Knowledge graphs

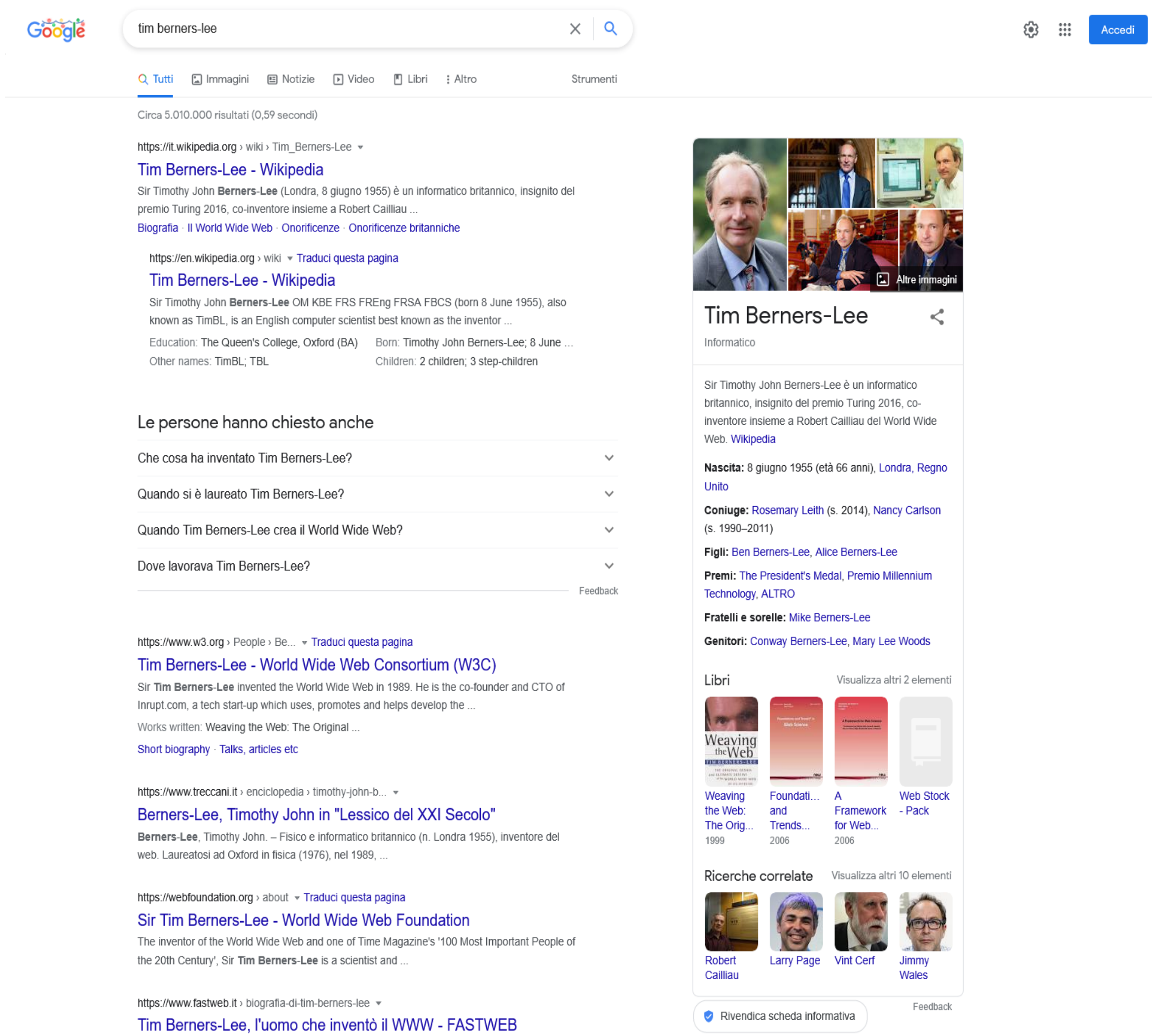
Riccardo Rosati

Knowledge Representation and Semantic Technologies
Master of Science in Engineering in Computer Science
Sapienza Università di Roma
a.a. 2021/2022

http://www.diag.uniroma1.it/rosati/krst/

Knowledge Graphs

"Knowledge graph" is a term that was re-introduced in Knowledge Representation after the creation of the **Google Knowledge Graph** in 2012



The Google Knowledge Graph is a knowledge base used by Google to enhance its search service

