Knowledge Graphs for Recommender Systems

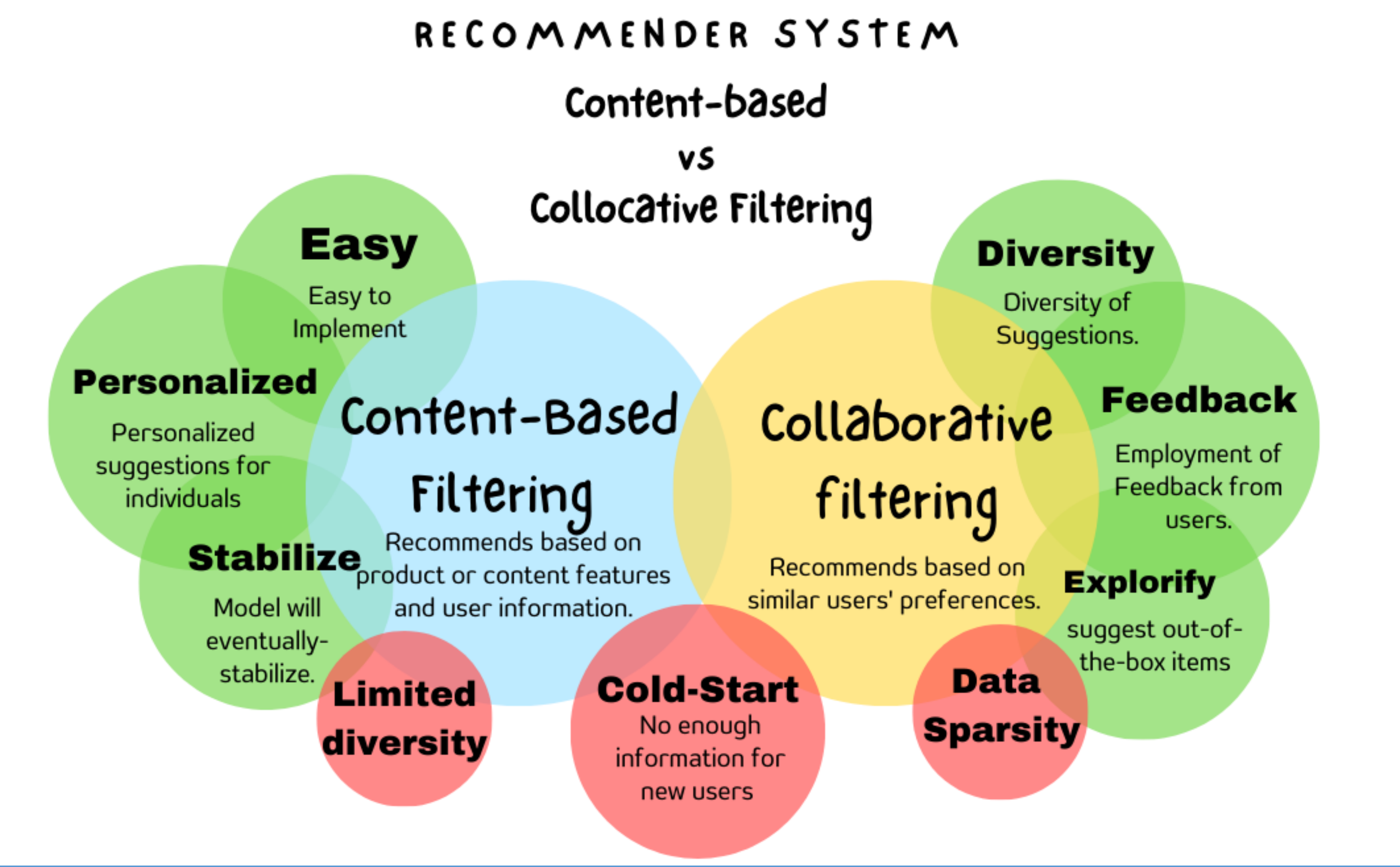
ALI G. A. ABUSALEH
ABD ALRHMAN MUSTAFA SALEEM ABU SBEIT
(ali.g.a.abusaleh | abd.alrhman.mustafa.saleem.abu)@estudiantat.upc.edu
Barcelona, Spain - eBISS 2023

