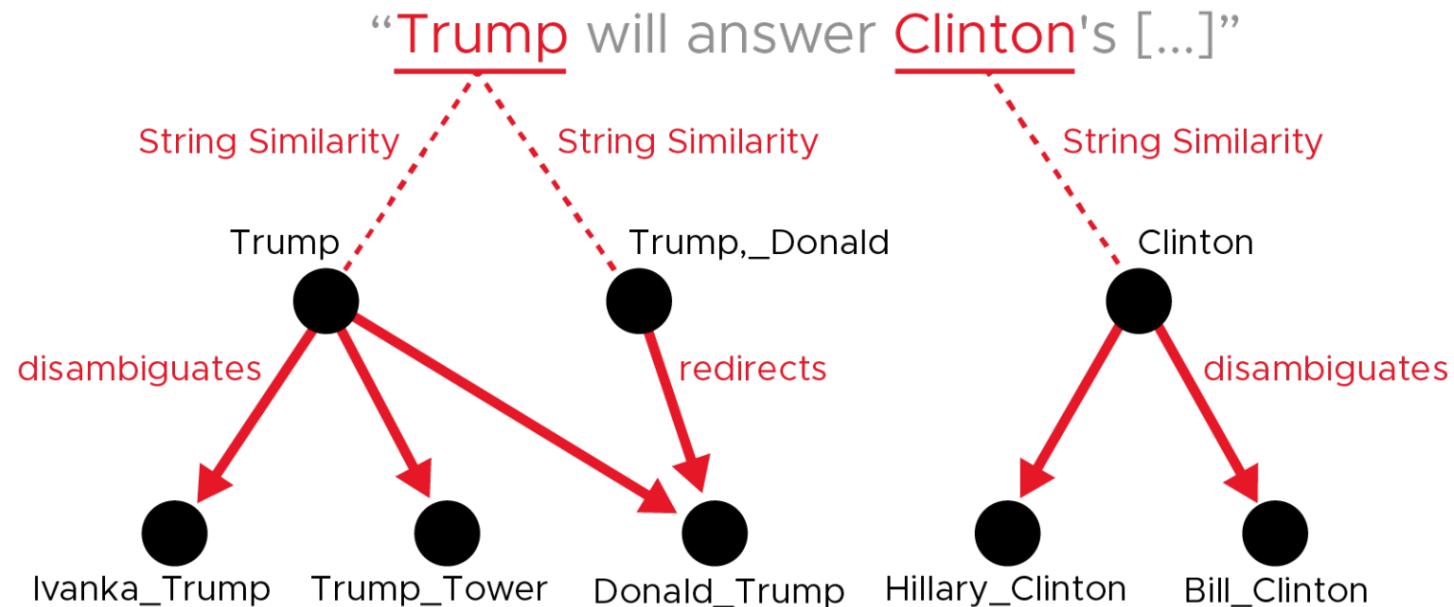
- Generate vertex embeddings using DeepWalk
- **Idea:** entities with the same context should have low embedding distance



# Candidate Finder

Idea: for each mention, select a few candidate vertices with index-based **string similarity**

- Fuzzy matching with 2-grams and 3-grams



# Empirical Evaluation

- We compared against 6 SoA EL algorithms, on 5 datasets
- Our Micro-averaged **F1** score is comparable with SoA supervised algorithms

Data Set	Ours	DoSeR	WK	AIDA	WAT	BB	SL
<b>ACE2004</b>	0.84	<b>0.90</b>	0.83	0.81	0.80	0.56	0.71
<b>AQUAINT</b>	<b>0.86</b>	0.84	<b>0.86</b>	0.53	0.77	0.65	0.71
<b>MSNBC</b>	<b>0.92</b>	0.91	0.85	0.78	0.78	0.60	0.51
<b>N3-Reuters</b>	0.82	<b>0.85</b>	0.70	0.60	0.64	0.53	0.58
<b>N3-RSS-500</b>	0.72	<b>0.75</b>	0.73	0.71	0.68	0.63	0.62

**WK** is Wikifier, **BB** is Babelfy, **SL** is Spotlight

# Agenda

- 1 ➤ Graph Data Model
- 2 ➤ Entity Linking Problem
- 3 ➤ Graph Machine Learning
- 4 ➤ Entity Linking with GraphML
- 5 ➤ Other Applications of GraphML
- 6 ➤ Summary

# Cyber Security

- Cyber security is a very important topic in the battle for Cloud
  - Invalid traffic detection
  - Cyber threat hunting
  - Malware detection
  - ...



