

Opinion Recommendation using Neural Memory Model

会议信息：EMNLP - 2017，作者：Zhongqing Wang and Yue Zhang

文章链接：http://people.sutd.edu.sg/~yue_zhang/pub/emnlp17a.zhongqing.pdf

摘要

本文提出并解决了一个全新的任务：观点推荐，即在当前有用户、物品的历史交互记录中，用户对购买或使用过的服务的 **打分信息** 和与之对应的 **评论**，现在需要对用户没有买过的商品 **生成可能的评论** 和 **预测一个打分**。这个问题的本质是一个基于多元数据的多任务联合学习（multi-task joint learning），而这正是神经网络模型的强项。本文中，作者使用了一个单层神经网络对用户和商品建模，挖掘他们之间的关联；同时使用了一个深度记忆网络模型来生成具有用户个性特征的商品模型，从而将个性化的打分信息和评论融合在一起，构成一个联合模型。

简介

在线购物或者是提供商业服务的网站上，有大量的用户的打分与文本评论。对于其他用户访问这些网站，首先看到的是对于某个商品的平均打分，在这之后会有一些其他用户的详细评论。但这样的信息展示方式还存在一些问题，用户的体验不佳，例如：

- 总体打分太过宽泛（通常是所有打分的平均值），与当前用户的品味有一定的差异
- 商品相关的评论通常成千上万，对于想要快速获取信息的用户来说，显然不可能每一条都认真阅读，有一个综述性的评论总结会更好

本文深入研究了问题的背景，给出了一个基本的解决方案。通过利用用户 u 写过的评论 r_1, r_2, \dots, r_n 对用户建模，利用所有用户对某商品 p 的评论 r'_1, r'_2, \dots, r'_m 对商品建模，从而生成用户 u 对商品 p 的一个预测打分和个性化的评论。这一任务被定义为“**观点推荐**”，与之相关的 NLP 研究主要有：情感分析、观点综述和推荐。

任务定义

在本文的工作中，即为用户 u 生成关于商品 p 的一个打分和评论，主要用到了如下三种数据作为输入：

1. 其他用户目标推荐商品 p 的其他用户的评论
2. 该用户 u 对于其他商品的评论
3. 其他用户对其他商品的评论

而最终的输出是两个：

1. 用户 u 对于商品 p 的个性化打分，实数值
2. 用户 u 对于商品 p 的个性化评论，文本

理想化的解决方案就是综合利用三种输入信息，联合训练，输出两种预测结果，因此，整个任务的重点和难点就在于从输入数据中抽取特征，而深度学习正可以通过若干全连接层融合多种信息，进行抽象、计算、推理，从而得到我们想要的结果，这样的优势也是解决该问题的自然的选择。

相关工作

情感分析

目前情感分析的重点在于如何从数据（文本）中抽取、设计文本特征，也开始借助外部（文本之外）数据对用户进行建模，以获得更为准确的具有用户某种喜好色彩的特征，因为同一个词对于不同用户来说的感情程度是不同的。

与传统的情感分析不同，本任务的前提并没有用户已经写好的评论，而是针对新商品的情感打分。

观点综述

这是一个很早就开始做的 NLP 任务，大部分工作主要在提取与观点相关的词或短语上。典型的方法有：使用关联关系挖掘找出频繁项集作为方面 (aspect) 词、基于序列化标注的方法、主题模型等，近期开始使用 word embedding 和 RNN 技术抽取这些观点方面词。

这种基于抽取的方式，很难体现出每个方面对于打分的贡献大小，更进一步，有人从文本中根据句子的可靠性、信息量和可读性特征，对句子进行选择 and 排序，生成一段综述。目前研究热点已转移到生成方法上来了，有基于图的综述框架和更为强大的使用 attention 的神经网络模型，从输入的多个文本段中吸收重要的信息后保留，进而生成合适的总结。

虽然本任务也是从某一商品的评论中生成评论的总结，但是对于不同用户生成的评论综述是不同的，具有个性化的特征。

推荐

主流是基于内容的过滤（对于冷启动现象效果好些）和协同过滤。后者可以分为基于模型和基于近邻的方式。矩阵分解是最流行的做法，从用户-商品的历史交互信息中抽象出两者的特征。现在有很多工作开始把这种方法融合到其他 NLP 相关的任务，如情感分析中，主要通过集成学习，把从用户-商品打分矩阵中学到的商品特征作为隐变量融合到情感分析的文本特征中。

