Knowledge Graph

Comprehensive Introduction

Anahita Pakiman

Fraunhofer SCAI

02/01/21

Introduction

Phrase “Knowledge Graph” born in 1972
A graph of data intended to accumulate and convey knowledge of the real world
Knowledge like

- Simple statement: Santiago is the capital of Chile → edge
- Quantified statement: All capitals are cities → rules/ontology → deductive knowledge

Ever-evolving shared substance of knowledge
open knowledge graph, Dbpedia, Freebase, Wikidata, YAGO Enterprise KG, web search(google), commerce(Airbnb), Social network(Facebook), Finance(Bloomberg)
General applications: search, recommendations, personal agents, advertising, business analytic, risk assessment, automation
Graph to represent data, enhanced with ways to represent knowledge Extraction → enrichment → quality assessment → refinement Extract value from LARGE SCALE data To postpone the definition of a schema
use cases Find entities through arbitrary length Centrality Clustering Summarizing ML over graph

Overview

- Data Graphs
- Schema, Identity, Context
- Deductive formalism (Theory → Observation)
- Inductive techniques (Observation → Theory)
- Creation and enrichment
- Enumerates for KG assessment
- KG refinement techniques
- KG publication principles and protocols
- Prominent KG and their applications
- Summary and future directions

Data Graphs

Models
relational schema: costly remodeling, reloading, reindexing,
not knowing what data needed to be modeled, need to have flexibility for incomplete and diverse data

- Directed Edge-Labelled Graph: flexibility for integrating new data sources beside child, parent connection allow cycles to be represented RDF, Resource Description Framework: standard data model based on edge-labelled graph
- Graph dataset: a set of named graph and a default graph used in query linked data, web data, importance if tracking source of the data
- Property Graph: additional flexibility to store property for both nodes and edges , Neo4j possibility to transfer to/from graph datasets without loss of information direct edge labelled minimal, property graph more flexible
- Others: individual edge, nested graph, previous ones are the most popular ones

Data Graph

Querying, language

- Graph pattern, convert input graph to a table, homomorphism-based (multiple variable to be mapped on the same term)semantics,isomorphism-based semantics (variables mapped to unique terms)
- complex Graph Pattern, use relational algebra to transform outputs , operators like: projection column selection, union, equalities,... can give rise to multiple result(use distinct methods)
- navigational graph pattern, ability to include path expressions in queries, which can combine pathes or have restriction of shortest path or exclude repetition
- complex navigational, add algebra relational operator, path expression

Schema, Identity, Context

Large Scale Data → numerous data sources→ need of schema, identity and context
Data Graph: collection of data represented as nodes and edges using a graph model Knowledge Graph: a data graph potentially enhanced with representation of schema, identity, context ontologies and rules (embedded in the data graph or layer above)

Schema

data graph benefit, postpone the schema definition. s
CWA-closed world assumption, add info can change the false of statement
OWA-open world assumption, false remains false need to have a definite "yes"/"no" answer -¿ LCWA Local closed world Assumption, portions of the data is complete
Types:

- semantic Schema (High level structure for the KG, class, class hierarchy),OWA , best choice for default schema
- validating Schema, attempt to guarantee that the data graph is in some sense complete. adding constrains on the classes with defining shapes. validating schema consider the semantics but if a semantic exists adapting a validating schema is required Shape Expressions (SHEx), Shapes Constraint Language (SHACL)
- emergent Schema,summarize data graph into higher-level topology, human-understandable of the data graph to

Schema - continue

- aid definition of semantic and validating schema.
- optimize indexing and querying of the graph
- guide integration of data graphs

semantic and validating schema requires domain expert while data graph exhibit latent structure which is extracted as emergent schema framework: quotient graph, partition group of nodes and how edges are defined in data graph according to some equivalence relation while preserving some structural properties. different quotation graph will provide different guarantees with respect to the structure they preserve.