- Solution
  - Use *graph* for enhancing cyber security

DZone Security Zone Over a million developers have joined DZone. [Log In](#)

REFCARDZ GUIDES ZONES | Agile AI Big Data Cloud Database DevOps Integration IoT Java Microservices Open Source Performance

## Graphs Are a Game-Changer for Cybersecurity

We see what Eric Spiegelberg, Senior Consultant and GraphAware, has to say about the fast-moving world of cybersecurity and graph databases. Check out this post to learn more!

by Bryce Merki Sasaki MVB · Oct. 09, 18 · Security Zone · Interview

Like (1) Comment (0) Save Tweet 2,803 Views

## Cybersecurity & Graph Technology: An Excellent Fit

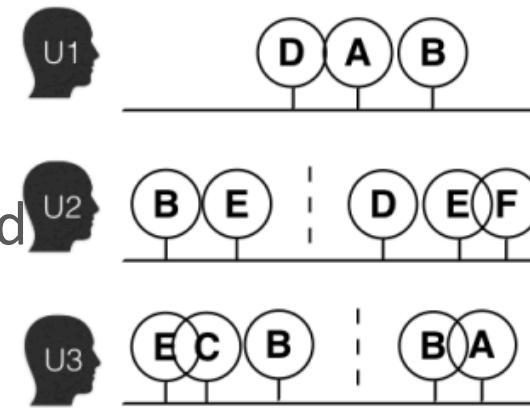
Eric Spiegelberg, Senior Consultant, GraphAware  
Oct 19, 2017 · 5 mins read

# Cyber Security

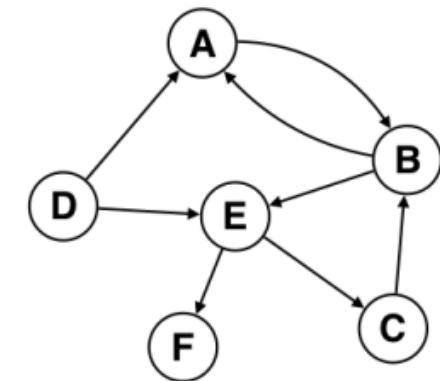
- Use *graph* as a semi-structured *data model*
  - Combines heterogeneous data sources, i.e., security data comes from various logs, traces, events, etc.
  - Captures connections between them, e.g., same IP connects to multiple domains.
- Graph captures connections between data entities
  - Exploit extra signals from graph for anomaly detection (e.g. via Graph ML techniques)
  - Enhance **cyber-threat detection**
- Graph enables interactive, visual exploration of security data
  - Environment for **cyber-threat hunting**

# Retail/E-commerce

- Graph Machine Learning
  - Item-based Collaborative Filtering
  - Construct item-graph from user sessions
  - Compute item embeddings using DeepWalk to capture the sequence information (represented in the item-graph topology)
  - Employ the item embeddings to rank items



(a) Users' behavior sequences.



(b) Item graph construction.

## • Related literature

- Taobao from Alibaba
  - *Billion-scale Commodity Embedding for E-commerce Recommendation in Alibaba* from KDD'18 -> [link](#)
- PinSage from Pinterest
  - *Graph Convolutional Neural Networks for Web-Scale Recommender Systems* from KDD'18-> [link](#)

# Agenda

- 1 ➤ Graph Data Model
- 2 ➤ Entity Linking Problem
- 3 ➤ Graph Machine Learning
- 4 ➤ Entity Linking with GraphML
- 5 ➤ Other Applications of GraphML
- 6 ➤ Summary

# Summary

- Novel EL algorithm that employs Graph Machine Learning
  - Paper on ACM: <https://dl.acm.org/citation.cfm?id=3328499>
- Applications of GraphML approaches to multiple other domains
  - Cyber-Security
  - Financial Crime Compliance
  - Health-Care
  - Retail/E-commerce
  - Program Analysis and others ...
- PGX links:
  - Documentation: [https://docs.oracle.com/cd/E56133\\_01/latest/index.html](https://docs.oracle.com/cd/E56133_01/latest/index.html)
  - Download: <https://www.oracle.com/technologies/developer-tools/pgx-downloads.html>
  - GitHub: <https://github.com/oracle/pgx-samples>

# Integrated Cloud Applications & Platform Services

**ORACLE®**