Improving Situational Awareness with Collective Artificial Intelligence over Knowledge Graphs

Meng Jiang

Department of Computer Science and Engineering
University of Notre Dame, Notre Dame, IN 46556, USA

ABSTRACT

Situational awareness (SA) was defined as perception of environmental elements within the situation, comprehension, and projection of future status. Here the perception is the representation of sensory information that includes multi-typed interacting objects forming "knowledge graphs." A typical problem in artificial intelligence (AI) research is to learn representations of objects that preserve structural information in knowledge graphs (KGs). Existing methods assume an AI agent has a complete knowledge graph, and any kind of prediction can be made accurately by a single AI. However, the real world needs multiple AI agents (e.g., warfighters, citizens) to collectively make a prediction. Each AI has a different, incomplete view of the knowledge graph with noise. In this work, I present a novel approach to improve representation learning and thus to improve SA with collective AI over KGs. I present the approach in four parts. First, I introduce knowledge graph and its nature of heterogeneity with multiple examples in the real world. Second, I discuss four ideas of making prediction with collective AI: prediction ensemble, data aggregation, representation aggregation, and joint representation learning. Third, I describe two state-of-the-art models to learn object representations from heterogeneous graphs: one is path-based embedding and the other is a graph neural network (GNN). Lastly, I present a new GNN framework that jointly learns object representations from multiple agents. Experimental results demonstrate that collective AI performs significantly better than individual AI. In future work, I discuss about federated learning that may improve security and privacy of the framework, which is quite necessary when any type of sharing (e.g., raw data, object representations, or learning process) is sensitive.

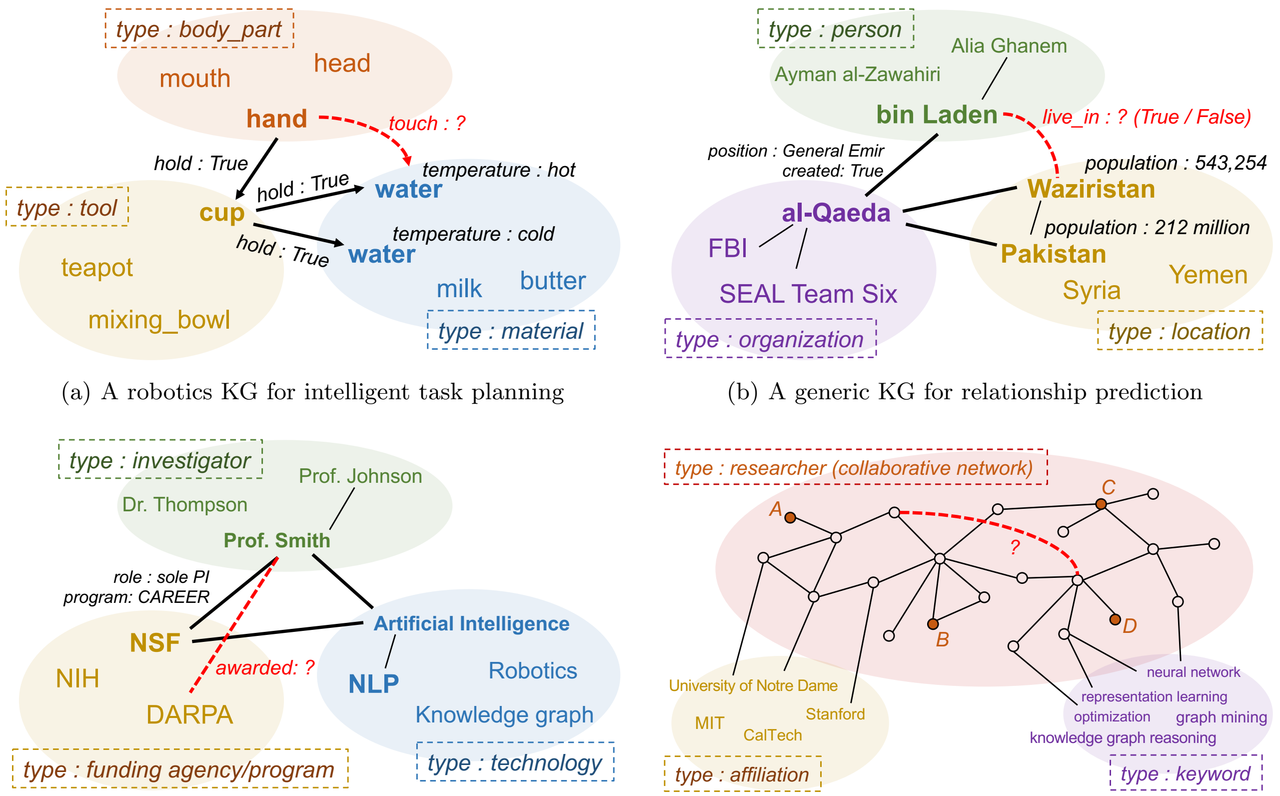
Keywords: Situational awareness, Collective intelligence, Knowledge graphs, Artificial intelligence

1. INTRODUCTION

Situational awareness (SA) was defined as a progression of three levels: (1) the perception of environments with respect to time or space within the situation, (2) the comprehension of their meaning, and (3) the projection of their future status.[1] It has been recognized as a critical foundation for successful decision-making in military operations.[2,3] Methods of gaining SA have two categories. One is military training that is to design training scenarios to increase knowledge and SA skills of military professionals. The other is crowdsensing that is to employ or use citizens as sensors to enhance SA analysis with crowd-generated inputs. Both types of methods collect information from individuals and rely on human intelligence to make perception, comprehension, and prediction with the information on critical matters. The form of the information should be able to describe objects in the environment and their complicated interactions, which has been introduced to the research field of data science as "knowledge graphs (KGs)." Most knowledge graphs are heterogeneous – consisted of multi-typed object nodes and multi-typed relationships between the nodes. For example, generic KGs have nodes of types *person*, *location*, and *organization* for representing and predicting relationships between them (e.g., "does bin Laden [*person*] live in Waziristan [*location*]"); robotics KGs have types such as *body_part*, *tool*, and *material* for intelligent task planning (e.g., "can we directly use hands [*body_parts*] to hold *boiling water* [*material*]").