- quotient graph: simulation, bisimulation, preservation by bisimilarity, preserving forward directed path
- relational table
- formal concept analysis

Identity

which nodes in the graph refer to the same real world entity, to avoid ambiguity (in merging graph issues with name clashes)

- Persistent identifiers(PIDs), long lasting PIDs in order to unique, use global identifiers (Digital object Identifiers (DOIs), ORCID iDs, International Standard Book numbers (ISBN), Alpha-2, international resource identifiers (IRIs, URIs leave ambiguity, IRIs path has /entity for entities and /prop/entity for relationships-namespaces), HTTP IRIs have flexibility and power to generate global identifiers from URL, to enhance persistence (if page go offline), Persistent URL PURL, to state a local entity has the same identity as another *coreferent* entity in an external source.

Identity - continue

- External identity links, associate the identity with uniquely identifying information the graph(geo-coordinates, postal code), identity links
- Datatypes, processes: equality checks, normalization, ordering, transformations, casting,
- Lexicalisation,
- Existential nodes, for modeling incomplete information. permit the use of existential nodes, blank nodes or canonical labels, rather to minimize the use of such nodes

Context

specific setting in which some unit of the knowledge is true, scope of truth. implicit vs explicit context, explicit allows for interpreting the data from different perspectives.

- Direct representation, Time Ontology, PROV Data Model
- Reification, making statement about a statement in a generic manner, define edges about edges
- Higher-arity representation, named graph(most flexible with assigning context to multiple edges), property graph(Neo4j), RDF (least flexible)
- annotations, automated mechanism for reasoning about context,

frameworks: contextual knowledge repositories, OnLine Analytic Processing (OLAP) with operation like "slice-and-dice", selecting according to given dimensions, "roll-up", agreeing knowledge at higher level.

Deductive Formalism (Theory → Observation)

human can deduce more from the data, with general rules about the world we know a priori, to know more that what explicitly is given by the data, “common sense knowledge”. when it is shared by few experts in an area, domain knowledge. Machines don’t have a priori access. once a priori is instructed, machines apply deductions with a precision, efficiency and scale beyond human performance. example of logical frameworks: First-Order Logic, Datalog, Prolog, Answer Set Programming, here focus is on graph based ontologies.

- Ontology, concrete formal representation of what terms mean within the scope, can guide how graph data are modeled, OWL, OBOF- Open Biomedical Ontologies Format
  - Interpretation, which interpretations are valid
  - Individuals, Properties and Classes
  - others, annotation properties (versioning, datatype), OWL 2 standards
- Semantic and Entailment
  - Model theoretic semantics, axiom enforces some conditions
  - Entailment, if-then vs if-only-if
- Reasoning,
  - Rules (N3, RIF, SWRL, SPIN) : Materialisation, Query rewriting
  - Description logic

Inductive Techniques (Observation → Theory)

Generalizing patterns, novel but potentiality imprecise predictions which is associated with a level of confidence.

- self-supervision: learn a function which generates the input-output pairs automatically from the input
- unsupervised: no input-output pair, apply predefined functions (statistical in nature) to map input to output.

Common techniques:

- Graph Analytics (central node-edges, interesting paths), un-supv
- KG Embeddings - adapt graph to ML method, self-supv
- GNN - adapt graph to ML, self-supv
- Symbolic Learning - learn logical formulae that can be combined with deductive reasoning

Graph analytics - Inductive Techniques

Process of discovering, interpreting and communicating meaningful patterns inherent to data collections (typically large).
Draws its techniques from graph theory and network analysis. use cases: social networks, the Web, internet routing, and more

Techniques:

- Centrality, find central nodes and edges.
  Measures: degree, betweenness, closeness, Eigenvector, PageRank, HITS, Katz
- Community detection, sub-graph that are densely connected
  minimum-cut, label propagation, Louvain modularity
- Connectivity,estimate how well-connected the graph is to reveal resilience and (un)reachability of elements.
  graph density, k-connectivity
- Node similarity, by virtue of how nodes are connected within their neighbourhood. Node similarity matrices:
  structural equivalence, random walks, diffusion kernels, etc.
- Path finding, discover interesting paths between pairs of nodes.
  simple path, shortest path

Graph analytics - Frameworks

for large scale graph-analytics often in a distributed (cluster) setting. Parallel frameworks that applies systolic abstraction based on directed graph. Nodes are processors that can send messages to other nodes along edges. These define the systolic computation abstraction on top f the data graph being processed. Taking care of distribution, message passing, fault tolerance, etc. Methods on “native” graph.

