7 지식 그래프(Knowledge Graph) 에 대한 소개

컴퓨터AI공학부
천세진

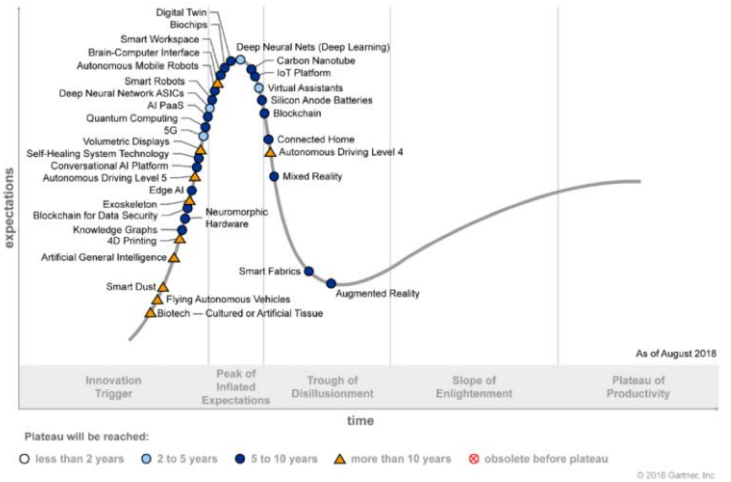
1

Goals

- Graph 기반 지식 표현에 대한 기본 개념 이해
- 그래프 데이터 관리 접근법과 Query Language 대한 기초 학습

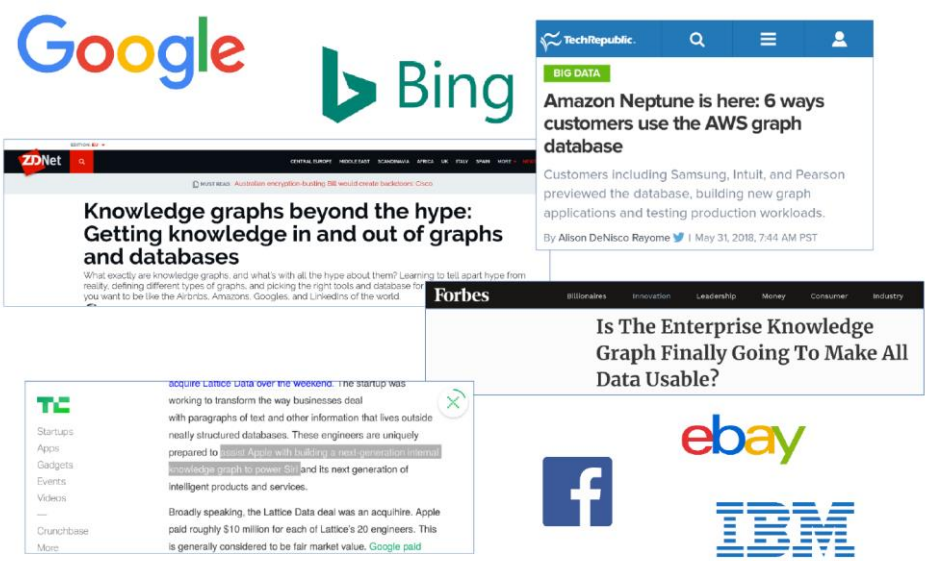
2

The Hype



3

Knowledge Graphs Everywhere



4

What is a Knowledge Graph?

The original “Knowledge Graph” (Google, 2012):

Many knowledge graphs, many techniques

- 매우 많은 지식그래프들이 누구나 이용가능한 형태로 존재함

WIKIDATA DBpedia BIO2RDF yago select knowledge Freebase

. . . and a variety of technologies for working with them:

RDF W3C SPARQL Apache TinkerPop Gremlin $G=(V,E)$

So what is Knowledge Graph?

- 초기정의

A Knowledge Graph is a knowledge base that is a graph.

- Knowledge base란?
  - "A knowledge base is a technology used to store complex structured and unstructured information used by a computer system. [. . .] [It] represents facts about the world" – Wikipedia (26 Oct 2020, id 983269427)
  - "A collection of knowledge expressed using some formal knowledge representation language." – Free Online Dictionary of Computing, 15 Oct 2018
  - 1. a store of information or data that is available to draw on.
    2. the underlying set of facts, assumptions, and rules which a computer system has available to solve a problem.

