Maverick: Discovering Exceptional Facts from Knowledge Graphs

12/03/19


Presented by: Juan Carrillo
Candidate for MASc. in Computer Software
Department of Electrical & Computer Engineering
University of Waterloo
Agenda

1. Introduction
2. Maverick core features
3. Experiments
4. Conclusions
5. Discussion
1. Introduction

From knowledge graphs to exceptional facts

Denzel Washington
Denzel Hayes Washington Jr. is an American actor, director, and producer. He has received three Golden Globe awards, a Tony Award, and two Academy Awards: Best Supporting Actor for the historical war drama film Glory and Best Actor for his role as a corrupt cop in the crime thriller Training Day.

Exceptional Facts

Among all the 95486 film directors, Denzel Washington is one of 4665 who appeared in a film.

Isolation: 0.8571

Context

\[ v_1 \xrightarrow{\text{directed_by}} v_2 \]

Among all the 60602 film actors, Denzel Washington is the only one who served as one of executive producers of Film (Chasing the Dream) and Film (Safe House).

Outlierness: 0.9792

Context

\[ v_3 \xrightarrow{\text{film}} v_1 \xrightarrow{\text{actor}} v_2 \]
1. Introduction

The problem, and the Maverick approach

Knowledge graphs (Linked Data)

Maverick approach

Pattern generator

Fact reporter

Exceptionality evaluator

Context evaluator

Manually designed queries

Automated detection of exceptional facts
1. Introduction

Related background

- **Outlier detection**
  - J Gao - 2010
  - On community outliers and their efficient detection in information networks

- **Outlying aspect mining**
  - F Angiulli - 2016
  - Outlying property detection with numerical attributes

- **Maverick: Discovering Exceptional Facts from Knowledge Graphs**
  - SIGMOD’18
    - Comprehensive description and math basis
  - VLDB’18
    - High-level description and demo
Maverick: Discovering Exceptional Facts from Knowledge Graphs

Maverick core features

Page 7
2. Maverick core features

Entity, context, pattern
2. Maverick core features

The overall framework
2. Maverick core features

Main Algorithm

```
FACT-DISCOVER (G, v₀, χ, k, w)
Input: G : the knowledge graph; v₀ ∈ V_G : the entity of interest;
       χ : the exceptionality scoring function; k : the size of
       output; w : the beam width
Output: H : k most exceptional context-subspace pairs

P₀ ← (V₀ = {x₀}, E₀ = ∅) ; // Initial state. x₀ is a variable.
P ← {P₀} ; // Beam.
i ← 1 ; // Iteration number.

while B ≠ ∅ and i ≤ MAX_ITERATION do
  i ← i + 1 ; Btmp ← ∅ ;
  foreach P ∈ B do
    // Obtain contexts of v₀ and matches to P. (Section 3.1)
    Cᵥ₀, Mp ← CONTEXT-EVALUATOR(P, v₀, G);
    foreach C ∈ Cᵥ₀ do
      // Exceptionality Evaluation. (Section 4)
      A ← EXCEPTIONALITY-EVALUATOR(v₀, C, χ, k);
      foreach A ∈ A do H ← H ∪ {(C, A)} ;
      // Find Y − the children of P. (Section 5)
      Y ← PATTERN-GENERATOR(v₀, P, Mp, w, G);
      Btmp ← Btmp ∪ Y ;
      B ← top-w of Btmp based on heuristics h ; // Section 5.4
  return top-k pairs in H based on exceptionality scores;
```
2. Maverick core features

Description of components

Context Evaluator
- Uses a graph query system (Neo4j)
- Takes a pattern as input and returns the matches
- Agnostic to query processing system

Exceptionality Evaluator
- Takes the entity of interest and its contexts
- Looks for the k subspaces with highest scores
- Implements scoring functions

Pattern generator
- Uses beam search to look for promising patterns
- Implements domain specific heuristics
- Beam width can be tuned to requirements
Experiments
3. Experiments

Experimental setup

Single node: 16-core, 32GB RAM

Methods compared

- Beam-Rdm
- Beam-Opt
- Beam-Conv
- Breadth-First

Datasets

WCGoals
49.078 nodes, 158.114 edges, 13 different edge labels, and 11 entity types.

OscarWinners
42.148 nodes, 63.187 edges, 24 distinct edge labels, and 13 entity types.
3. Experiments

Efficiency

Figure 7: The heat map of exceptionality scores ($\chi_0$) and timestamps of all the discovered context-subspace pairs during 2-minute runs for 10 entities of interest ($v_0$) in WCGoals ($k = 10, w = 10$).
3. Experiments

Efficiency

a Varying $k$, fixing $w = 10$.  b Varying $w$, fixing $k = 10$.  
Figure 8: Effect of $k$ and $w$ on the number of evaluated patterns.
3. Experiments

Effectiveness

Figure 13: Score distributions of top-10 context-subspace pairs for 10 entities, 10 2-minute runs per entity.

Figure 14: Average coverage error on 10 entities. Beam width 10.
Conclusions

Maverick: Discovering Exceptional Facts from Knowledge Graphs
4. Conclusions

Takeaways and paper contributions

✓ The authors model an exceptional fact as a context-pattern pair on a knowledge graph
✓ Exponential complexity of search is handled using beam search
✓ The framework is adaptable to domain specific requirements
Thanks for your attention
Discussion
5. Discussion

Research

1. What other heuristics could be proposed in addition to the two presented in the paper? Design requirements for a third heuristic?
2. How Maverick would perform over a completely different dataset? Different proportions among nodes, edges, edge labels, and entity types.
3. What if we add attributes to the nodes and edges? Constraints
4. How to adapt Maverick to work over multiple/linked knowledge graphs?

Industry

5. What is an example of an application over Google knowledge graph?