**“推荐算法不够精准？让知识图谱来解决”**

* 内容要点如何根据具体推荐场景的特点将各种辅助信息有效地融入推荐算法一直是推荐系统研究领域的热点和难点，如何从各种辅助信息中提取有效的特征也是推荐系统工程领域的核心问题。
* 知识图谱的维度更高，语义关系更丰富
* 用户和物品的交互信息往往是非常稀疏（sparse）; 冷启动问题---->常见思路是在推荐算法中额外引入一些辅助信息（side information）作为输入.
* 将知识图谱引入推荐系统的工作分为两类：
* 通用的基于特征的推荐方法（generic feature-based methods）：这类方法统一地把用户和物品的属性作为推荐算法的输入。此方法将知识图谱弱化为物品属性，但它并非专门针对知识图谱设计，因此无法高效地利用知识图谱的全部信息。例如，该类方法难以利用多跳的知识，也难以引入关系（relation）的信息。
* 基于路径的推荐方法（path-based methods）：优点是充分且直观地利用了知识图谱的网络结构，缺点是需要手动设计meta-path或meta-graph，这在实践中难以到达最优；同时，该类方法无法在实体不属于同一个领域的场景（例如新闻推荐）中应用，因为我们无法为这样的场景预定义meta-path或meta-graph。
* 知识图谱特征学习（Knowledge Graph Embedding）为知识图谱中的每个实体和关系学习得到一个低维向量，同时保持图中原有的结构或语义信息。知识图谱特征学习的模型分类两类：
* 基于距离的翻译模型（distance-based translational models）：使用基于距离的评分函数评估三元组的概率，将尾节点视为头结点和关系翻译得到的结果。TransE、TransH、TransR
* 基于语义的匹配模型（semantic-based matching models）。这类模型使用基于相似度的评分函数评估三元组的概率，将实体和关系映射到隐语义空间中进行相似度度量。SME、NTN、MLP、NAM
* 利用知识图谱特征学习，我们可以很方便地将知识图谱引入各种推荐系统算法中，且可以降低知识图谱的高维性和异构性；增强知识图谱应用的灵活性；减轻特征工程的工作量；减少由于引入知识图谱带来的额外计算负担。

**“如何将知识图谱特征学习应用到推荐系统？”**

* 将知识图谱特征学习应用到推荐系统中主要通过三种方式——依次学习、联合学习、以及交替学习。
* 遵循依次学习的框架，我们首先需要提取知识图谱特征，步骤：
* 实体连接（entity linking）。即从文本中发现相关词汇，并与知识图谱中的实体进行匹配；
* 知识图谱构建。根据所有匹配到的实体，在原始的知识图谱中抽取子图。子图的大小会影响后续算法的运行时间和效果：越大的子图通常会学习到更好的特征，但是所需的运行时间越长；

知识图谱特征学习。使用知识图谱特征学习算法（如TransE等）进行学习得到实体和关系向量。

四个类的特点是每一句都有特征词，而且不需要多跳，考虑能否用基于特征推荐的方式，