    – Lexico (Oxford University Press/Dictionary.com), 26 Oct 2020

So what is Knowledge Graph?

- What is a graph?
  - "a collection of points and lines connecting some (possibly empty) subset of them" – Wolfram MathWorld, 26 Oct 2020
  - "a collection of vertices and edges that join pairs of vertices" – Merriam-Webster, 26 Oct 2020
  - "a structure amounting to a set of objects in which some pairs of the objects are in some sense 'related'." – Wikipedia (26 Oct 2020, id 984093316)

So what is Knowledge Graph?

- In summary:
  - a collection of facts, rules, or other forms of knowledge
  - that express some kind of relationships or connections

  ⇝ a paradigm rather than a specific class of things

What is special about Knowledge Graphs?

- 두번째 정의

A Knowledge Graph is a data set that is:

- structured (in the form of a specific data structure)
- normalised (consisting of small units, such as vertices and edges)
- connected (defined by the – possibly distant – connections between objects)

Moreover, knowledge graphs are typically:

- explicit (created purposefully with an intended meaning)
- declarative (meaningful in itself, independent of a particular implementation or algorithm)
- annotated (enriched with contextual information to record additional details and meta-data)
- non-hierarchical (more than just a tree-structure)
- large (millions rather than hundreds of elements)

(Counter-)Examples

Typical knowledge graphs:

- Wikidata, Yago 2, Freebase, DBpedia (though hardly annotated)
- OpenStreetMap
- Google Knowledge Graph, Microsoft Bing Satori (presumably; we can't really know)

(Counter-)Examples

Typical knowledge graphs:

- Wikidata, Yago 2, Freebase, DBpedia (though hardly annotated)
- OpenStreetMap
- Google Knowledge Graph, Microsoft Bing Satori (presumably; we can't really know)

Debatable cases:

- Facebook's social graph: structured, normalised, connected, but not explicit (emerging from user interactions, without intended meaning beyond local relations)
- WordNet: structured dictionary and thesaurus, but with important unstructured content (descriptions); explicit, declarative model
- Global data from schema.org: maybe not very connected
- Document stores (Lucene, MongoDB, etc.): structured, but not normalised; connections sub-ordinary

(Counter-)Examples

Typical knowledge graphs:

- Wikidata, Yago 2, Freebase, DBpedia (though hardly annotated)
- OpenStreetMap
- Google Knowledge Graph, Microsoft Bing Satori (presumably; we can't really know)

Debatable cases:

- Facebook's social graph: structured, normalised, connected, but not explicit (emerging from user interactions, without intended meaning beyond local relations)
- WordNet: structured dictionary and thesaurus, but with important unstructured content (descriptions); explicit, declarative model
- Global data from schema.org: maybe not very connected
- Document stores (Lucene, MongoDB, etc.): structured, but not normalised; connections sub-ordinary

Primarily not knowledge graphs:

- Wikipedia: mostly unstructured text; not normalised; connections (links) important but sub-ordinary (similar: The Web)
- Relational database of company X: structured and possibly normalised, but no focus on connections (traditional RDBMS support connectivity queries only poorly)

13

컴퓨터 과학과 수학에서의 GRAPHS

14

What is a graph?

Definition 1.1: A simple undirected graph G consists of a set V of vertices and a set E of edges, where each edge is a set of two vertices. Two vertices $v_1, v_2 \in V$ are adjacent (in G) if there is an edge $\{v_1, v_2\} \in E$.

- Vertices들은 nodes라고 불리우기도 하며, undirected edges들은 arcs로 불리우기도 한다.
  - 특별한 언급이 없다면, 모든 그래프는 finite하다고 가정함

Directed and other graphs

Definition 1.2: A simple directed graph (a.k.a. simple digraph) G consists of a set V of vertices and a set $E \subseteq V \times V$ of (directed) edges from a source vertex to a target vertex.

