KompaRe: A Knowledge Graph Comparative Reasoning System

Lihui Liu (UIUC)

Boxin Du (UIUC)

Yi Ren Fung (UIUC)

Heng Ji (UIUC)

Jiejun Xu (HRL)

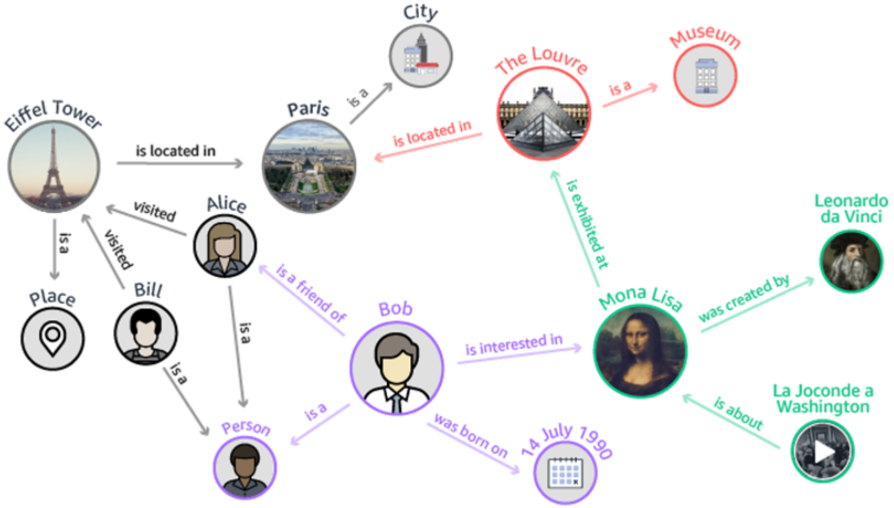
Hanghang Tong (UIUC)





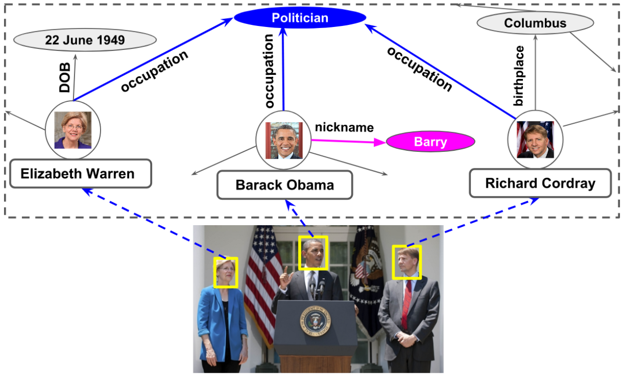

Knowledge Graph

- KG = collection of interlinked entities
  - Objects, events or concepts
  - Multiple types of entities and relations exist
- Facts are represented as triples (h, r, t)
  - <'Paris', 'is_a', 'city'>
  - <'Alice', 'is_friend_of', 'Bob'>
  - …

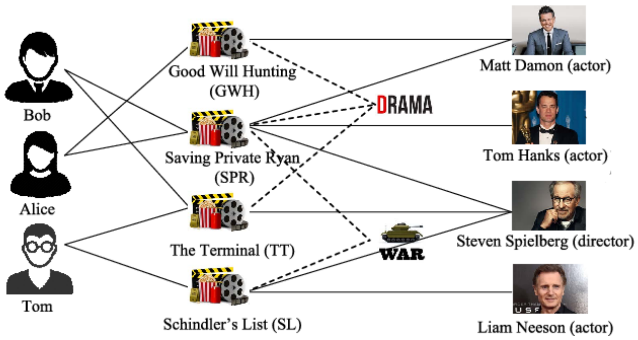


Knowledge Graph Applications

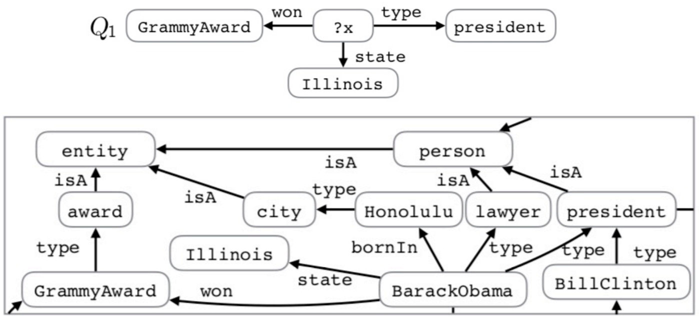
Computer Vision [Y. Fang IJCAI-17]



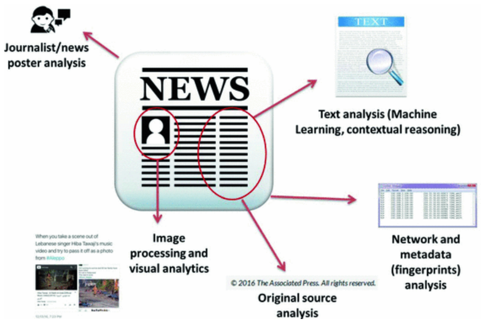
Recommendation [F. Zhang KDD-16]



Question Answering [S. Hu TKDE-18]