Google Knowledge Graph

- The Google Knowledge Graph is a very large knowledge base with a graph-like structure
- In 2020, it contained about 500 billion facts about 5 billion entities
- The detailed structure of Google Knowledge Graph is not public
- It is based on **Wikidata**, Wikipedia and other sources
  - Wikidata is in turn a knowledge graph (a project of the Wikimedia Foundation that was partially funded by Google)
  - Wikidata contains about 100 million items (2021)
- The information of the Google Knowledge Graph is used by the search engine to build the "infobox" appearing in the search results page

Knowledge graphs

Nowadays, the term knowledge graph is used to denote a (usually very large) graph-structured knowledge base

Knowledge bases specified in different formalisms are called knowledge graphs. E.g.:

- Google Knowledge Graph
- Facebook Knowledge Graph
- Wikidata
- Yago
- Graph databases (like Neo4J datasets)
- RDF models
- OWL ontologies
- …

A formalization of knowledge graphs

A possible, general definition of a knowledge graph:

A knowledge graph is a set of triples (or triplets) (h,r,t), where h (head) and t (tail) are **entities** and r is a **relation**

- i.e. the triple represents a directed edge (with label r) between the entities h and t

This is essentially the same as RDF triples

So, what is new about knowledge graphs?

Knowledge graphs

The "Knowledge Graph era" in Knowledge Representation has been characterized by the increasing application of **statistical** and **Machine Learning** techniques to knowledge bases

This idea is not new in Knowledge Representation, but:

With respect to previous approaches, the availability of very large datasets (knowledge graphs) and the progress in Machine Learning have produced much more interesting results

A key concept in this direction is the notion of **Knowledge Graph Embedding**

Knowledge graph embedding

An embedding of a knowledge graph is a projection of the entities and relations of a knowledge graph in a continuous low-dimensional space

Every entity and every relation is represented by a vector of continuous values

In this numerical representation, Machine Learning and Deep Learning techniques can be used to solve interesting problems:

- Triple classification (deciding whether a triple is true or false)
- Link prediction (assigning a score expressing the likelihood of a triple, entity/relation prediction)
- Clustering
- Entity recognition (determining whether two entities represent the same object)
- …

Knowledge graph embedding

The representation space of the embedding is a k-dimensional space (k is a hyperparameter of the embedding)

Every entity and relation is represented by a vector of k values

The process of identifying the embedding is driven by a **scoring function** f and a **loss function** L

Different choices of these functions (and of encoding models) are made by different embedding techniques

Synthetic negatives (false triples) are defined starting from the triples in the knowledge graph

Such negative examples are needed by the learning algorithms

Scoring function

The scoring function f assigns a score to every triple

The score must be proportional to the probability of the triple to be true

E.g. (TransE): $f(h,r,t) = - \| (\mathbf{h} + \mathbf{r}) - \mathbf{t} \|_n$

where **h** is the embedding of h (i.e. the vector representing h), **r** is the embedding of r, and **t** is the embedding of t

Loss function

A loss function L drives the training process

The goal is to minimize the value of the loss function

Example (TransE):

$$\mathcal{L} = \sum_{(h,\ell,t)\in S} \sum_{(h',\ell,t')\in S'_{(h,\ell,t)}} \left[\gamma + d(\boldsymbol{h} + \boldsymbol{\ell}, \boldsymbol{t}) - d(\boldsymbol{h'} + \boldsymbol{\ell}, \boldsymbol{t'})\right]_+$$

(S is the knowledge graph, S’ are the synthetic negatives)

Encoding models

- Geometric models (e.g. translational models: TransE, TransH, TransR,…)
- Tensor decomposition models (bilinear, non-bilinear)
- Deep learning models (RNN, CNN, …)

References

- Aidan Hogan et al.: Knowledge Graphs. ACM Comput. Surv. 54(4): 71:1-71:37 (2021)
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko: Translating embeddings for modeling multi-relational data. NIPS, 2013
- Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, Juanzi Li: OpenKE: An Open Toolkit for Knowledge Embedding. EMNLP (Demonstration) 2018: 139-144
- Federico Bianchi, Gaetano Rossiello, Luca Costabello, Matteo Palmonari, Pasquale Minervini: Knowledge Graph Embeddings and Explainable AI. Knowledge Graphs for eXplainable Artificial Intelligence 2020: 49-72