- 화살표를 사용하여 directed edges를 표현하기도 함 $v_1 \xrightarrow{e_1} v_2$.

Directed and other graphs

Definition 1.2: A simple directed graph (a.k.a. simple digraph) G consists of a set V of vertices and a set $E \subseteq V \times V$ of (directed) edges from a source vertex to a target vertex.

- 화살표를 사용하여 directed edges를 표현하기도 함 $v_1 \xrightarrow{e_1} v_2$.

Definition 1.3: The following generalisations apply to directed and to undirected graphs.

- A graph with self-loops is a graph extended with the option of having edges that relate a vertex to itself.
- A multi-graph is a graph that may have multiple edges with the same vertices (in the same direction).
- An edge-labelled graph is a graph that has an additional labelling function $\lambda : E \to L$ that maps each edge in E to an element from a set of labels L (similarly for vertex-labelled graphs).

Other basic notions

Definition 1.4: An edge are said to be incidental to the vertices it connects. The degree of a vertex is the number of edges that are incidental to it. In a digraph, the in-degree of a vertex is the number of edges pointing towards it; analogously for out-degree.

Other basic notions

Definition 1.4: An edge are said to be incidental to the vertices it connects. The degree of a vertex is the number of edges that are incidental to it. In a digraph, the in-degree of a vertex is the number of edges pointing towards it; analogously for out-degree.

Definition 1.5: A directed path in a digraph is a sequence of consecutive edges $v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \cdots \xrightarrow{e_n} v_n$. An undirected path is a sequence of edges that may point either way (or that are simply undirected).

A simple path (directed or undirected) is a path without repeated vertices other than possibly the first and last node.

Other basic notions

Definition 1.4: An edge are said to be incidental to the vertices it connects. The degree of a vertex is the number of edges that are incidental to it. In a digraph, the in-degree of a vertex is the number of edges pointing towards it; analogously for out-degree.

Definition 1.5: A directed path in a digraph is a sequence of consecutive edges $v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \cdots \xrightarrow{e_n} v_n$. An undirected path is a sequence of edges that may point either way (or that are simply undirected).

A simple path (directed or undirected) is a path without repeated vertices other than possibly the first and last node.

Definition 1.6: Two vertices are connected if there is an undirected path from one to the other. A graph is connected if any pair of two distinct vertices is connected. A digraph is strongly connected if there is a directed path from any vertex to any other vertex (hence: one directed path in either direction).

Representing graphs (1)

Definition 1.7: The adjacency matrix of a graph $G = \langle V, E \rangle$ is the boolean $|V| \times |V|$ matrix that contains, at any coordinate $\langle v_1, v_2 \rangle$, the value 1 if there is an edge connecting v_1 and v_2.

Notes

- 무방향 그래프의 Adjacency matrices 는 symmetric
- Loops 는 주대각(diagonal)이 1이다
- Matrix는 edges의 수를 저장함으로써 multi-graphs로 적용
- Matrix는 labels을 저장함으로써 labeled simple graph로 적용

Representing graphs (1)

Definition 1.8: The adjacency list of a graph $G = \langle V, E \rangle$ is the list of all of its edges.

Notes

- Edges는 pairs 형태로 쓴다
- Loops는 반복된 vertices를 가진 edges로 표현
- Adj. list는 각 line에 edges의 수를 추가함으로써 multi-graph로 적용
- Adj. list는 각 line에 labels을 추가함으로써 labelled graphs로 적용
- Adj, list는 V를 encode하지 않음
  - V를 표현하기 위해 별도의 list를 구축

Which graph representation to pick?

Each representation has its pros and cons:

- Matrix:
  - \+ space efficient for dense graphs (1 bit per edge);
  - \+ can be processed with matrix operations (highly parallel);
  - – space inefficient for sparse graphs;
  - – not natural for labelled multi-graphs
- List:
  - \+ space efficient for sparse graphs;
  - \+ easy to use for labelled multi-graphs;
  - – harder to process (esp. if edge order can be random);
  - – not space efficient for dense graphs

Note: Knowledge graphs are typically sparse and labelled, but parallel processing still makes matrices attractive in some applications.

Credits

- TU DRESDEN
  - Markus Krotzsch