Frameworks

Apache Spark(GraphX), GraphLab, Pregel, Signal-Collect, Shark

- message phase (MSG), functions to compute message values
- aggregation phase, accumulate messages

Its limitation, not all analytics can be expressed in this framework. Some additional features:

- global step, perform a global computation on all nodes, making it available to each node
- mutation step, allows adding or removing nodes and edges during processing.

Graph analytics - Analytics on data graphs

data graph as a subject of analytics

- Projection, selecting a sub-graph from the data-graph. all edge meta-data are dropped
- Weighting, converting edge meta-data to numerical values according to some functions
- Transformation, transforming the graph to a lower arity model, lossy/lossless (original graph can/cannot be recovered)
- Customisation, changing analytical procedure to incorporate edge meta-data

The results of an analytical process may change drastically depending on which strategies are chosen to prepare the data for analysis.

Graph analytics - Analytics with queries

Query languages and analytics may complement each other. Language: SPARQL, Cypher, G-CORE

- allow for outputting sub-graph
- express some limited (non-recursive) analytics
- , optimization to distribute a large data graph over multiple machines
- rank query results

Graph analytics - Analytics with entailment

Graph schema or ontology that defines the semantics of domain terms is giving rise to entailment. The combination of analytics and entailment has not been well-explored.

- semantically-invariant analytics that yields the same results over semantically-equivalent graph

KG Embedding - Inductive Techniques

ML in the context of knowledge graph,

- refining the knowledge graph(predict new edges, identify erroneous edges)
- downstream tasks, train models for classification, recommendation, regression traditional techniques assume inputs in form of vectors, which is distinct from graphs itself
  methods in vector
  - one-hot encoding, a vector for each node with length of $|L|.|V|$, where $|L|$, the number of edge labels, $|V|$, the number of nodes in the input graph ¿ large sparse vectors which will be detrimental for ML models

  KG embedding, create a dense representation of the graph(embed the graph) in a continuous, low dimensional vector space to be used in ML tasks. The dimensionality is fixed $50 >= d >= 1000$,
  entity embedding for each node, e vector with d dimension denote by e
  relation embedding for each edge label: a vector with d dimension denote by r
  Goal: abstract and preserve latent structure in the graph.

KG Embedding - Models

Transitional, geometric perspective

Interpret edges as transformation from subject nodes to object nodes.
TransE, TransH, TransR, TransD

Tensor decomposition, rank decomposition

DistMult, RESCAL, HolE, ComplEx, SimplE, TuckER (state of the art)

Neural, non-linear scoring functions

Semantic Matching Energy (SME), Neural Tensor Networks (NTN), Multi Layer Perception (MLP), ConvE (using convolutional kernels), HypER (avoid the need to wrap vectors embedding into matrices. Instead, applies a fully connected layer. Outperform ConvE)

KG Embedding - Models

Language

RDF2Vec (random walk into sentences to as input to word2vec, KGlove (based on Glove model)

Entailment-aware use deductive knowledge to improve the embeddings

Wang et al (use constraint rules to refine the predictions), KALE (joint embedding, consider both the data graph and rules while computing the embeddings, complex rule), RUGE (joint model over ground rules), FSL (simple rule)

GNN - Inductive Techniques

KG Embedding: provide a dense numerical representation of graphs suitable for use within existing ML models.
GNN - to build custom ML models adapted for graph-structured data
Graph looks similar as NN, but NN is homogeneous while graph topology is heterogeneous.
GNNs builds a NN based on the data graph. A model is learnt to map input features for nodes to output features in supervised manner.