UPC UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Introduction

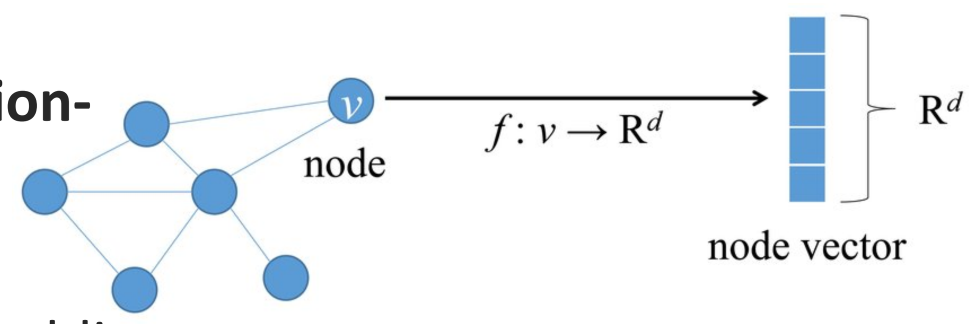
Knowledge graph recommender systems have emerged as a solution for the growing complexity and demand for accurate recommendations (1).
By utilizing **compact embeddings** to represent **entities** and **relationships** in knowledge graphs, these systems leverage **structural** and **semantic** information for **improved** recommendations (2).

