Knowledge Graph – Enrich the Results in Search Engine and Recommender System
DBKDA - InfoSys 2021

Yan WANG
Capgemini Engineering, Direction of Research and Innovation (DRI)
yan.wang2@altran.com
# Dr. Yan WANG

<table>
<thead>
<tr>
<th>Year</th>
<th>Position</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020</td>
<td>Research Project Manager</td>
<td>DRI, Capgemini Engineering</td>
</tr>
<tr>
<td>2019</td>
<td>Computer Science Postdocs</td>
<td>LIP6, Sorbonne University</td>
</tr>
<tr>
<td>2018</td>
<td>Computer Science Postdocs</td>
<td>CRI, Paris 1 Panthéon-Sorbonne University</td>
</tr>
<tr>
<td>2018</td>
<td>Enterprise Engineering PhD degree</td>
<td>IMS, University of Bordeaux</td>
</tr>
<tr>
<td>2014</td>
<td>Enterprise Engineering Master</td>
<td>IMS, University of Bordeaux</td>
</tr>
<tr>
<td>2014</td>
<td>Software Engineering Master</td>
<td>Harbin Institute of Technology</td>
</tr>
<tr>
<td>2012</td>
<td>Computer Science Bachelor</td>
<td>Harbin Institute of Technology</td>
</tr>
</tbody>
</table>

**Research interests:**
- Discrete Event Modeling and Simulation
- Process Mining
- Fuzzy Logic
- Constraint Programming
- Reinforcement Learning
- NLP
- Knowledge Graph
- Recommender System.
Research and Development Project TNT

TNT (Talent Needs Trends) is a research and development project of the program Future of Engineering. The objective of this project is to propose a competence management system advanced and adapted to Capgemini. The purpose is to improve the synergy between skills, resources and customers, simplify the process of HR-Analytics.

TNT is launched from 2014 with a lot of propositions and development tools. This year, we focus on the matching tool with three parts – search engine, recommender system and constraint solver.
Context of Project TNT

• The rapid evolution of the competences of the candidates
• The continuing evolution of candidate experience and technology
• The rapid evolution of the requirements of the clients
• HR adaptation and treatment time is getting longer and longer
Knowledge Graph (KG) has been proposed to discover the relation information with the property of powerful language understanding and rapid data analysis. It is first proposed in 2012 by Google, a theory of semantic structure combining applied mathematics, computer graphics, information visualization and machine learning. Knowledge Graph is constructed based on “Entity-Relation-Entity” with the associated property on entity.
1. KG in ML

Typical Knowledge Graph
Application Scenarios of the Knowledge Graph
Search Engine in history

Google

Bing

Yahoo!

Baidu
Recommender System in History

1998    Amazon item-to-item recommendation
2004-Now Special sessions in recommender system in several important conferences & journals:
        AI Communications; IEEE Intelligent Systems; International Journal of Electronic Commerce; International
        Journal of Computer Science and Applications; ACM Transactions on Computer-Human Interaction; ACM Transactions on
        Information Systems
2007    First ACM RecSys conference
2008    Netflix online services (& innovative HMI)
2008-09 Netflix RS prize
2010-Now RS become essential: YouTube, Netflix, Tripadvisor, Last.fm, IMDb, etc...
Why Knowledge Graph?

With the help of KG, users can get a more accurate recommendation as well as the explanations for recommended items. (Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He. A survey on knowledge graph-based recommender systems. IEEE Transactions on Knowledge and Data Engineering. 2020)


2. Things, not Strings

Knowledge Representation
Knowledge Modeling
Knowledge Representation

- Scope of the knowledge
- Compatible for machine
- Structural for scale
Knowledge Modeling

RDF = Resource Description Framework
Purpose: to provide a structure for describing identified things

OWL = Web Ontology Language
Purpose: to develop ontologies that are compatible with the World Wide Web
3. Processing

Knowledge Mining
Knowledge Storage
Knowledge Query
Knowledge Analytics
Knowledge Mining

- Decision Tree
- Genetic Algorithm
- Neural Network
- Fuzzy Logic
- Instruct data
- Struct Data
- Demi-struct data
Tools for Knowledge Storage

- OrientDB
- Nebula Graph v2.0 GA
- Apache Spark
- rdf4j
- JanusGraph
- Dgraph
- HyperGraph DB
- HugeGraph
- neo4j
Neo4j

The Neo4j Graph Platform is an example of a tightly integrated graph database and algorithm-centric processing, optimized for graphs. It is popular for building graph-based applications and includes a graph algorithms library tuned for its native graph database.
Proposed Knowledge Graph

Label - part of a group:
- Competence Keywords

Relationship type:
- Has
- Is a

Scale-free network - a scale-free network is produced when there are power-law distributions and a hub-and-spoke architecture is preserved regardless of scale, such as in the World Wide Web.