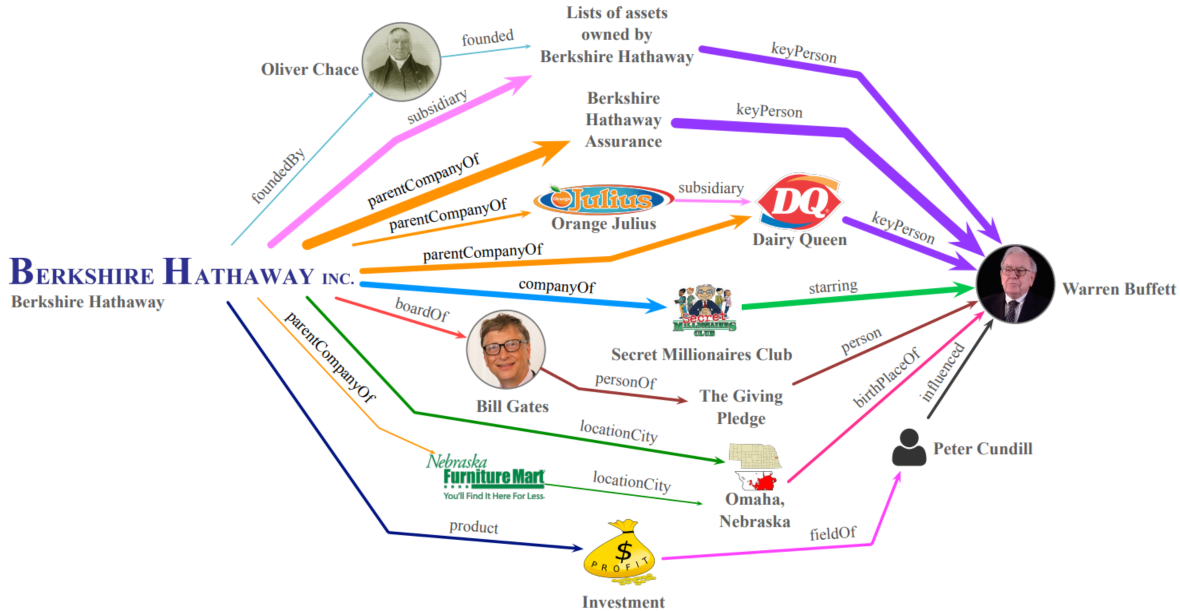
Fact Checking [P. Shiralkar ICDM-17]



- Y. Fang, K. Kuan, J. Lin, C. Tan, and V. Chandrasekhar. 2017. Object Detection Meets Knowledge Graphs(IJCAI-17).
- F. Zhang, J. Yuan, D. Lian, X. Xie, and W. Ma. 2016. Collaborative Knowledge Base Embedding for Recommender Systems(KDD '16).
- S. Hu, L. Zou, J. X. Yu, H. Wang, and D. Zhao. 2018. Answering Natural Language Questions by Subgraph Matching over Knowledge Graphs (TKDE 18).
- P. Shiralkar, A. F Flammini, F. Menczer, and G. Luca. 2017. Finding Streams in Knowledge Graphs to Support Fact Checking (ICDM'17).

Traditional Methods for Fact Checking

Claim: <Berkshire_Hathaway, keyPerson, Warren_Buffett>



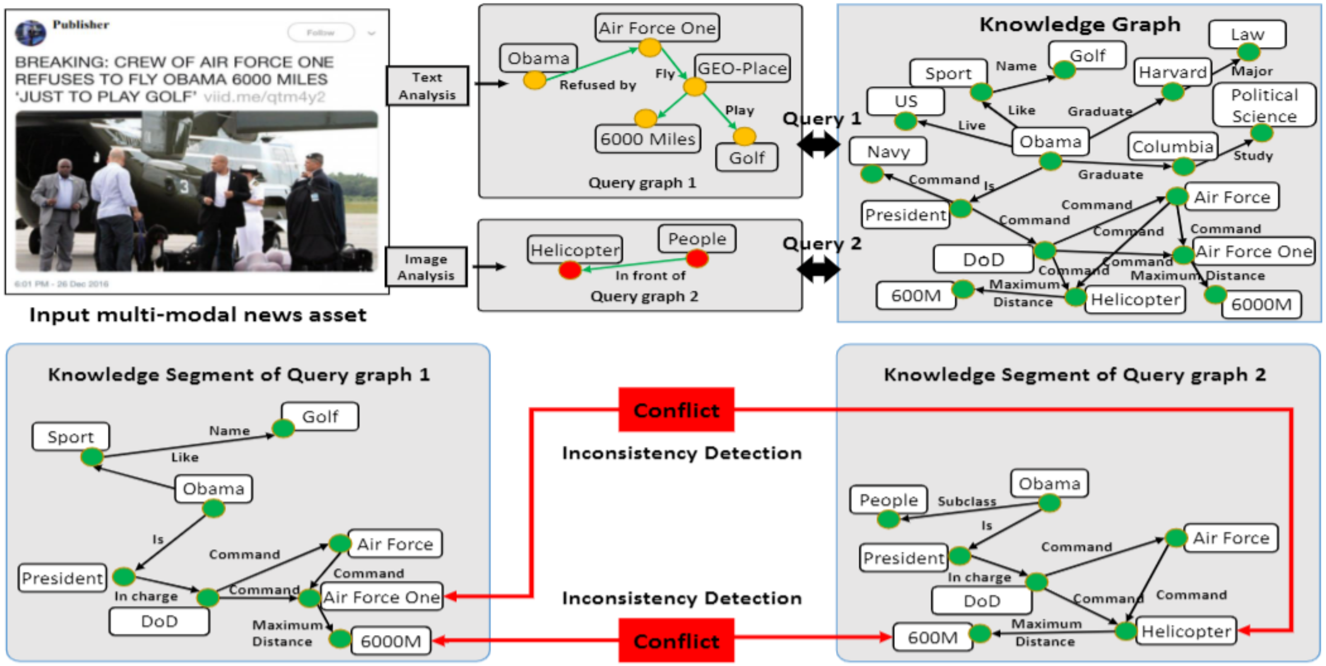
Knowledge Stream [Prashant et al. ICDM’ 17]

- Limitations
  - Only perform fact checking w.r.t. a single claim
  - None of them supports fact checking w.r.t. multiply claims at the same time

- G. Ciampaglia, P. Shiralkar, and Rocha. 2015. Computational Fact Checking from Knowledge Networks.(PLOS ’15).
- P. Shiralkar, A. F Flammini, F. Menczer, and G. Luca. 2017. Finding Streams in Knowledge Graphs to Support Fact Checking (ICDM’17).

Comparative Reasoning

- **Goal**: Find commonality and inconsistency
- An Example



- Advantages: a more complete picture w.r.t. the input clues

- L. Cui, S. Wang, and D. Lee. 2019. SAME : Sentiment-AwareMulti-Modal Embedding for Detecting Fake News. (ASONAM '19).