Recursive GNNs - RecGNNs

similar to the systolic abstraction, but learn the function instead, thereafter apply it to label other examples.

GNN - continue

Convolutional GNNs - ConvGNNs

Transition function is applied on a node and its neighbours instead of pixel and its neighbours.

Questions

- how the regions of the graph are defined
- can similar kernel be used overall

Solutions

- work with spectral or spatial representation of the graph.
- use attention mechanism, learn nodes whose features are most important to the current node

Symbolic learning - Inductive Techniques

KG embeddings and GNNs are often difficult to understand. An alternative is to adopt symbolic learning to learn hypotheses in a symbolic language. offer an interpretable model for new knowledge that will offer domain experts the opportunity to verify the models.

Rule mining

Discovering meaningful patterns in the form of rules from large collection of background knowledge.

Axiom mining

- Specific axiom, disgointness, between named classes / class description
- General axiom for class learning (DL-learner). used for: refinement operator, confidence scoring, search strategy, scoring functions based on query counting.

Symbolic learning - Rule Mining

Positive entailment - "confirmed" to be true
Goal - identify new rules that entail a high ratio of positive edges (observed) from other positive edges, but entail a low ratio of negative edges (assumption of completeness) from positive edges. High support and confidence.

- **rule positive entailments** - set of positive edges (not including entailed edge)
- **rule support** - number of the positive entailments
- **rule confidence** - ratio of the positive entailments ("confirmed")

Inductive Logic Programming (ILP)

Long explored method but KG challenges with the scale of the data.

Partial Completeness Assumption (PCA)

Dealing with incomplete KG (OWA), measures the ratio of the support to all entailments in the positive or negative set. - methods: AMIE, AMIE+

Differentiable rule mining, end-to-end learning of rules. NeuralLP, DRUM

Creation and enrichment

Human

- limitations: error, bias, costly, vandalism, disagreement
- Good for collaboration to improve or verify the KG

Text sources

Pre-processing, Named Entity Recognition (NER), Entity Linking (EL), Relation Extraction (RE), Joint tasks

Markup Sources

Imprecisely extracted, Language samples: HTML, Wikitext, TeX, Markdown

Structured Sources

Precisely transformed, Language samples: csv, Json, XML, relational databases

Schema-Ontology Creation

Generate schema based on external source of data including human knowledge

Creation and enrichment - Text sources

- Pre-processing: Part-of-Speech (POS) tagging, Dependency Parsing, Word Sense Disambiguation (WSD) to identify meaning (WordNet, BabelNet)
- Named Entity Recognition (NER): supervised, bootstrapping, distant supervision, manually crafted rules. Output: emerging entity (generate new node), entity linking
- Entity Linking (EL): challenges multiple name for the same thing and same name for different things. improvements with. context, centrality
- Relation Extraction (RE)
  - Binary RE: Closed setting (fix set relation): Manual pattern, supervised (manual-labeled, bootstrapping, distant supervision), Open setting - unsupervised: Open Information Extraction (OIE)
  - N-array RE: Frame semantic and Frame of words (FrameNet)
  - Discourse Representation Theory (DRT), logical expression of text based on existential events
- Joint tasks: WSD-EL, NER-EL, NER-RE

Aligning different REs

- Mapping and rules for aligning n-array, distributional and dependency based similarities
- Rule mining, Markov clustering, linguistic techniques for aligning OIE among others

Creation and enrichment - Markup Sources

Extraction Techniques:

- General, wrapper of a specific format (manual, (semi-)automatically and distant supervision (LODIE))
- Focused, specific form (web table extraction)
- Form-based,

Creation and enrichment - Structured Sources

It is done in two steps with custom mapping or direct mapping (automatically generate graph):

- create mapping from source to graph
- use the map to materialize (use Extract-Transform-Load (ETL) to map the table) and virtualization (Query Rewriting (QR), query translated ex. Ontology based Data Access-OBDA)