Background

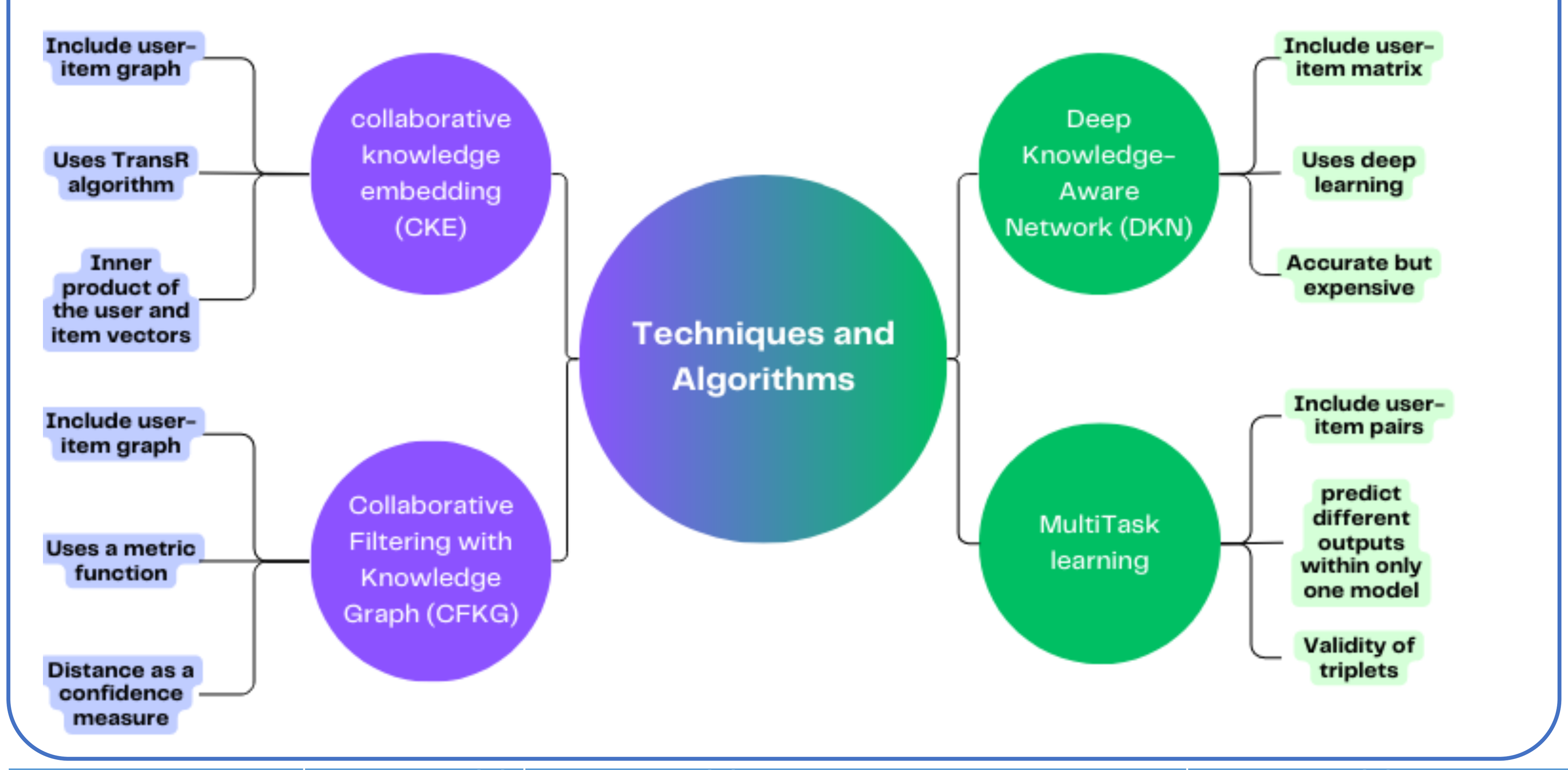


Embedding-Based knowledge Graphs-Based Recommender System

recommender systems involve representing **entities** and **relation-ships** from the knowledge graph as **compact**,
meaningful **vectors** called embedding.