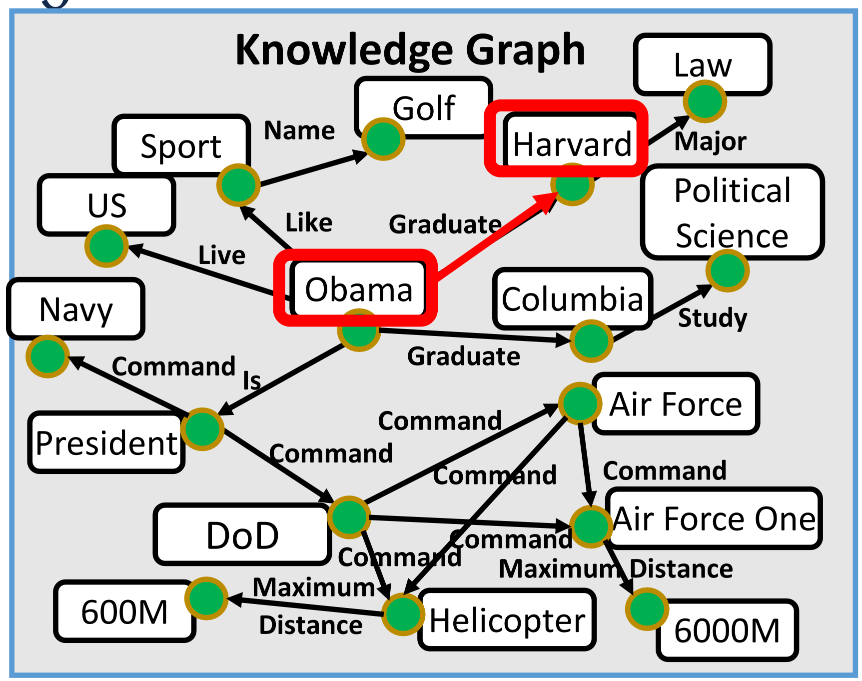
Outline

- Motivations
- Problem Definitions
- Key Ideas and Solutions
- Experiments and Prototype
- Conclusion

Problem Definition #1: Single claim fact checking

- **Goal:** Answering whether a claim is true or false

- **Input:**
  - A background knowledge graph $\mathcal{G}$
  - A claim as a triple $< s, p, o >$



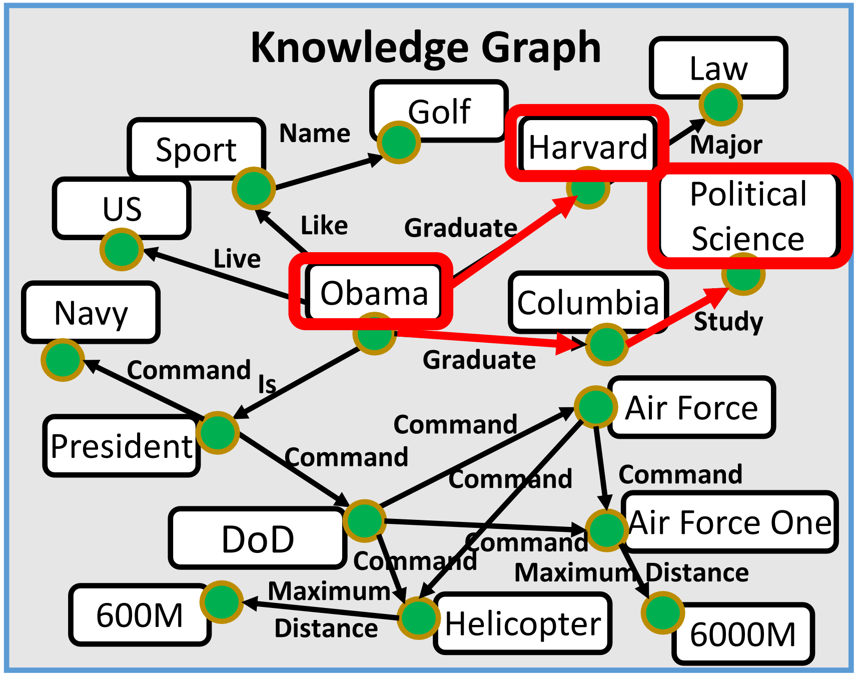
s and o are two nodes, p is a relationship.
e.g., <Barack_Obama, graduateFrom, Harvard>

- **Output:**
  - True or false

Problem Definition #2: Pair-wise fact checking

- **Goal:** Answering whether two separate claims are consistent
  - **Consistent:** Both claims are true at the same time
- **Input:**
  - (1) A knowledge graph $\mathcal{G}$
  - (2) A pair of claims which are denoted as: $<s_1, p_1, o_1> and <s_2, p_2, o_2>$

s_1, o_1, s_2, o_2 are nodes p_1, p_2 are relationships.
e.g., <Barack_Obama, majorIn, Political Science>
< Barack_Obama, graduatedFrom, Harvard>



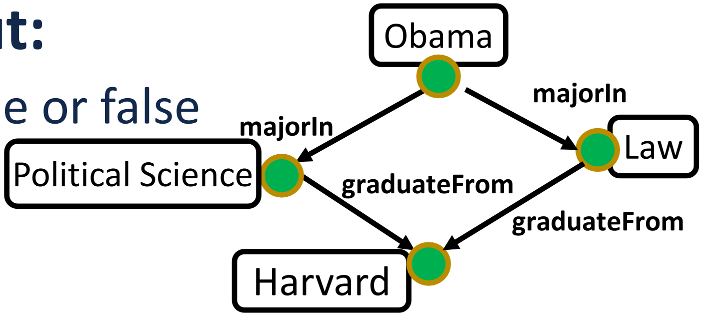
- **Output:**
  - The two facts are consistent or not