In certain situations and to some degree, artificial intelligence (AI) outperforms human intelligence on discovering knowledge and making prediction with massive knowledge graph data from various sources, in terms of

Further author information: (Send correspondence to M.J.)
M.J.: E-mail: mjiang2@nd.edu, Telephone: 1 574 631 7454



(a) A robotics KG for intelligent task planning

(b) A generic KG for relationship prediction

(c) A research KG for measuring PI qualification

(d) A academic KG on recommending collaborators

Figure 1. Examples of knowledge graphs (KGs) in different domains. The KGs describe interactions (links and attributes) among multi-typed objects. AI algorithms are needed to learn the KG data to make accurate predictions.

accuracy and efficiency. As knowledge graphs are of heterogeneity and high dimensionality, a typical problem in AI research is to learn low-dimensional representations of objects in the heterogeneous graph. For a single node, the raw features (e.g., connecting or not connecting to other nodes) are high-dimensional (often at million level – equals to the graph size), drawing serious computational complexity to predictive models. Also, the raw features are generated from narrow local neighborhoods (i.e., one-hop) and thus too limited to describe the characteristics of the objects. The goal of representation learning is to preserve structural information in the graphs into latent representations (of 50, 100, or 300 dimensions).

Existing methods assume an AI agent has a complete knowledge graph; representations can be learned and any prediction can be made by a single AI. For example, Google uses the KG technology to enhance its search engine's results with massive amounts of data on people, places, things and facts gathered from a variety of sources.[*] Amazon is building Product (Knowledge) Graph for complex tasks of associating every product on Amazon with concrete and abstract concepts.[†] However, the real world needs multiple AI agents (e.g., warfighters, citizens) to collectively make accurate prediction. Each AI has a different, incomplete view of the knowledge graph with noise. Figure 1 gives examples in four different domains:

- *Robotics KG:* Service robots perform intelligent task planning collaboratively for serving humans. One robot may be trained to use cooking tools such as mixing bowls and pans; another may be trained to identify and use materials such as water, milk, and butter. Planning and verifying a plan of making a pancake need knowledge from a group of well-trained, specialized robots.

[*] Knowledge Graph: https://en.wikipedia.org/wiki/Knowledge_Graph
[†] Making search easier: https://blog.aboutamazon.com/innovation/making-search-easier

- *Generic KG:* Soldiers are threatened by terrorism (e.g., ex-soldiers were sentenced at terror attacks). It is important to make prediction on relationships between persons, locations, and organizations. However, individual prediction is unreliable due to limitation of isolated knowledge. Collective learning can significantly improve the effectiveness of relationship prediction.

- *Research KG:* Program directors, officers, and reviewers have limited knowledge when measure the qualification of Principal Investigators and evaluate the quality of grant proposals. There have been program committee meetings and panel discussions to integrate human intelligence to make accurate prediction.

- *Academic KG:* When one is looking for collaborators for a particular project/topic, he/she may have very limited information about the knowledge and skill set of researchers and their social networks, while there could be tens of thousands that may have required knowledge and skills.

The goal of this work is to harness any human (i.e., a warfighter or a citizen) with the power of AI on representation learning, reasoning, and inference, and to improve situational awareness with the "crowd of AI." A wide range of research has proved that collective intelligence, which makes collective efforts of several intelligent agents, holds the promise of a symbiotic intelligence that could be greater than the sum of the individual parts.[2,3] In this work, I present a novel approach to improve representation learning and thus to improve SA with collective AIs. I present the approach in four parts. First, I introduce knowledge graph and its nature of heterogeneity with multiple examples. Second, I discuss four ideas of making predictions with collective AI:

- *Prediction ensemble:* Each AI agent learns object representation and makes prediction from its own knowledge graph, then the collective AI aggregates the predictions using ensemble methods such as voting, weighted voting, bagging, bootstrapping, and boosting.[4]

- *Data aggregation:* Suppose every AI agent can share its knowledge graph. The idea is to aggregate information from various sources into a large-scale KG. The challenge is to identify the reliability of the sources and the trustworthiness of the objects and/or their relationships. With the aggregated KG, a single AI model, if capable for large scale, can learn representation and make prediction.[5,6]

- *Representation aggregation:* Each AI agent learns object representation but does not make prediction. The idea is to aggregate the representations using different kinds of strategies such as mean pooling, max pooling, or Long Short-Term Memory (LSTM) aggregator.[7] Then prediction can be made using a single predictive model on the aggregated representations.

- *Joint representation learning:* Due to some reason like data privacy, AI agents do not share their raw data (i.e., knowledge graphs), so I develop a learning model as collective AI that shares each AI's learning process, delivers jointly learned representation, and make prediction.

Third, I present two state-of-the-art models to learn object representations from heterogeneous graphs:

- *Path-based embedding:* If the object representations are expected to preserve path structures, e.g., the probability of two object nodes being connected through a particular type of path within a number of hops, the model determines a loss function and optimizes the representations by minimizing the loss.[8–10]

- *Graph neural networks:* The idea is to build neural architectures with observed graph data – nodes as neurons, links as neural connections, and neighbors as a former layer. Then node representations can be generated through the neural layers and optimized by pre-defined loss functions.[11,12]

Lastly, I present a new GNN framework that jointly learns object representations from multiple agents and knowledge graphs. Experimental results in a real-world dataset demonstrate that collective AI performs significantly better than individual AI. In future work, I will discuss about federated learning that may improve security and privacy of the framework, which is quite necessary when data sharing is sensitive.