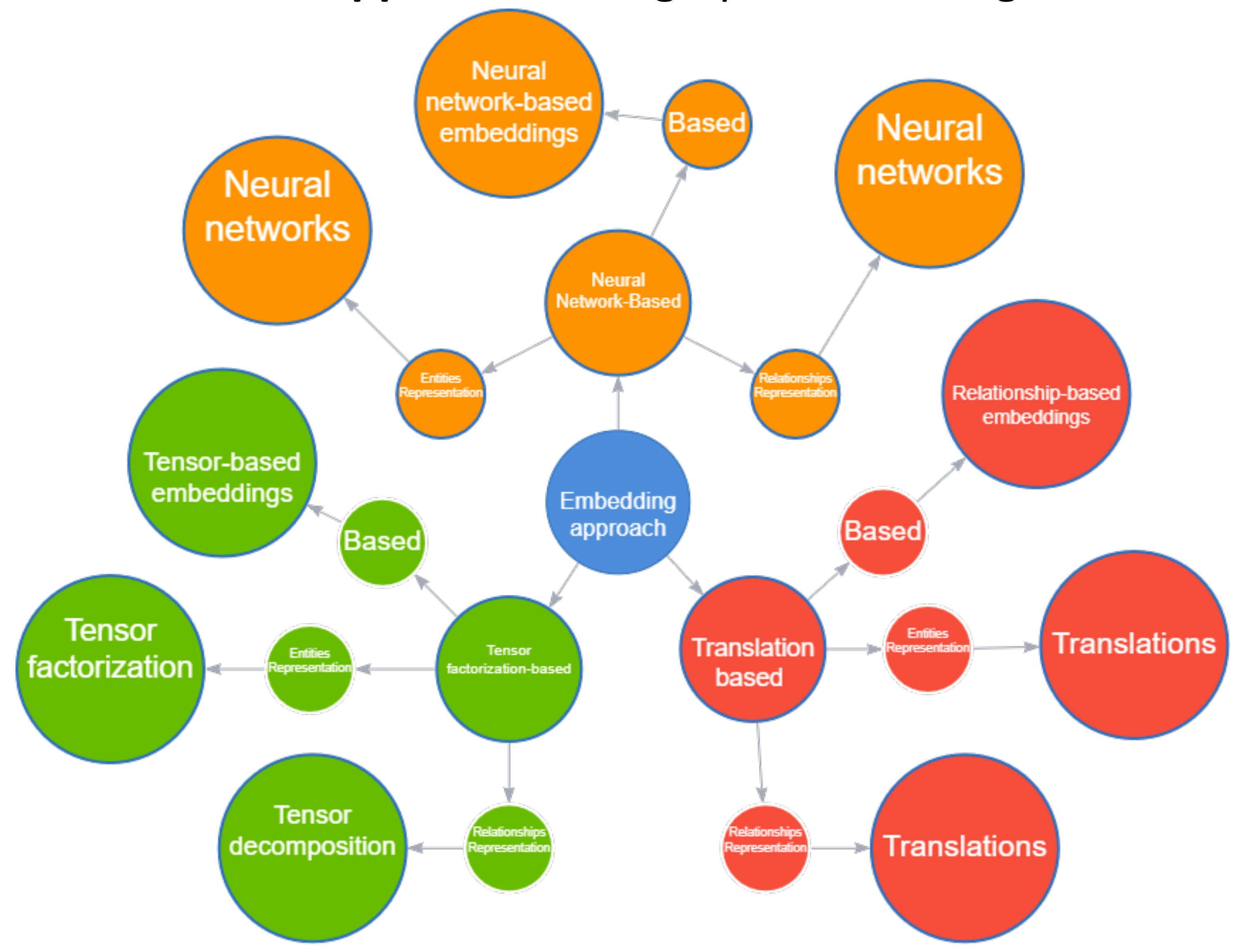


Auxiliary Information, Techniques and Algorithms



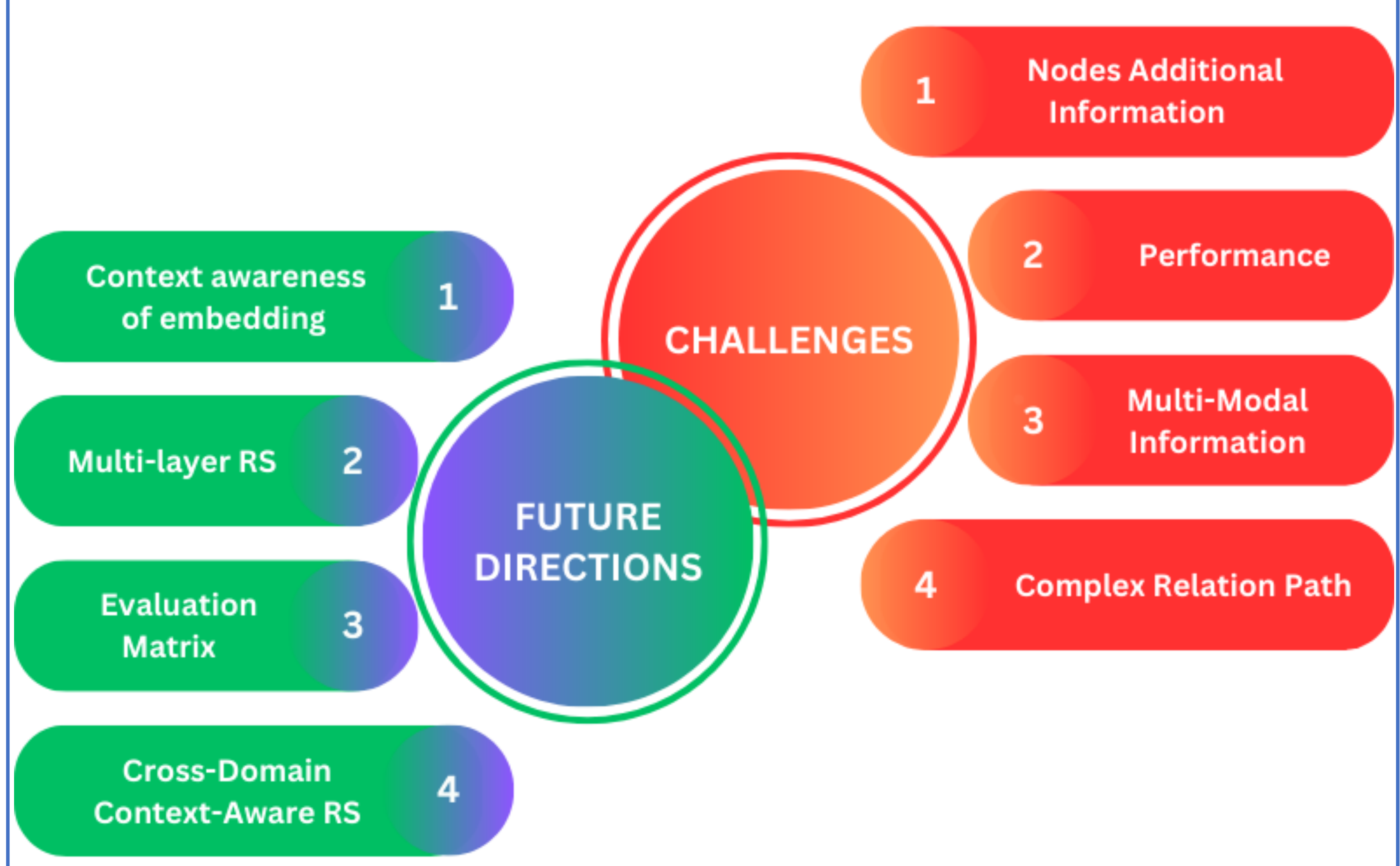
Embedding-Based knowledge Graphs-Based Approaches