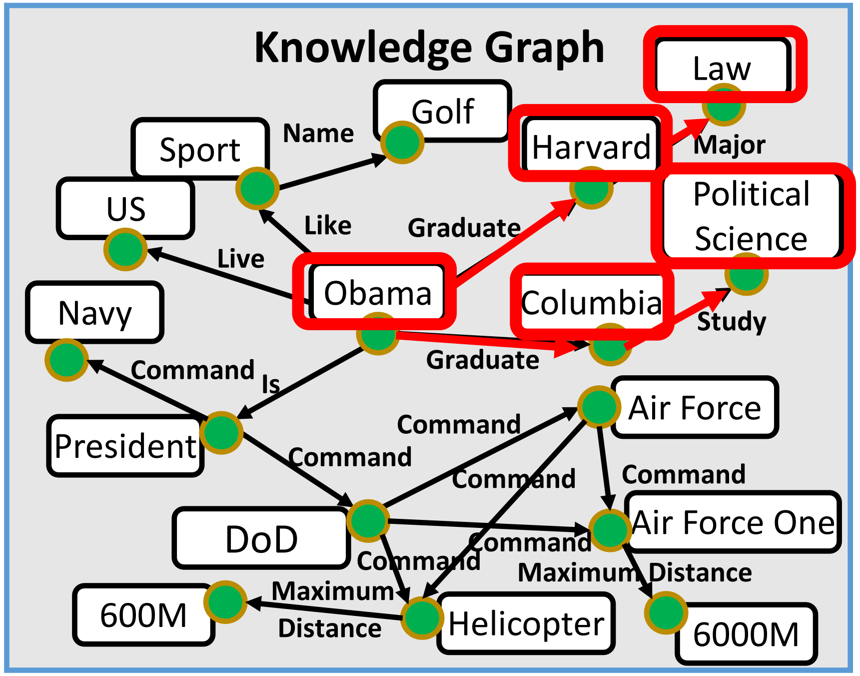
Problem Definition #3: Collective fact checking

- **Goal:** Answering whether a query graph is consistent. A query graph consists of a set of inter-connected edges/triples
  - **Consistent:** All claims are true at the same time
- **Input:**
  - (1) A knowledge graph $\mathcal{G}$
  - (2) A query graph $\mathcal{Q}$

Barack Obama finished both his bachelor's degree in political science and a master's degree in law at Harvard University

- **Output:**
  - True or false



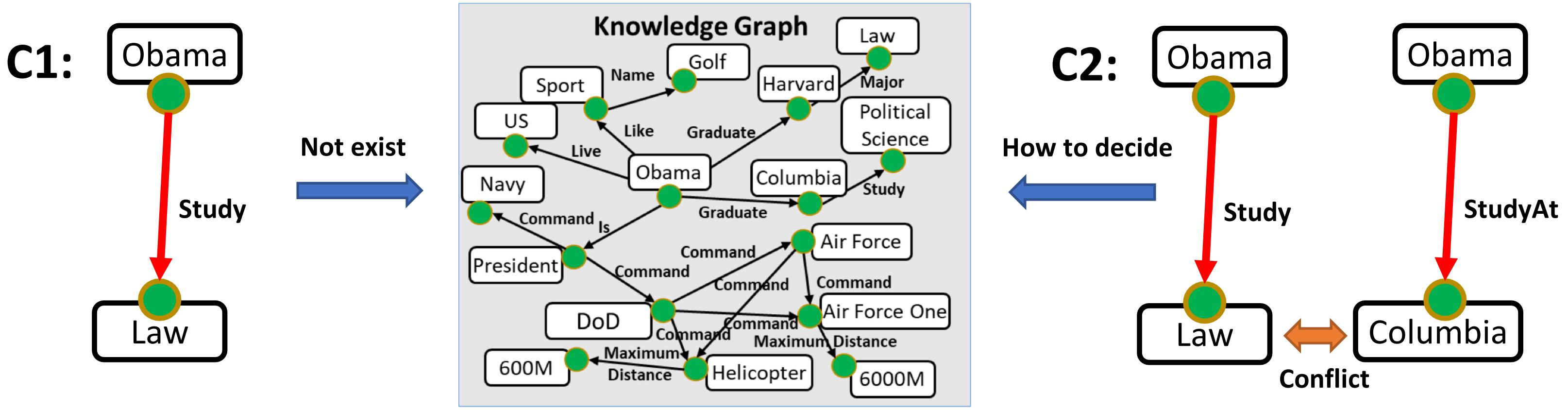


Outline

- ✓ Motivations
- ✓ Problem Definitions
- ➡ Key Ideas and Solutions
- Experiments and Prototype
- Conclusion

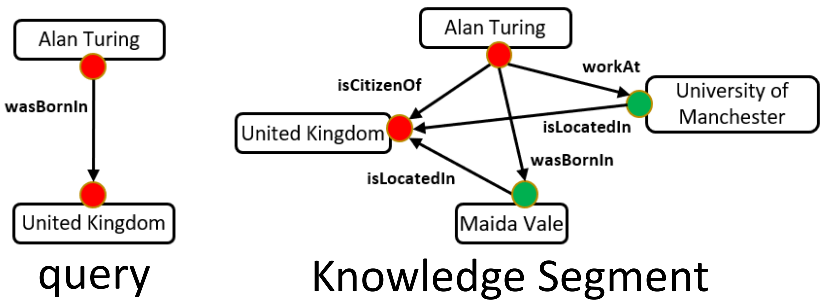
Challenges and Research Questions

- **Goal:** Detecting inconsistency inside a pair of claims or a query graph
- **Challenge #1**: How to express the claim?
  - Claim might not exist in the knowledge graph
  - Q1: How to utilize other related information in the knowledge graph
- **Challenge #2**: How to quantify inconsistency?
  - Too much irrelevant or noise information in the knowledge graph
  - Q2: How to decide the relevant/important information



Key Idea #1: Knowledge segment extraction

- **Challenge #1:** How to express the claim?
- Our solution:
  - Transform the knowledge graph into a weighted graph
  - Use K-simple shortest paths between two nodes to find knowledge segment
- Knowledge segment: (KS for short)
  - A connection subgraph of the knowledge graph
  - Describes the semantic context of a piece of given clue
    - i.e., a node, a triple or a query graph
- Advantages:
  - Useful when query claim does not directly exist in KG
  - Utilizing the 'background' or related entities