2. APPROACH

In this section, I present the approach in four parts. First, I introduce knowledge graph and its nature of heterogeneity with multiple examples in the real world. Second, I discuss four ideas of making prediction with collective AI: prediction ensemble, data aggregation, representation aggregation, and joint representation learning. Third, I present two state-of-the-art models to learn object representations from heterogeneous graphs: one is path-based embedding and the other is a graph neural network. Lastly, I present the new GNN framework that jointly learns object representations from multiple agents.

2.1 Knowledge Graphs and Heterogeneity

A *knowledge graph* represents an abstraction of the real world in one's mind, focusing on the *objects* (e.g., concepts, entities) and the *relations* between the objects. It turns out that this level of abstraction has great power in not only representing the essential information about the real world but also providing a useful tool to make prediction and/or inference from it, by exploring the power of massive links.[13] Formally, a knowledge graph is defined as follows.

DEFINITION 2.1 (KNOWLEDGE GRAPH). *A knowledge graph is defined as a directed graph* $G = (\mathcal{O}, \mathcal{E} \subseteq \mathcal{O} \times \mathcal{O})$ *with (1) an* object type *mapping function* $\tau : \mathcal{O} \rightarrow \mathcal{T}$*, (2) an* object attribute value *mapping function:* $\theta : \mathcal{O} \times \mathcal{A} \rightarrow \mathcal{V}$*, and (3) a* relation attribute value *mapping function:* $\psi : \mathcal{E} \times \mathcal{R} \rightarrow \mathcal{V}$*, where*

- *Each object node* $o \in \mathcal{O}$ *belongs to one particular object type* $\tau(o) \in \mathcal{T}$*;*
- *Each link* $e = (o_i, o_j) \in \mathcal{E}$ *connects objects* $o_i \in \mathcal{O}$ *and* $o_j \in \mathcal{O}$*;*
- *The value of object o's attribute* $a \in \mathcal{A}$*,* $\theta(o, a)$*, belongs to the value space* $\mathcal{V}$ *(e.g., numerical values, binary values, temporal values, spatial values, categorical values);*
- *The value of link* $e = (o_i, o_j)$*'s relation attribute* $r \in \mathcal{R}$*,* $\psi(o_i, o_j, r)$*, belongs to the value space* $\mathcal{V}$*. A link must have at least one relation attribute/value.*

Heterogeneity: Clearly, heterogeneity is a natural property of knowledge graph. A knowledge graph naturally has multiple types of objects and multiple types of relations. When apply methods based on homogeneous graphs (of single-type objects and single-type relations) into heterogeneous graphs, we have to either project the heterogeneous graphs into homogeneous ones, or simply ignore the type information associated with nodes and links. Unfortunately, both ways will cause severe information loss. Therefore, heterogeneous graph mining methods have been proposed to utilize the semantic meaning of heterogeneous nodes and links. The novel concept, "meta path", is a path consisting of a sequence of relations defined between different object types (i.e., structural paths at the meta level). It provides guidance of search and mining of the heterogeneous graph and help analyze and understand the semantic meaning of the objects and relations in the graph.

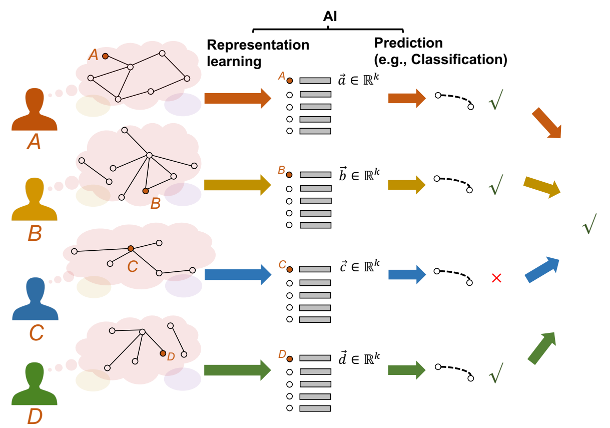
Examples: I present four heterogeneous knowledge graphs in different domains, as illustrated in Figure 1.

EXAMPLE 1 (ROBOTICS KG FOR INTELLIGENT TASK PLANNING). *To embed the facts in sentences "*... hand holds cup ... cup holds hot water ...*", a KG has:*

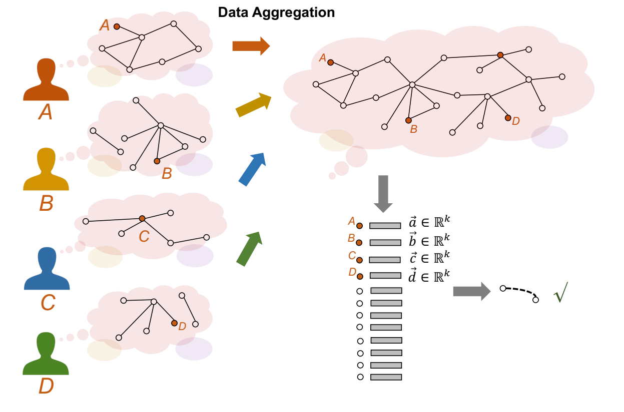
- *Nodes* hand, cup, water $\in \mathcal{O}$ *and links* (hand, cup), (cup, water) $\in \mathcal{E}$*;*
- *Node types* τ*(hand)* = body_part, τ*(cup)* = tool, τ*(water)* = material*;*
- *Node attributes:* θ*(water, temperature)* = hot*;*
- *Link (relation) attributes:* ψ*(hand, cup, hold)* = True *and* ψ*(cup, water, hold)* = True*.*

