目的：

在推荐系统中，通常通过观察用户-项目评分矩阵来进行算法的推导和研究。但是，随着电子商务服务中用户和项目数量的爆炸式增长，导致用户-项目矩阵的维度呈现指数增加的趋势，伴随而来的就会造成评分矩阵的稀疏。该特性也是推荐系统质量恶化的主要因素之一。为了提高准确性，用于特定场景的推荐技术要求不仅考虑用户的评分信息，还要考虑用户所在地区的人口统计信息，基于用户行为的社会网络以及指定相关项目描述的一些辅助信息。目前，研究者们已经提出了基于文档建模方法（如LDA主题模型等方法）以及能够产生更加准确的潜在模型的CDL协同深度学习的方法。在最近的工作成果里，利用卷积神经网络（CNN）来进行文档感受野的采集，并利用下采样来防止过拟合的思路获得了一定成都的成功。但是，使用CNN进行用户-项目矩阵分解（ConvMF）的缺陷在于其不能体现出单个句子中的词序关系。而词序在一定程度上决定了语句的内在含义和思想。在本工作中，我们计划提出一种R-ConvMF的方法，即尝试在现有的ConvMF中嵌入RNN（循环神经网络层），以挖掘单个语句向量中词序的关系。这样一来，不仅增强了多个文本向量之间的局部感受野（CNN效果），还针对单个句子做了更进一步的信息挖掘（RNN），最终以获得更为准确的推荐效果。

现状和发展趋势：

现有的基于文档的建模方法利用项目描述文档(评论，摘要或简介)来进行语句向量的回归，对比起传统的协同过滤的方法，这种方法较为···············································································································································································合理地利用了辅助信息，从而在评分预测精度方面产生了一种更加准确的潜在模型。

但是，这种方法造成的缺陷在于，它们是基于词袋模型进行模型搭建的，也就是说每一个词在句子中的顺序是不确定的，也就是说，现有的集成模型并不能完全捕获句子的内建信息，因为他们不能利用预先训练的词嵌入模型（如Glove）

在最近的工作中，提出了一种CNN和MF结合的模型ConvMF，这种模型的优点在于其通过模型组件（卷积核）捕获了文档之间的上下文信息，通过增加CNN层和下采样层到PMF中，比较好地处理了上下文之间地信息稀疏的问题。但是，针对于单个语句中词序的关系，该模型没有对词粒度的信息进行挖掘。因此，对词粒度信息的挖掘是提升推荐系统性能的下一个发展方向

难点：  
1.RNN如何无缝连接到ConvMF中？（如何将RNN置于一个比较合理的上下文环境下？）

2.RNN的权重是否需要预训练？

3.loss如何计算？如何体现出单个语句的词序重要性？

参考文献：

[1] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric

Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang

Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and

mandarin. In International Conference on Machine Learning. 173–182.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine

translation by jointly learning to align and translate. arXiv preprint

arXiv:1409.0473 (2014).

[3] DavidMBlei, AndrewY Ng, and Michael I Jordan. 2003. Latent dirichlet allocation.

Journal of machine Learning research 3, Jan (2003), 993–1022.

[4] Emmanuel J Candès and Benjamin Recht. 2009. Exact matrix completion via

convex optimization. Foundations of Computational mathematics 9, 6 (2009), 717.

[5] Li Chen, Guanliang Chen, and Feng Wang. 2015. Recommender systems based