Key Idea #1: Knowledge segment extraction

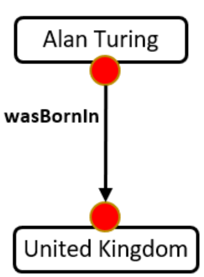
- **Details**: Knowledge segment extraction
  - Converting the knowledge graph into a weighted graph according to predicate-predicate similarity

$$
\begin{array}{r|c|c|c|c|c|}
\vdots & 3 & 0 & 4 & 0 & 1 \\
\text{predicate } \boldsymbol{p_i} & 0 & 3 & 1 & 0 & 2 \\
\vdots & 4 & 1 & 5 & 0 & 1 \\
\text{predicate } \boldsymbol{p_j} & 0 & 0 & 0 & 3 & 0 \\
\vdots & 1 & 2 & 1 & 0 & 0 \\
 & \cdots & \boldsymbol{p_i} & \cdots & \boldsymbol{p_j} & \cdots
\end{array}
$$

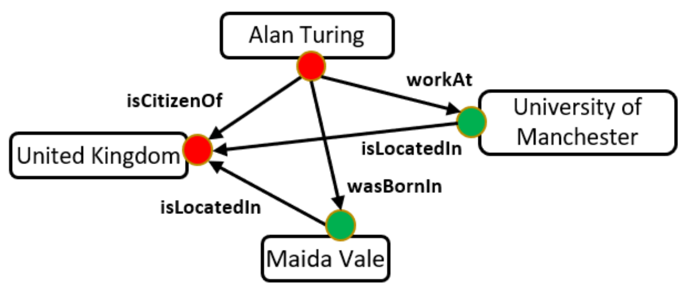
Co-occurrence matrix

Sim($\boldsymbol{p_i}$, $\boldsymbol{p_j}$)=cosine($\boldsymbol{row_i}$, $\boldsymbol{row_j}$)

  - Finding K-simple shortest paths between two nodes



query

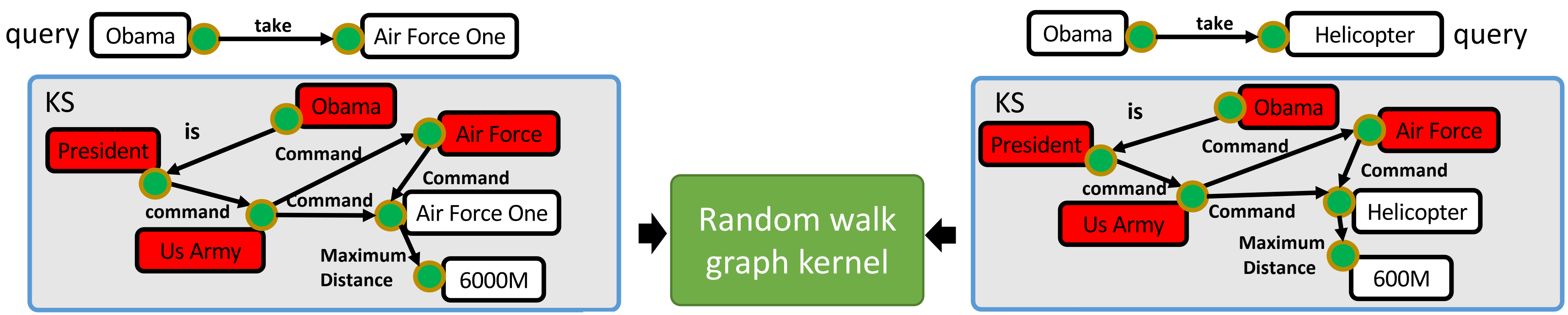


Knowledge Segment

- H. John and M. Matthew and S. Subhash. 2007. Finding the k Shortest Simple Paths: A New Algorithm and Its Implementation. (TALG ’07).

Key Idea #2: Inconsistency quantification

- **Challenge #2:** How to quantify inconsistency?
- Our solution for pair-wise fact checking
  - Influence function
    - Similarity between two knowledge segments
    - Find the nodes, edges, node attributes with the highest influence



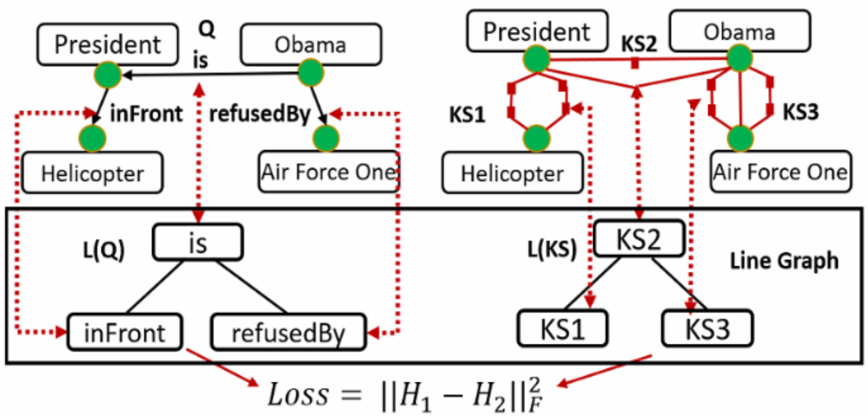
$$\mathrm{Sim}(KS_1, KS_2) = q_\times'(I - cN_\times A_\times)^{-1} N_\times P_\times$$

$$\frac{\partial Sim(KS_1, KS_2)}{\partial Node} \qquad \frac{\partial Sim(KS_1, KS_2)}{\partial Edge} \qquad \frac{\partial Sim(KS_1, KS_2)}{\partial Node\_Attibute}$$

- Q. Zhou, L. Li, N. Cao, L. Ying, and H. Tong. 2019. adversarial attacks on multi-network mining: problem definition and fast solutions(ICDM ’19)
- S. Vishwanathan and N. Schraudolph and R. Kondor and M. Borgwardt. 2010. Graph Kernels. (JMLR ’10).