*The robotics KG can be used to learn object representations and answer questions like "*how to add boiling water?*" for the task of* making a tea*.*

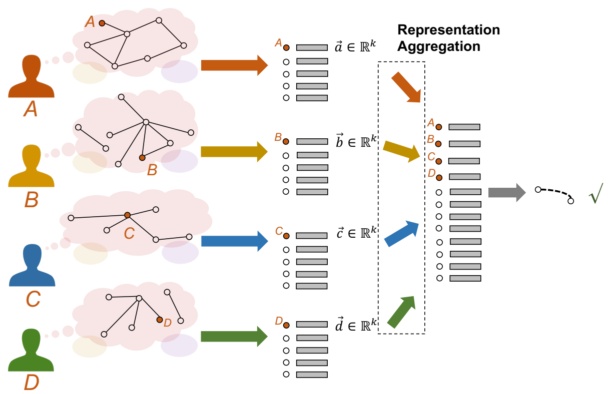
EXAMPLE 2 (GENERIC KG FOR RELATIONSHIP PREDICTION). *Suppose a warfighter wanted to predict whether bin_Laden lived in Waziristan. The generic KG has:*



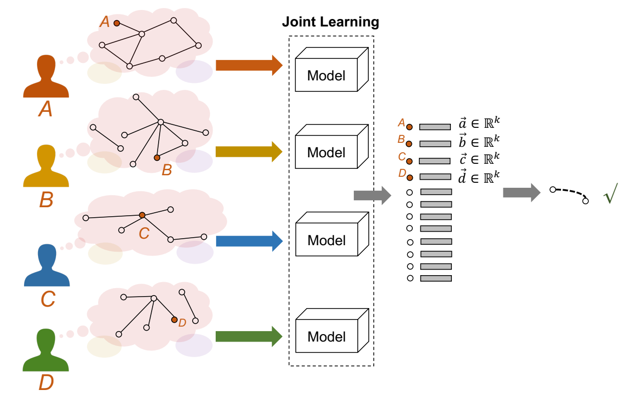
(a) Aggregating *prediction* from each AI based on its own data and learned representations



(b) Integrating *data* from all the AIs as a large, integrated KG to learn representations and make prediction



(c) Aggregating *learned representations* from each AI to make predictions without merging data



(d) Learning *learned representations* jointly from all the AIs to make prediction

Figure 2. We present four ideas to do *representation learning* and *prediction* with collective AIs over KGs. Each idea has its pros and cons in terms of accuracy, privacy, security, and efficiency.

- *Nodes* bin_Laden, al-Qaeda, Waziristan $\in \mathcal{O}$;
- *Links* (bin_Laden, al-Qaeda), (al-Qaeda, Waziristan) $\in \mathcal{E}$;
- *Node types* τ*(bin_Laden)* = person, τ*(al-Qaeda)* = organization, τ*(Waziristan)* = location;
- *Node attributes:* θ*(Waziristan, population)* = 543,254;
- *Link attributes:* ψ*(bin_Laden, al-Qaeda, created)* = True *and* ψ*(bin_Laden, al-Qaeda, role)* = General_Emir.

Example 3 (Research KG for measuring PI qualification). *Program officers and reviewers are willing to measure the qualification of PI(s) based on their research grant proposals. The research KG has:*

- *Nodes* Prof._Smith, NSF, AI $\in \mathcal{O}$ *and links* (Prof._Smith, NSF), (NSF, AI) $\in \mathcal{E}$;
- *Node types* τ*(Prof._Smith)* = investigator, τ*(NSF)* = funding_agency, τ*(AI)* = technology;
- *Node attributes:* θ*(Prof._Smith, title)* = Chair Professor;
- *Link attributes:* ψ*(Prof._Smith, NSF, role)* = sole PI *and* ψ*(Prof._Smith, NSF, program)* = CAREER.

Example 4 (Academic KG for recommending collaborators). *The academic KG, mainly consisted of co-authorship social networks, has:*

- *Nodes* Alice, Bob, Notre_Dame, GNN $\in \mathcal{O}$ *and links* (Alice, Notre_Dame), (Bob, GNN) $\in \mathcal{E}$*;*
- *Node types* τ*(Alice)* = researcher*,* τ*(Notre_Dame)* = affiliation*,* τ*(GNN)* = keyword*;*
- *Node attributes:* θ*(Alice, status)* = graduate_student*;*
- *Link attributes:* ψ*(Alice, Notre_Dame, program)* = CS PhD *and* ψ*(Bob, GNN, published_times)* = 7 *(i.e., Bob published 7 papers that have keyword "GNN").*

