View Reviews

Paper ID
50

Paper Title
Maverick: Discovering Exceptional Facts from Knowledge Graphs

Track Name
Research Paper First-Round

REVIEWER #1

REVIEW QUESTIONS

1. Overall Evaluation
Weak Reject

2. Reviewer's confidence
Knowledgeable

3. Originality
Medium

4. Importance
Medium

5. Summary of the contribution (in a few sentences)
This paper presents a framework (called Maverick) for discovering exceptional facts in knowledge graphs. An exceptional fact is modeled as a context-subspace pair, and given an exceptionality scoring function, Maverick efficiently searches through a large space using beam search on a Hasse diagram of patterns. Experiments show promising results on real data. While the paper tackles a novel problem, it can also be improved by clarifying what exceptionality means and conducting more experiments on why Maverick is effective for finding exceptional facts.

6. List 3 or more strong points, labelled S1, S2, S3, etc.
S1: Finding exceptional facts is an interesting problem.
S2: Various techniques are proposed to make the framework efficient.
S3: Experiments show promising results.

7. List 3 or more weak points, labelled W1, W2, W3, etc.
W1: Throughout the paper, it was not clear what exceptionality means.
W2: Maverick seems to be a general framework that is not necessarily tied to finding exceptional facts, which is confusing given the claims of the paper.
W3: Experiments are missing in-depth analysis on how Maverick finds exceptional facts.

8. Detailed evaluation. Number the paragraphs (D1, D2, D3, ...)
D1: Despite the emphasis that Maverick discovers exceptional facts, it was not clear throughout the paper what an exceptional fact actually is. Starting from Figure 1, I do not agree that the “Did you know” fact is exceptional (instead, it is merely a fact that can be extracted from sources like Wikipedia). The first concrete definition of exceptionality seems to appear from Section 4.2, which I think is too late. Looking at the scoring functions, it wasn’t clear if they are actually effective in finding exceptional facts. For example, how would they compare with a naive scoring function that simply ranks facts by the inverse of their frequencies? Also, how do the functions deal with noise where some anomalous facts look exceptional just because they are not that frequent? Exceptionality can also be subjective where people have their own preference of what is surprising. I think defining exceptionality is a hard problem, and there needs to be a throughout discussion before moving onto efficiency issues.

D2: It was surprising that Maverick is indifferent to the choice of exceptional scoring functions. That means the framework is general and can be used to find any facts given any scoring function, which is not consistent with what the paper claims. If not, there needs to be a discussion on what properties of exceptional facts Maverick exploits better than other systems.

D3: The experiments did not help with understanding exceptionality, either. It would be useful to have some user study where humans rate the exceptionality of top-K facts found by Maverick. It is likely that there will be many disagreements due to the subjective nature of the problem.

D4: The related work section looks rather thin where the comparison with other work is not clearly described. How does Maverick compare with the paper “Mining Subjective Properties on the Web” (Trummer et al., SIGMOD 2015)?

9. Candidate for a Revision? (Answer yes only if an acceptable revision is possible within one month.)
Yes

10. Required changes for a revision, if applicable. Labelled R1, R2, R3, etc.
Please address D1-D4.

REVIEWER #2

REVIEW QUESTIONS

1. Overall Evaluation
Weak Accept

2. Reviewer's confidence
Knowledgeable

3. Originality
Medium

4. Importance
Medium

5. Summary of the contribution (in a few sentences)

The authors presented the exceptional fact discovery problem inside knowledge graphs. They proposed a beam search based framework with a set of rules and heuristics to find contexts and subspaces. They conducted comprehensive experiments to demonstrate the effectivenss and efficiency of their framework.

6. List 3 or more strong points, labelled S1, S2, S3, etc.

S1, the authors presented a novel and interesting problem, the exceptional fact discovery problem, inside knowledge bases, with various applications for this problem.

S2, they proposed a beam search based framework with a set of rules and heuristics to find contexts and subspaces.
S3, the problem is well formulated and the algorithms are fully explained. The overall writing is good.
S4, they conducted comprehensive experiments to demonstrate the effectivenss and efficiency of their framework.

7. List 3 or more weak points, labelled W1, W2, W3, etc.

W1, the related work and conclusion sections are too short and there is lack of future work for this project.
W2, although the authors presented some motivations behind the context and subspace, it'd better if there are more explanations about why choosing the context and subspace to find exceptional facts for entities, e.g. why using attributes (one-hop or single-edge paths), why not use a path with two more nodes to evaluate the exceptioness of an entity?
W3, it would be great if the authors could present results about some top-ranked patterns and subspaces for some typical entities. And does a complex pattern with many nodes really make sense for common users? A subspace with many attributes would weaken the expressiveness of the exceptional facts for users?

8. Detailed evaluation. Number the paragraphs (D1, D2, D3, ...)

See strong points and weak points.

9. Candidate for a Revision? (Answer yes only if an acceptable revision is possible within one month.)
Yes

10. Required changes for a revision, if applicable. Labelled R1, R2, R3, etc.

R1, strengthen the related work and conclusion, add future work. Maybe move some content on related work from Introduction to Related Work.
R2, add more explanations about the movtivations of using context and subspace to discover the exceptional facts for entities.
R3, present some top-ranked patterns and subspaces for some example entities and resolve questions in W3.