Key Idea #2: Inconsistency quantification

- Our solution for collective fact checking
  - Transforming the query graph and knowledge segment graph into two line graphs
    - Given a graph G, its line graph L(G) is a graph such that
      - Each vertex of L(G) represents an edge of G
      - Two vertices of L(G) are adjacent if and only if their corresponding edges share a common endpoint ("are incident") in G

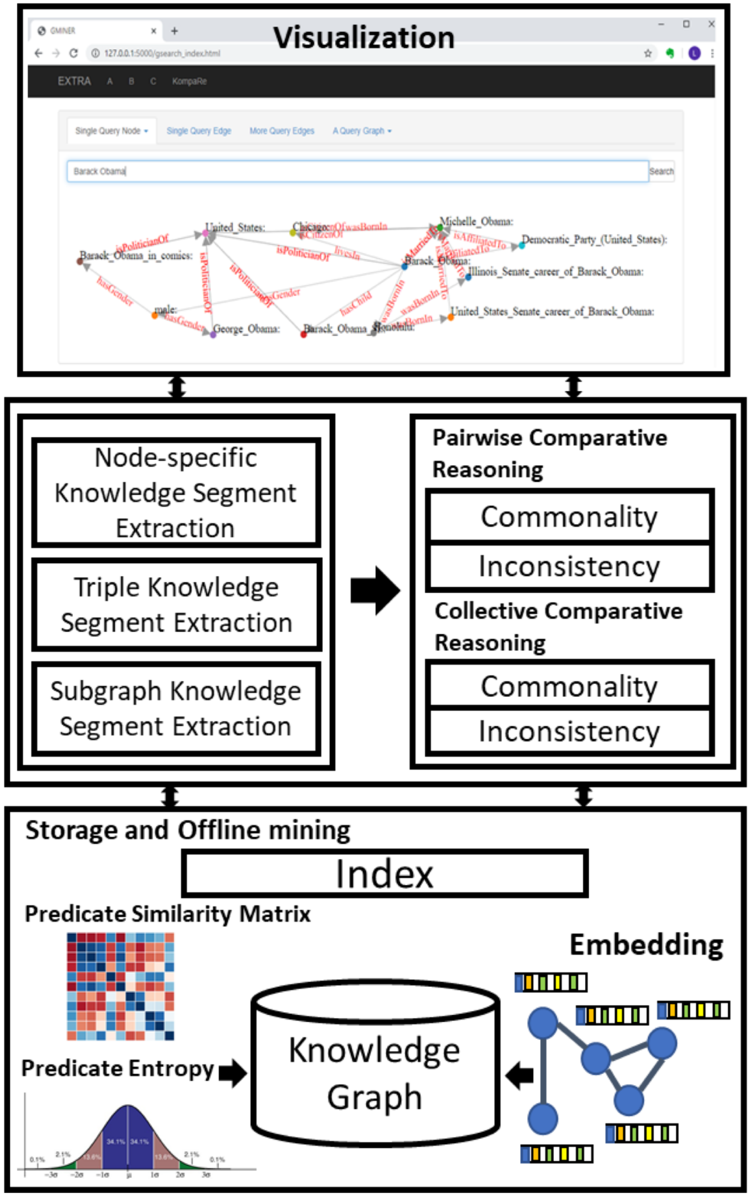


  - Finding the importance of nodes/edges/node attributes with influence function
    - Node/edge/node attribute influence $\frac{\partial Loss}{\partial Node}$ $\frac{\partial Loss}{\partial Edge}$ $\frac{\partial Loss}{\partial Node\_Attibute}$

Outline

- ✓ Motivations
- ✓ Problem Definitions
- ✓ Key Ideas and Solutions
- ➡ Experiments and Prototype
- Conclusion

System Architecture



User Interface

- Function selection
- Query input
- Visualization

Online Reasoning

- Single claim fact checking
- Pair-wise claim fact checking
- Collective fact checking

Offline Mining

- Predicate-predicate similarity calculation
- Predicate entropy calculation

Experiments

Datasets Summary:

- YAGO
  - 4,295,825 entities
  - 39 predicates
  - 12,430,705 triples
- Covid-19
  - 55,434 entities
  - 5,527,628 triples

- **Baseline Methods:**
  - TransE [Bordes et al. NeurIPS' 13]
  - Jaccard Similarity [Liben-Nowell et al. JASIST' 07]
  - Knowledge Linker [Ciampaglia et al. PLOS' 15]
  - KGMiner [Shi et al. KBS' 16]

- B. Antoine, U. Nicolas, G. Alberto, W Jason, and Y. Oksana. Translating Embeddings for Modeling Multi-relational Data. (NIPS '13).
- D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," (JASIST '2007).
- G. Ciampaglia, P. Shiralkar, and Rocha. Computational Fact Checking from Knowledge Networks.(PLOS '15).
- B. Shi and T. Weninger. Discriminative Predicate Path Mining for Fact Checking in Knowledge Graphs. (KBS'16).

Effectiveness Results

- Pair-wise comparative reasoning
  - 10 query sets. For each query set, each of them contains 300 query pairs
  - Accuracy = $\frac{N}{M}$, N is the number of queries correctly classified. M is the total number of queries

Dataset	# of queries	TransE	Jaccard	KL	KGMiner	Kompare
Family members positive	300	0.682	0.831	0.618	**0.983**	0.944
Family members negative	300	0.335	0.169	**1.000**	**1.000**	0.941
Graduated college positive	300	0.686	0.335	0.502	0.769	**0.794**
Graduated college negative	300	0.626	0.993	0.947	0.901	**0.994**
Live place positive	300	0.567	0.415	0.489	**0.834**	0.762
Live place negative	300	0.802	0.585	**0.907**	0.900	0.888
Birth place positive	300	0.590	0.435	0.537	0.698	**0.800**
Birth place negative	300	0.845	**1.000**	0.973	0.927	0.927
Work place positive	300	**0.751**	0.319	0.445	0.698	0.720
Work place negative	300	0.624	0.994	0.942	0.927	**0.995**
$mean \pm std$	-	0.651 ± 0.424	0.608 ± 0.302	0.736 ± 0.221	0.864 ± 0.105	$\mathbf{0.877 \pm 0.095}$

Accuracy of pair-wise comparative reasoning.