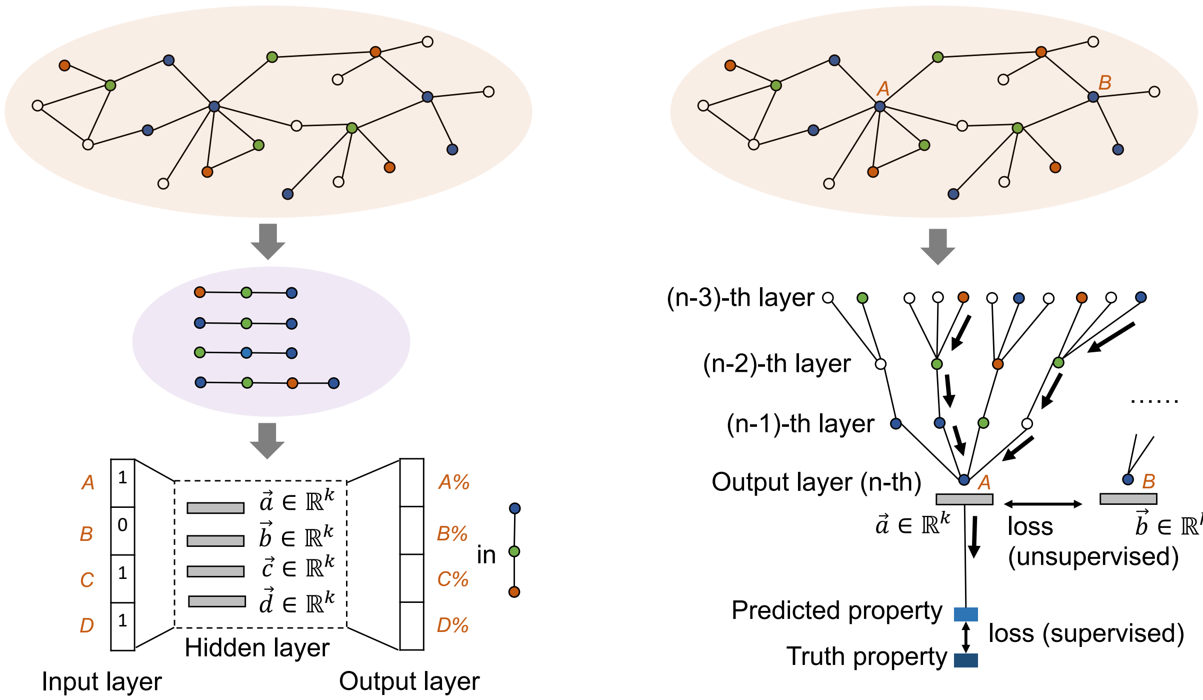
2.2 Collective AI over KGs

Collective efforts are expected from several intelligent agents. Figure 2 present four ideas to explore:

- *Prediction ensemble:* The above two directions are to aggregate either data or knowledge and then to make a prediction. Actually, the intelligent agents can make prediction locally by themselves; in the battlespace, this is more common: it is easier and faster for warfighters (e.g., aircraft, soldier) to transmit their predictions than data or knowledge. Bagging models (e.g., voting classifiers) can be used to integrate the predictions for a more robust, successful decision. Boosting models (e.g., AdaBoost, XGBoost) can further improve the integration by estimating the reliability of the predictions.
- *Data aggregation:* A "universal" knowledge graph could be generated by integrating multiple graphs from individuals. To address the conflicts during the integration, truth finding techniques must be applied. The assumptions are (a) more reliable sources were more likely to generate trustworthy attribute values and relational links; and (b) if more trustworthy facts were generated, the source would more likely be a reliable one. Learning on the universal knowledge graph will generate more comprehensive node embeddings than learning on any single graph. However, the universal graph could be too big to operate (e.g., load into memory) with limited computational resources.
- *Representation aggregation:* Basically, the idea is to jointly learn node embeddings across multiple knowledge graphs because the graphs may share entity nodes. Each graph makes an optimization term in the objective function, however, the importance of graph, as weight of the term, needs to be estimated if the weights are uniformly distributed, the learning process will be biased to bigger graphs, not the ones of high quality and less noise.
- *Joint representation learning:* Instead of sharing raw data (i.e., knowledge graphs), a joint learning model shares each AI's learning process, delivers jointly learned representation, and make prediction. Section 2.3 will introduce two representation learning models. Section 2.4 will present the new approach for joint representation learning over multiple knowledge graphs.

2.3 AI Models for KG Representation Learning

A computer system captures data/information from an individual (a warfighter, a citizen, or an artificial intelligent agent) about his/her/its perception of the situation, specifically, the entities in the situation.[14] Such information includes (1) entity's attributes, such as weight carrying capacity of an aircraft and maximum range of a missile, and (2) relations between entities, such as distance between two points of interests and social/family relationship of a terrorist. Naturally, the information can be represented as a "knowledge graph": The graph is consisted of nodes of entities, nodes attributes, and relational links between nodes. If data were in text form, natural language processing (NLP) techniques can be applied to find entity names, attributes, and relations. Traditional SA methods would represent entities or events with textual features (e.g., bag-of-words, topics) and compute relevance (with Boolean query or cosine similarity) to describe the situation.[15] The knowledge graph would bring tremendous information if properly utilized; however, it contains various nodes and relations and is of large scale and high dimensionality. Representation learning models are desired to transform the large



(a) metapath2vec

(b) HeteroGNN

Figure 3. Two models for transforming large high-dimensional data (i.e., knowledge graphs) into a form (i.e., low-dimensional latent features) that the system can utilize to solve complex tasks.

high-dimensional data into a form that the system can utilize to solve complex tasks. We present two types of representation learning models as below:

Meta path-based embedding learning: (See Figure 3(a)) The embedding learning methods expect object representations (or called "embeddings") to be able to preserve path structures. For example, given the embeddings, a simple regression model would be able to predict the probability of two object nodes being connected through a particular type of meta path within a number of hops, the model determines a loss function and optimizes the representations by minimizing the loss.

Heterogeneous graph neural networks (HeteroGNN): (See Figure 3(b)) HeteroGNN learns low-dimensional features of each entity node by preserving important structures of the graph. The model can aggregate attributed and relational information from local neighborhood and apply non-linear transformations through multiple layers to fit the complicated real graph data. Two popular types of loss functions are (1) random walk-based unsupervised loss and (2) cross-entropy supervisory loss. The node embeddings make comprehension over knowledge graphs, which can be used for relevancy computation, similarity search, anomaly detection, prediction, recommendation, and decision making. Results from prior research have proved that GNNs generate more effective features than raw textual features such as term frequent/inverse document frequency (TF/IDF).