Flavors of Graphs:
- Disconnected
- Unweighted
- Tree
- Sparse
- Bipartite – function and specialty
Graph Search Algorithms

Breadth First Search
Visits nearest neighbors first

Depth First Search
Walks down each branch first
Pathfinding Algorithms

Shortest Path
- Shortest path between 2 nodes (A to C shown)
- Calculated by number of hops

All-Pairs Shortest Paths
- Optimized calculations for shortest paths from all nodes to all other nodes

Single Source Shortest Path
- Shortest path from a root node (A shown) to all other nodes
- Traverses to the next unvisited node via the lowest cumulative weight from the root

Minimum Spanning Tree
- Shortest path connecting all nodes (A start shown)
- Traverses to the next unvisited node via the lowest weight from any visited node

Random Pathfinding Algorithm
- Random Walk
  - Provides a set of random, connected nodes by following any relationship, selected somewhat randomly
  - Also called the drunkard’s walk
## Overview of Pathfinding and Graph Search Algorithms

<table>
<thead>
<tr>
<th>Algorithm type</th>
<th>What it does</th>
<th>Spark example</th>
<th>Neo4j example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breadth First Search</td>
<td>Traverses a tree structure by fanning out to explore the nearest neighbors and then their sublevel neighbors</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Depth First Search</td>
<td>Traverses a tree structure by exploring as far as possible down each branch before backtracking</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Shortest Path</td>
<td>Calculates the shortest path between a pair of nodes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>All Pairs Shortest Path</td>
<td>Calculates the shortest path between all pairs of nodes in the graph</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Single Source Shortest Path</td>
<td>Calculates the shortest path between a single root node and all other nodes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Minimum Spanning Tree</td>
<td>Calculates the path in a connected tree structure with the smallest cost for visiting all nodes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Random Walk</td>
<td>Returns a list of nodes along a path of specified size by randomly choosing relationships to traverse</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Embedding Method</td>
<td>Authors</td>
<td>References</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>Meta-path + Bayesian</td>
<td>Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han.</td>
<td>Personalized entity recommendation: A heterogeneous information network approach. In Proceedings of the 7th ACM International Conference on Web search and data mining, pages 283–292. ACM, 2014.</td>
<td></td>
</tr>
</tbody>
</table>
4. Propositions

Search Engine (DBKDA paper 2021)
Recommender System
Proposed Approach of Search Engine

In the search engine, we take two kinds of files as input – CV and clients inquiries. We make a pre-processing to remove the ambiguous (for example, “JS” is “Java Script”). Then we use NER tool based on DistilBERT-base-multilingual-case to extract the competence keywords and TF-IDF to calculate the score of the competence keywords. We also use Weight Average Method (WAM) to calculate a global score. The scores are stored in Redis with index and inverted index. We select a list of candidates with high score as well as the related competence keywords from the Knowledge Graph.
Proposed Approach of Recommender System

In the recommender system, we take two kinds of files as input – CV and mission. We make a pre-processing to remove the ambiguous (for example, “JS” is “Java Script”). Then we use NER tool based on DistilBERT-base-multilingual-case to extract the competence keywords and TF-IDF to calculate the score of the competence keywords. We also use Weight Average Method (WAM) to calculate a global score. The scores are stored in Redis with index and inverted index. We need to propose an adapted KGE method in order to choose a list of missions with high score.
# Perspectives of the Knowledge Graph

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Recommendation</td>
<td>It is natural to integrate other types of side information and build a KG for dynamic recommendation</td>
</tr>
<tr>
<td>Multi-task Learning</td>
<td>It would be interesting to exploit transferring knowledge from other KG-related tasks, such as entity classification and resolution, for better recommendation performance</td>
</tr>
<tr>
<td>Cross-Domain Recommendation</td>
<td>It could be promising to follow works by incorporating different types of user and item side information in the user-item interaction graph for better cross-domain recommendation performance</td>
</tr>
<tr>
<td>Knowledge Enhanced Language Representation</td>
<td>It is promising to apply the strategy of knowledge-enhanced text representation in the new recommendation task and other text-based recommendation tasks for better representation learning to achieve more accurate recommendation results</td>
</tr>
<tr>
<td>Knowledge Graph Embedding Method</td>
<td>Another research direction lies in comparing the advantages of different KGE methods under various conditions</td>
</tr>
<tr>
<td>User Side Information</td>
<td>Considering user side information in the KG could be another research direction</td>
</tr>
</tbody>
</table>
Thank you for your attention!

Yan WANG
Capgemini Engineering, Direction of Research and Innovation (DRI)
yan.wang2@altran.com