Effectiveness Results

- Collective comparative reasoning
  - YAGO: 6 query sets. Each query of collective comparative reasoning contains 3 edges

Dataset	# of queries	TransE	Jaccard	KL	KGMiner	Kompare
Birth place positive	300	0.542	0.418	0.389	0.678	**0.795**
Birth place negative	300	0.465	**0.996**	0.968	0.970	0.829
Live place positive	300	0.448	0.451	0.465	0.635	**0.989**
Live place negative	300	0.558	**1.000**	0.860	0.924	0.743
Graduated college positive	300	0.488	0.269	0.335	0.585	**0.963**
Graduated college negative	300	0.545	**0.996**	0.928	0.907	0.829
$mean \pm std$	-	0.508 ± **0.045**	0.688 ± 0.313	0.658 ± 0.265	0.783 ± 0.155	**0.858** ± 0.089

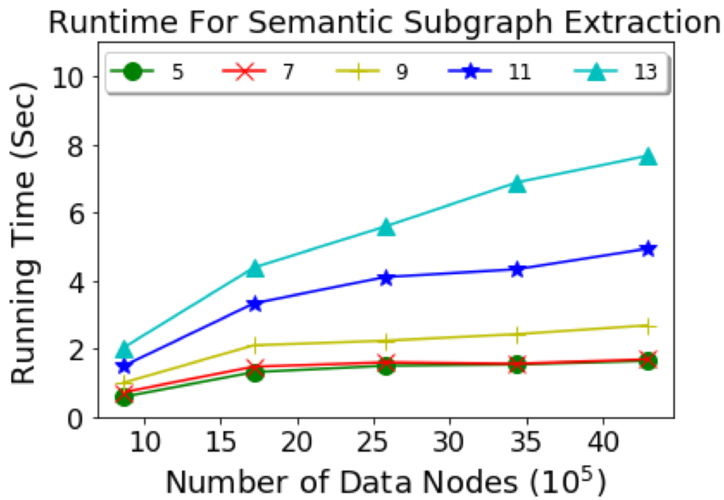
Accuracy of collective comparative reasoning.

  - Covid-19: Each query contains less than 8 nodes

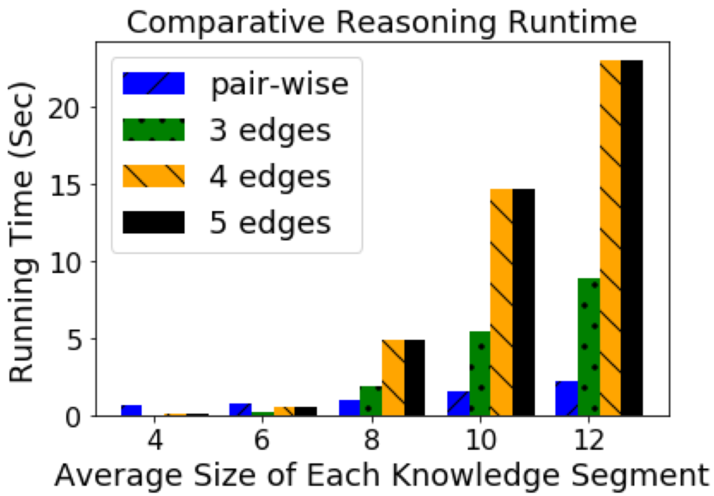
Dataset	# of queries	TransE	Jaccard	KL	KGMiner	Kompare
Positive	36	0.667	0.611	**1.000**	0.694	**1.000**
Negative	36	0.528	0.361	0.722	0.553	**0.863**
Average accuracy	-	0.598 ± 0.071	0.486 ± 0.126	0.861 ± 0.138	0.623 ± 0.071	**0.932 ± 0.063**

Accuracy of Covid-19 dataset.

Effectiveness vs. Running Time



Subgraph-specific KS extraction



Comparative reasoning

- The runtime for semantic subgraph extraction scales sub-linearly w.r.t. the number of nodes in the knowledge graph
- Average runtime of comparative reasoning is less than 8 seconds