2.4 A Joint Learning Model over Collective KGs

Problem 1. *Given m knowledge graphs $\{\mathcal{G}_i = (\mathcal{O}_i, \mathcal{E}_i)\}_{i=1}^{m}$, where $\mathcal{O}_i \subset \mathcal{O}$ is the set of object nodes in the i-th agent's KG and $\mathcal{E}_i \subset \mathcal{E}$ is the set of links ($\mathcal{O}$ is the set of nodes in all KGs and $\mathcal{E}$ is the set of links in all KGs), learn the representations of each object node $f(v) : \mathcal{O} \rightarrow \mathbb{R}^d$, where d is the number of dimensions of the node embeddings. The node embeddings can be used for tasks of prediction, inference, and reasoning.*

We design a novel graph neural network model to learn node representations from heterogeneous, attributed knowledge graphs collectively. The model has two parts: One is an algorithm for node embedding generation

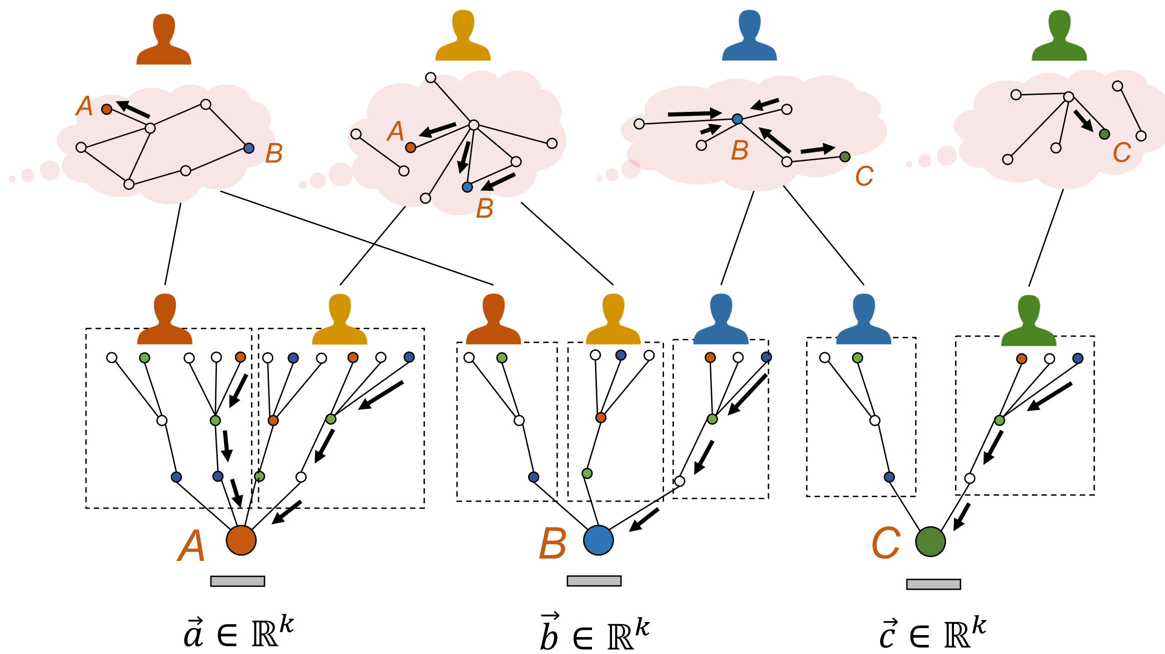


Figure 4. The proposed approach: A joint learning model over collective knowledge graphs.

given raw data and model parameters; the other is loss function(s) for training the model parameters to perform accurately on some supervisory task(s) that need information being preserved as expected. The loss functions can be (1) cross-entropy supervisory loss when node labels at the global level are available, and (2) unsupervised pair-wise loss that uses random walkers to generate similarity to optimize the embedding similarities.

Node embedding generation: (See Figure 4) First, we use matrix $\mathbf{M}^{(i)} \in \mathbb{R}^{k \times d}$ to transform from node's k-dimensional raw features $\mathbf{g}^{(i)}(v) \in \mathbb{R}^d$ (from function θ) to the initial latent embeddings $\mathbf{v}_0^{(i)} \in \mathbb{R}^k$ on the i-th knowledge graph, and aggregate into a single latent embedding $\mathbf{v}_0 \in \mathbb{R}^k$:

$$\mathbf{v}_0 = \sigma\left(\text{AGG}_{v \in \mathcal{O}_i, i=1\ldots m} \mathbf{v}_0^{(i)}\right) \tag{1}$$

$$\mathbf{v}_0^{(i)} = \mathbf{M}^{(i)} \cdot \mathbf{g}^{(i)}(v), \tag{2}$$

where $\sigma(\cdot)$ is an activation function, which can be sigmoid, hyperbolic tangent, ReLU, etc.

Second, for the j-th layer of the neural network ($j \in \{1 \ldots n\}$, where n is the number of layers), which means in the j-th iteration of the embedding generation algorithm, we generate the embedding vector $\mathbf{v}_j \in \mathbb{R}^k$ for node v in graphs $\{G^{(i)}\}_{i=1}^m$ as follows:

$$\mathbf{v}_j = \sigma\left(\beta \cdot \text{AGG}_{v \in \mathcal{O}_i} \mathbf{v}_j^{(i)} + (1-\beta) \cdot \mathbf{A}_j \cdot \mathbf{v}_{j-1}\right) \tag{3}$$

$$\mathbf{v}_j^{(i)} = \alpha \cdot \mathbf{W}_j^{(i)} \cdot \text{AGG}_{u \in \mathcal{N}^{(i)}(v)} \mathbf{u}_{j-1}^{(i)} + (1-\alpha) \cdot \mathbf{B}_j^{(i)} \cdot \mathbf{v}_{j-1}^{(i)}, \tag{4}$$

where

- $\mathbf{v}_j^{(i)} \in \mathbb{R}^k$ is the embedding of node v at the j-th layer from the i-th knowledge graph G_i; $\mathbf{v}_{j-1}^{(i)} \in \mathbb{R}^k$ is the corresponding embedding at the $(j-1)$-th layer;
- $\mathbf{v}_{j-1} \in \mathbb{R}^k$ is the aggregated embedding of node v at the $(j-1)$-th iteration;
- $\mathbf{u}_{j-1}^{(i)} \in \mathbb{R}^k$ is the embedding of node u as a neighboring node of v in graph G_i at the $(j-1)$-th iteration;