REVIEWER #3

REVIEW QUESTIONS

1. Overall Evaluation
Weak Reject

2. Reviewer's confidence
Knowledgeable

3. Originality
Medium

4. Importance
Medium

5. Summary of the contribution (in a few sentences)
The authors tackle the problem of mining patterns in knowledge graphs and propose Maverick, a framework able to identify graph patterns or exceptional facts in a knowledge graph. Given an entity of interest, an exceptional fact corresponds to a SPARQL basic graph pattern that includes the given entity as a subject of the triple patterns in the discovered graph pattern.
Maverick is empowered with a graph traversal algorithm named Beam, which is able to search in a knowledge base the top-k exceptional facts for an entity of interest; exceptionality scoring functions are used to rank exceptional facts. Beam implements an A* search method that relies on exceptionality scoring functions, and a set of rules and heuristics to efficiently traverse a knowledge graph and reach the nodes reachable from an entity of interest; three representative exceptionality scoring functions are presented as proof-of-concept. Main properties of Beam are formally demonstrated; an extensive empirical evaluation suggest that Beam is more efficient than a baseline approach that traverses knowledge graphs using a Breath-First search method.

6. List 3 or more strong points, labelled S1, S2, S3, etc.
S1. The authors formally define the problem of graph pattern discovery tackled in the paper, as well as main properties of the Beam algorithm.
S2. As proof of concept, exceptionality scoring functions are defined; however, the Beam algorithm is generic and can be tailored for any domain-specific scoring functions.
S3. An extensive empirical evaluation suggests that Beam is able to outperform a baseline on existing knowledge graphs.

7. List 3 or more weak points, labelled W1, W2, W3, etc.
W1. The scalability of the Beam algorithm on large knowledge graphs is not discussed. Experiments were conducted on datasets which contain an almost similar number of nodes (V_G), on which the search space pattern depends (Theorem 5.1). It is not clear whether the algorithm will behave the same, when the order of the graph or pattern increases or decreases.
W2. In the evaluation, state-of-the-art benchmarks for graph pattern mining are not included. For example: i) Gaston: Nijssen and Kok (KDD'04), ii) FFSM: Huan, et al. (ICDM'03), and iii) FTOSM: Horvath et al. (KDD'06)
W3. The authors have ignored the work on frequent graph patterns and graph embeddings. Moreover, the role of the semantics encoded in knowledge graphs is not discussed. The question is how the meaning of the properties will impact the behavior of the Beam algorithm? What happens if there exist sub-properties, or functional or transitive properties? Would the Beam algorithm allow for discovering patterns even in this situation?
W4. The performance of the Beam algorithm is reported in terms of execution time; other metrics like number of visited nodes or edges would allow for a better understanding of how Beam is able to guide the search and prune the search space
W5. Parameters that impact the performance of Beam are not discussed. For example, is the density of graph, in degree or out degree distribution affecting Beam?
W6. The lack of description of the evaluated knowledge graphs prevent to ensure reproducibility of the results
W7. The time complexity of the problem is not discussed.
W8. The authors largely ignored the theory about semantics of SPARQL queries; in consequence, they propose own definitions of pattern and match, which basically correspond to SPARQL basic graph patterns and mappings.
W9. The authors do not discuss what happens if the knowledge graph contains cycles. Hartig et. al (ACM'12) show the complexity of traversing linked data and demonstrated that the problem is undecidable in general. The source of undecidability is that there may be cycles in a knowledge graph.
W10. In the evaluation, it is not clear why the execution time is limited to 2-minute runs. No justification is given.
W11. Excessive notation and symbols make the paper difficult to read. In addition, existing notations are not reused. For instance, the authors may utilize the notations proposed by Perez et. al (ISWC'06) to denote basic graph patterns or mappings and evaluation of a basic graph pattern.

8. Detailed evaluation. Number the paragraphs (D1, D2, D3, ...)
Overall the paper is well-written and motivated; main properties of the Maverick framework and the Beam search method are illustrated with a running example. Albeit exhaustive, the reported empirical evaluation does not reveal the behavior of Maverick with state-of-the-art approaches that allow for mining graph patterns in knowledge graphs. Moreover, because existing benchmarks are not utilized and the datasets used in the study are not available, reproducibility of the reported results cannot be ensured.

Some questions which should be answered by the authors are:

Q1. How does Beam compare with a A* search guided by one of the proposed scoring functions?
Q2. What is the difference between a pattern and a SPARQL basic graph pattern?
Q3. How does Beam compare to knowledge graph embedding methods like TransHR?
Q4. How does the semantics of the properties of the knowledge graph impact on the performance of Beam?
Q5. What happens if there exist sub-properties, or functional or transitive properties? Would the Beam algorithm allow for discovering patterns even in this situation?

9. Candidate for a Revision? (Answer yes only if an acceptable revision is possible within one month.)

Yes

10. Required changes for a revision, if applicable. Labelled R1, R2, R3, etc.

R1 Consideration of graph pattern mining benchmarks in the evaluation

R2 Use of standard definitions and notations in particular for (SPARQL) graph pattern matching

R3 Discussion of the questions raised above