There are different **approaches** for graph embedding:



In table [3], we are introducing three algorithms for each approach with some of their **advantages** and **limitations.**

Challenges and Future Directions



Conclusion

Knowledge graph-based recommender systems leverage semantic relationships between entities for personalized recommendations. Graph embeddings capture complex relationships, enhancing accuracy and personalization. Context awareness and multi-layer architectures improve tailored recommendations. Cross-domain approaches bridge gaps for comprehensive recommendations, advancing user satisfaction.

References

Algorithm Family	Algorithm(s)	Advantage(s)	Limitation(s)	Key Idea
Translation based	TransE	Good performance for large-scale knowledge graphs.	1) It can handle only one-to-one relationships effectively 2) Assuming all relations are located in Single semantic space	TransE learns translation vectors to represent relationships between related entities. **By minimizing the distance** between the translated source entity and the target entity.
	TransH	1) Handle one-to-many and many-to-many 2) Reduce the false positive labeling	Hyperplane is on same space	TransH maps head and tail into **a new hyperplane**
	TransR	1) Having relation-specific spaces hyperplane 2) Handle one-to-many and many-to-many	Loosing the simplicity of the Translations-method based	Entities are represented as vectors in an **entity space R**, and each relation is associated with a **specific space R** and modeled as a translation vector in that space.
Tensor factorization based	RESCAL	Captures fine-grained interactions and semantic relationships in the knowledge graph	1) Complex and expensive Operation. 2) Not suitable for large-scale graphs.	Represents entities as vectors and relationships as **matrices**
	ComplEx	Allowing for the modeling of symmetric and asymmetric relations in the graph	1) Complex and expensive Operation. 2) Not suitable for large-scale graphs.	It represents entities and relationships as **complex vectors** by extending **RESCAL** by using complex-valued embeddings
	DistMult	Liner complexity	Works only for symmetric	Simplifies the tensor factorization process by **utilizing diagonal matrices for relationships.**
Neural network based	SME	Use semantics to predict relationships.	1) Relay on observed relationships to learn embeddings 2) Not suitable for large-scale graphs 3) Provide linear modeling only	SME uses **neural network** to model the semantic matching between entities. Then it measures the **confidence** of each triplet using NN-based function.
	ConvKB	Non-Linear modeling	Doesn't differentiate the importance of different relationship, and treats all relationships equally	ConvKB uses **matrix representation for triplets**, applies CNN for graph embeddings, and calculates **confidence scores**.
	R-GCN	Provide relation-specific transformation	1) Not suitable for large-scale graphs 2) Can face overfitting	R-GCN extends **GCNs** with relational graph convolutions, leveraging node features and relation types for relation semantics and node embeddings, **using linear transformation.**

Table 3: Comparison of various knowledge graph embedding algorithms