- $\mathbf{A}_j \in \mathbb{R}^{k \times k}$ is the transformation matrix from the aggregated embedding of a node on the $(j-1)$-th layer to the j-th layer;
- $\mathbf{W}_j^{(i)} \in \mathbb{R}^{k \times k}$ is the transformation matrix from the aggregated information of neighboring nodes on the i-th knowledge graph G_i from the $(j-1)$-th layer to the j-th layer;
- $\mathbf{B}_j^{(i)} \in \mathbb{R}^{k \times k}$ is the transformation matrix from the embedding of a node on the i-th knowledge graph G_i from the $(j-1)$-th layer to the j-th layer;
- α is a hyperparameter weighting the aggregation of neighboring nodes;
- β is a hyperparameter weighting the node embedding on specific knowledge graphs;
- AGG is an aggregation function, which can be mean pooling, max pooling, or LSTM aggregator.

The final embedding vectors are $\{\hat{\mathbf{v}} \equiv \mathbf{v}_n\}|_{v \in \mathcal{O}}$, generated with model parameters including $\mathbf{M}^i|_{i=1}^m$, $\mathbf{A}_j|_{j=1}^n$, $\mathbf{W}_j^{(i)}|_{j=1}^n|_{i=1}^m$, and $\mathbf{B}_j^{(i)}|_{j=1}^n|_{i=1}^m$ from multiple heterogeneous knowledge graphs $G^{(i)}|_{i=1}^m$. Each final embedding vector $\hat{\mathbf{v}}$ preserves structural and attributed information of node v in the graphs.

3. PRELIMINARY EXPERIMENTS

Data description: Early definitions identify SA as the answer to three fundamental battlefield questions:[2,3] (1) Where am I? (2) Where are my friends? (3) Where is the enemy? We translate these questions into academic networks as three predictive tasks:

Suppose each researcher has a "knowledge graph" of his/her understanding of the academic network, which is absolutely partial of the entire academia. The graph has nodes of researchers, affiliations, their publications, the publications venues, and keywords, and also node attributes such as publications number of citation and researchers H-index. A researcher may be interested in: (1) What could be my next affiliation/job? (2) Who could be my new collaborators? (3) Who are my competitors that publish the same/similar topics in the same/similar venues but never collaborated?

Suppose a researcher can develop an AI model to learn from the partial knowledge and make predictions. We investigate the effectiveness of the proposed collective AI method. We use open academia data: 409,504 authors, 28,670 keywords, 13,081 venues, and 1,133,443 papers. The partial knowledge graph for each author was constructed based on one-hop association from the author (e.g., co-author, authored paper).[9,10] We have experimental results on the second question predicting new collaborators.

Results: We aim at predicting the next collaborator. It is a very challenging problem because there are almost half a million authors in the dataset, which means random guess can only have an accuracy of 0.00002%. We use Recall@10 (i.e., if the truth is covered in the top 10 predictions) to evaluate the performance. Note that the maximum can only be 10% because there is only one truth. We develop collective AI using the second type of integration, i.e., knowledge representation integration. Here are the results: (Recall@10 in percentage)

- Single AI with raw features: 0.24;
- Single AI with GNNs: 0.27;
- Collective AI with raw features: 2.18;
- Collective AI with GNNs: 5.41. (The best: significantly better than others)

Conclusions based on the experimental results are (1) GNN model performs better than using raw features; (2) Collective AI can significantly improve the performance over single AI.

More experiments will be conducted to answer the first and third questions. We leave them as future work.

4. RELATED WORK

Graph data learning has been a popular research topic in the field of knowledge discovery, machine learning, and artificial intelligence. Handling the data heterogeneity is always one of the biggest challenges in the related research. Jiang *et al.* proposed a scalable algorithm based on minimum description length (MDL) principles to generate graph summaries from large heterogeneous graphs as a compression problem.[5] Gui *et al.* proposed a hyper-edge embedding method to learn representations from multi-typed and multi-dimensional graphs.[17] Wang *et al.* proposed an algorithm to learn representations of objects from a great number of sets of objects by preserving the set structures. They further proposed a method to measure the complementarity among the objects in the set, which becomes very effective for making decisions on applications such as recommender systems.[10] Recently, Yu *et al.* discovered the usefulness of knowledge graph learning for identifying intentions of human behaviors.[16]

5. CONCLUSIONS AND FUTURE WORK

In this work, I focused on learning representations of objects that preserve structural information in knowledge graphs. Existing methods assumed an AI agent had a complete knowledge graph, and any kind of prediction could be made accurately by a single AI. However, the real world needs multiple AI agents to collectively make a prediction. Each AI has a different, incomplete view of the knowledge graph with noise. I presented a novel approach to improve representation learning and thus to improve SA with collective AI over KGs. First, I introduced knowledge graph and its nature of heterogeneity with multiple examples in the real world. Second, I discussed four ideas of making prediction with collective AI: prediction ensemble, data aggregation, representation aggregation, and joint representation learning. Third, I described two state-of-the-art models to learn object representations from heterogeneous graphs, path-based embedding and graph neural network. Lastly, I presented a new GNN framework that jointly learns object representations from multiple agents. Experimental results demonstrated that collective AI performs significantly better than individual AI.