Techniques

- Map from table
- Map from trees (XML, Json), GRDLL standard (XML to (RDF) graph), JSON-LD standard (JSON to (RDF) graph)), XSPARQL (query from XML and RDF, supports materialization and virtualization)
- Map from other KG, source samples DBpedia, LinkedGeoData, Wikidata, YAGO and BabelNet. Procedure:
  - Extract a relevant sub-graph, (SPARQL)
  - Entity and schema alignment, (event-centric knowledge, spatiotemporal KG)

Creation and enrichment - Schema/Ontology Creation

Ontology Engineering - manually maintained

fixed (old) vs iterative and agile methods. ex. DILIGENT, XD, MOM, SAMOD. key elements:

- Ontology requirements: specify the indeed task (Competency Question - CQs)
- Ontology Design Patterns (ODPs), specify generalisable ontology modeling as modeling templates or directly reusable components.

Ontology Learning - (semi-)automatically

extract info from text that is good for ontology engineering.

- Terminology extraction, unithood(n-grams) and termhood(how relevant in domain)
- Axiom extraction, modify noun+adj (incrementally specialise concept), Hearst patterns, Textual definition - hypernym relation

Enumerates for KG assessment - Quality assessment

After creation and enrichment it is required due to:

- usually incomplete graph
- existence of duplicate, contradictory or incorrect statement

Quality is , fitness for the purpose.

- capture qualitative aspect of the data
- Quality metrics, ways to measure quantitative aspects
- Grouping of the dimensions

Accuracy

- Syntactic, Degree data is correct with respect to defined rules for the model (ex. data type) meas: No. incorrect values of a property/ total No. of values
- Semantic, how correctly it is representing the real world. Manual verification, check stated relations against several sources, validate individual process with precision (human expert)
- Timeliness, how up-to-date is the KG, meas: KG update frequency / source updates

Coherency

Evaluate how kg conform schema semantics and constrains. Consistency (KG is free of contradictions) and validity (KG is free of constraint violations)

Enumerates for KG assessment - Quality assessment

Coverage

Avoiding omission of domain-relevant elements. Problem: incomplete query results or entailment

- Completeness, Schema (classes and properties), property (ratio of missing values of a property), population (ratio of all-real-world entity), linkability
- Representatives, focus on high-level bias (assume graph is incomplete). Bias in: data, schema and during the reasoning

Succinctness

Inclusion of only relevant content.

- Conciseness, avoids information irrelevant to the domain. Intentional (no redundant schema elements), Extensional (no redundant entity or relationship)
- Representational, refers to the content which is compactly represented in KG. Intentional and Extensional.
- Understandably, refers to the ease that data can be interpreted.

KG refinement techniques

Completing and correcting the KG (semi-)automatically. No extraction or mapping from external sources, instead targets improvement of the local KG.

Completeness

Inclusion of only relevant content. Prediction of:

- General Link, (bus, flight, type, etc), inductive techniques like KG embedding or rule/axiom mining,
- Type Link, traditional classification tasks or GNN (ex. Santiago-type-city)
- Identity Link, searching for nodes that refer to the same entity. Considered in general integration settings. **Value matches**, similarity matrices on string, numbers, etc. **Context matches**, similarity of entities based on various nodes and edges.

Correctness

Identify and remove existing incorrect edges.

- Fact validation, assign a plausibility/veracity to a given fact/edge, typically in reference to external sources (structured or unstructured). It is online assessment (link prediction is offline)
- Inconsistency repairs, resolve inconsistency found in KG through ontology axiom, by experts or symbolic learning. Detection is not enough, requires to repair it as well which is not trivial task.

KG publication principles and protocols

Making the KG accessible to the public.

FAIR Principle

context of publishing scientific data. Applies to data, metadata or both. Findability (IRIs in RDF, VoID, DataHub), Accessibility, Interoperability, Reusability.

Linked Data Principle

A technical basis for one possible way in which FAIR principles can be achieved.

- Use IRIs as names for things.
- Use HTTP IRIs so those names can be looked up.
- When a HTTP IRI is looked up, provide useful content about the entity that the IRI names using standard data formats.
- Include links to the IRIs of related entities in the content returned.