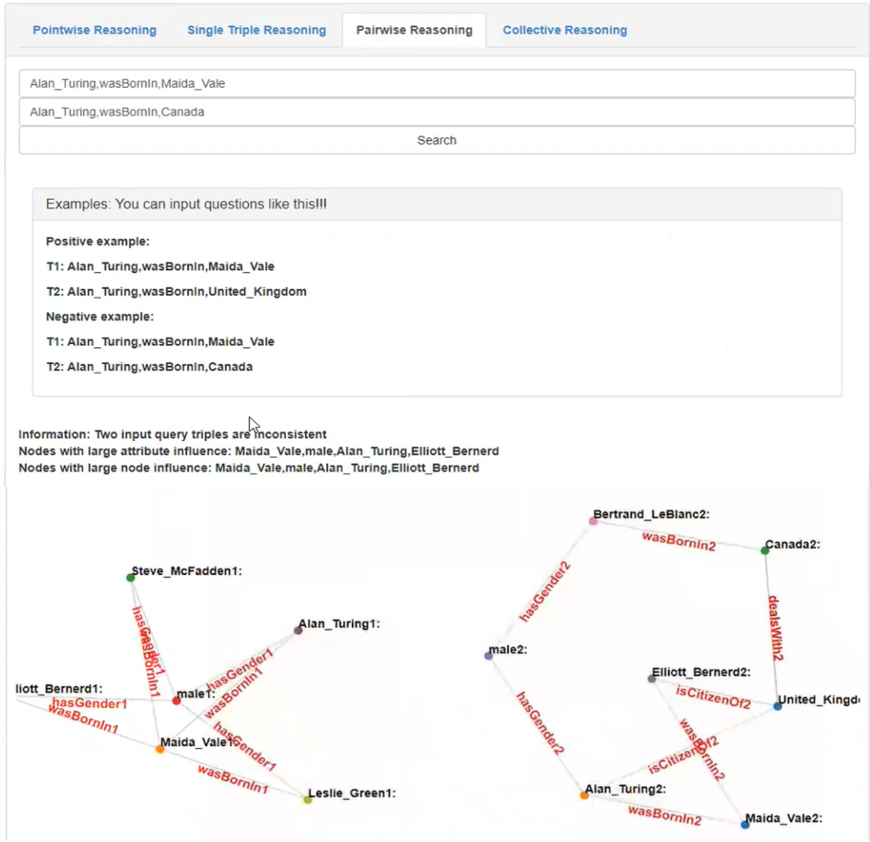
System Demonstration

- System demonstration: https://github.com/lihuiliullh/KompaRe

Input box

Fact checking result

Knowledge segment

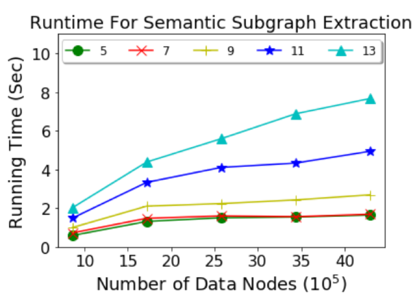


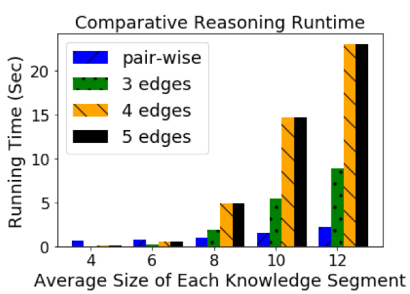
Outline

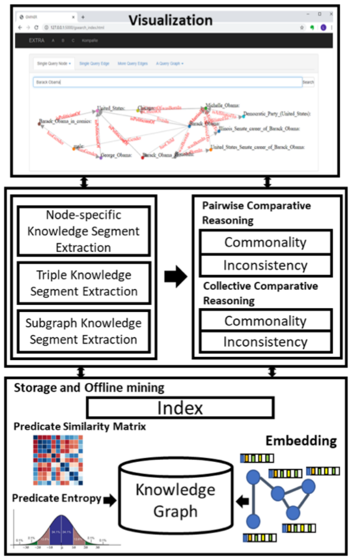
- ✓ Motivations
- ✓ Problem Definitions
- ✓ Key Ideas and Solutions
- ✓ Experiments and Prototype
- ➡ Conclusion

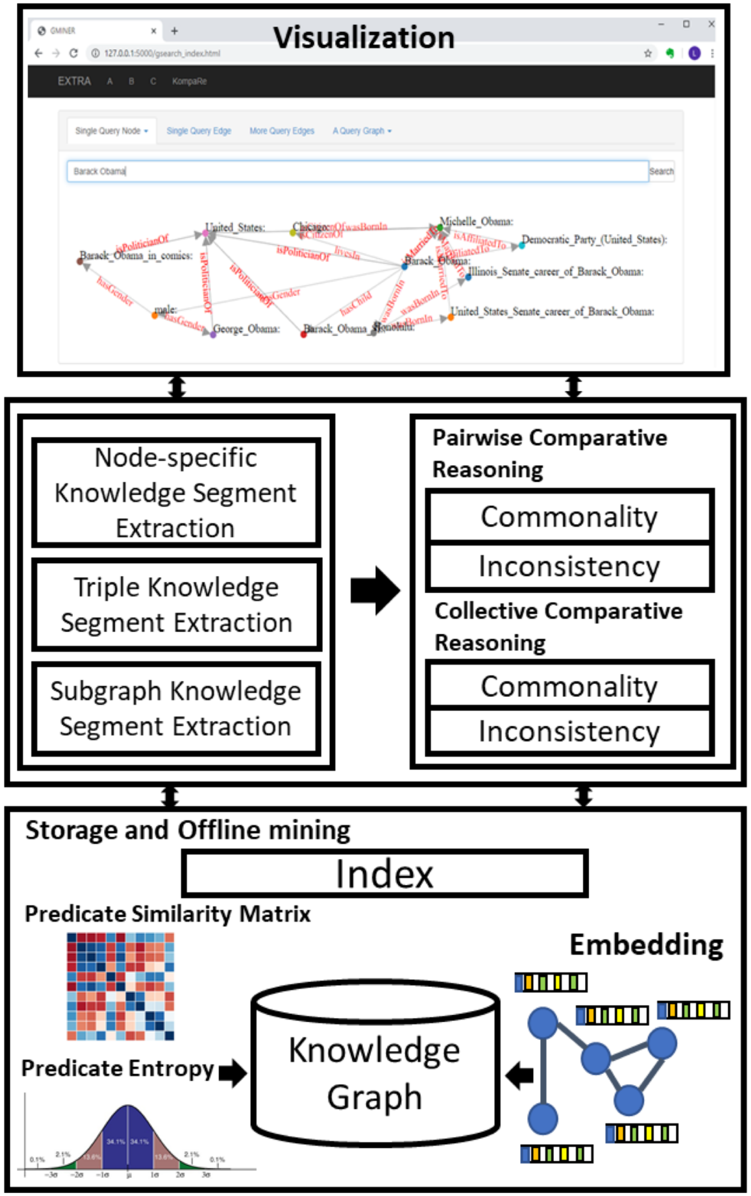
Conclusion

- Contribution:
  - ***We build a knowledge graph comparative reasoning system***
- Support functions:
  - Key function (1): single claim fact inconsistency checking
  - Key function (2): pair-wise fact inconsistency checking
  - Key function (3): collective fact inconsistency checking
- Results:
  - High accuracy of fact inconsistency checking
  - Fast running time on large knowledge graphs
  - Sublinear scalability









Thank you !

- KompaRe: A Knowledge Graph Comparative Reasoning System
- System demonstration: https://github.com/lihuiliullh/KompaRe
- Lihui Liu : lihuil2@illinois.edu