Privacy and security have become critical concerns in recent years, particularly as companies and organizations increasingly collect detailed information about their users. This information can enable machine learning methods that make better predictions. However, it also has the potential to allow for misuse, especially when private data about individuals is involved. Recent research shows that privacy and utility do not necessarily need to be at odds, but can be addressed by careful design and analysis.[18–25] An approach that has the potential to address a number of problems in this space is Federated Learning (FL). FL is an ML setting where many AI agents collaboratively train a model under the orchestration of a central server, while keeping the training data decentralized. Organizations and mobile devices have access to increasing amounts of sensitive data, with scrutiny of ML privacy and data handling practices increasing correspondingly. These trends have produced significant interest in FL, since it provides a viable path to state-of-the-art ML (e.g., collective AI over knowledge graphs) without the need for the centralized collection of training data and the risks and responsibilities that come with such centralization.

REFERENCES

[1] Endsley, M. R., “Toward a theory of situation awareness in dynamic systems,” in [*Situational awareness*], 9–42, Routledge (2017).

[2] Mittrick, M., Richardson, J., Asher, D. E., James, A., and Hanratty, T., “A visualization technique to improve situational awareness,” in [*Next-Generation Analyst VI*], **10653**, 106530L, International Society for Optics and Photonics (2018).

[3] Mittrick, M., Richardson, J., Dennison Jr, M., Trout, T., Heilman, E., and Hanratty, T., “Investigating immersive collective intelligence,” in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*], **11006**, 110060U, International Society for Optics and Photonics (2019).

[4] Yu, W., Li, Z., Zeng, Q., and Jiang, M., “Tablepedia: Automating pdf table reading in an experimental evidence exploration and analytic system,” in [*The World Wide Web Conference*], 3615–3619 (2019).

[5] Jiang, M., Faloutsos, C., and Han, J., “Catchtartan: Representing and summarizing dynamic multicontextual behaviors,” in [*Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*], 945–954 (2016).

[6] Wang, X., Zhang, H., Li, Q., Shi, Y., and Jiang, M., "A novel unsupervised approach for precise temporal slot filling from incomplete and noisy temporal contexts," in [*The World Wide Web Conference*], 3328–3334 (2019).

[7] Hamilton, W., Ying, Z., and Leskovec, J., "Inductive representation learning on large graphs," in [*Advances in neural information processing systems*], 1024–1034 (2017).

[8] Dong, Y., Chawla, N. V., and Swami, A., "metapath2vec: Scalable representation learning for heterogeneous networks," in [*Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*], 135–144 (2017).

[9] Wang, D., Jiang, M., Zeng, Q., Eberhart, Z., and Chawla, N. V., "Multi-type itemset embedding for learning behavior success," in [*Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*], 2397–2406 (2018).

[10] Wang, D., Jiang, T., Chawla, N. V., and Jiang, M., "Tube: Embedding behavior outcomes for predicting success," in [*Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*], 1682–1690 (2019).

[11] Zhang, C., Song, D., Huang, C., Swami, A., and Chawla, N. V., "Heterogeneous graph neural network," in [*Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*], 793–803 (2019).

[12] Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., and Yu, P. S., "Heterogeneous graph attention network," in [*The World Wide Web Conference*], 2022–2032 (2019).

[13] Sun, Y. and Han, J., "Mining heterogeneous information networks: a structural analysis approach," *Acm Sigkdd Explorations Newsletter* **14**(2), 20–28 (2013).

[14] Ganger, R., Coles, J., Ekstrum, J., Hanratty, T., Heilman, E., Boslaugh, J., and Kendrick, Z., "Application of data science within the army intelligence warfighting function: problem summary and key findings," in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*], **11006**, 110060N, International Society for Optics and Photonics (2019).

[15] Richardson, J., James, A., Mittrick, M., Hanratty, T., and Heilman, E., "Methodology for using the operations order to filter intelligence reports," in [*Next-Generation Analyst VI*], **10653**, 1065304, International Society for Optics and Photonics (2018).

[16] Yu, W., Yu, M., Zhao, T., and Jiang, M., "Identifying referential intention with heterogeneous contexts," in [*The Web Conference*], (2020).

[17] Gui, H., Liu, J., Tao, F., Jiang, M., Norick, B., Kaplan, L., and Han, J., "Embedding learning with events in heterogeneous information networks," *IEEE transactions on knowledge and data engineering* **29**(11), 2428–2441 (2017).

[18] Konečnỳ, J., McMahan, B., and Ramage, D., "Federated optimization: Distributed optimization beyond the datacenter," *arXiv preprint arXiv:1511.03575* (2015).

[19] Zhu, H. and Jin, Y., "Multi-objective evolutionary federated learning," *IEEE transactions on neural networks and learning systems* (2019).

[20] Lalitha, A., Kilinc, O. C., Javidi, T., and Koushanfar, F., "Peer-to-peer federated learning on graphs," *arXiv preprint arXiv:1901.11173* (2019).

[21] Yang, Q., Liu, Y., Chen, T., and Tong, Y., "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)* **10**(2), 1–19 (2019).

[22] Chai, D., Wang, L., Chen, K., and Yang, Q., "Secure federated matrix factorization," *arXiv preprint arXiv:1906.05108* (2019).

[23] Kim, Y., Sun, J., Yu, H., and Jiang, X., "Federated tensor factorization for computational phenotyping," in [*Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*], 887–895 (2017).

[24] Li, T., Sahu, A. K., Talwalkar, A., and Smith, V., "Federated learning: Challenges, methods, and future directions," *arXiv preprint arXiv:1908.07873* (2019).

[25] Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y.-C., Yang, Q., Niyato, D., and Miao, C., "Federated learning in mobile edge networks: A comprehensive survey," *arXiv preprint arXiv:1909.11875* (2019).