与本任务的不同主要是使用的信息是用户的 文本 评论数据而不是购买的数值记录。

神经网络模型

本文的定位：多任务联合学习，是神经网络的强项，因为神经网络模型可以通过共享参数和隐藏语义空间，在多个任务上都有较为平均的好结果。

此外，本文还用到了动态的记忆网络，主要是受到 QA 相关研究的启发：不同的用户有不同的口味偏好，那么对于同一个商品，他们所看到的也是不同的特征。

模型

问题的输入：某个目标商品的所有评论、某个用户的所有评论、某用户的近邻的评论

问题的输出：情感打分（0-5，实数）和个性化的评论

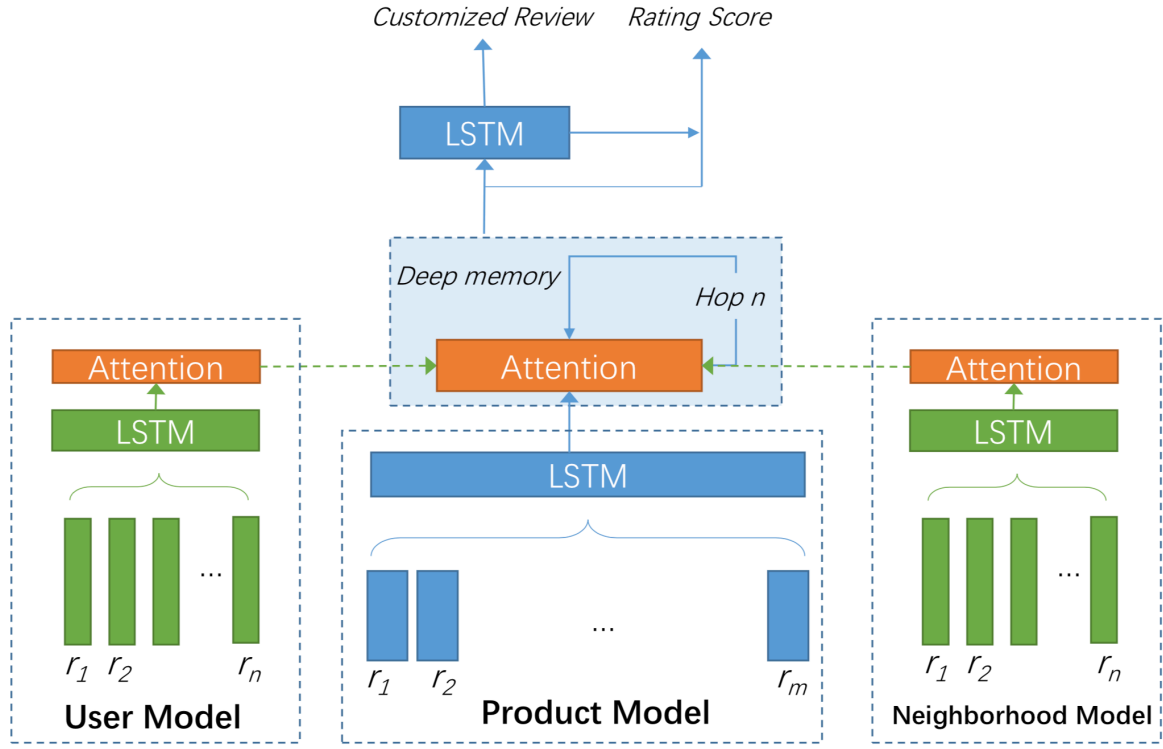


Figure 2: Overview of proposed model.

评论建模

本文的处理是，对于每一条评论就表示为一个向量，具体做法是对该评论中的每一个词的词向量取平均。

用户建模

使用一个 LSTM 对一个用户的所有评论进行抽象，对得到的 hidden state 使用 attention 机制，这是因为不是所有的评论在表示一个用户时的地位或者说是权重是一样的，最终得到一个用户的表示： v_U 。

寻找用户近邻

模型图中显示的 Neighborhood Model 部分，也是根据该用户 u 的若干个最近邻的评论通过 LSTM + attention 得到的，与上述 用户建模 部分是一样的结构。那么这里主要解决的问题就是如何选取用户 u 的最近邻。稳妥采取的方案是基于矩阵分解的方法，即已有的打分矩阵 $M \in \mathbb{R}^{n_t \times n_u}$ (n_t : 商品数, n_u : 用户数) 可以通过 SVD 分解得到用户基于某个隐藏话题的概率分布，根据这个分布，可以计算向量内积作为用户之间的相似度，具体的求解过程如下：

$$\begin{aligned} \min_{F, S, T} \|M - FST^T\| \\ s. t. S \geq 0, F \geq 0, T \geq 0 \end{aligned}$$

这里的 $F \in \mathbb{R}^{n_t \times K}$ 表示的是商品对于预设的 K 个话题的概率分布， $S \in \mathbb{R}^{K \times K}$ 表示每个话题的分布， $T \in \mathbb{R}^{n_u \times K}$ 则是我们需要的用户对于这 K 个话题的概率分布。

下面为了求解 T ，可以使用梯度下降的方法，定义用户相似度 $sim(u_i, u_j) = \sum_{k=1}^K T_{ik}T_{jk}$ ，若两者相似度大于某个预设值，则认为满足邻居关系。

通过这些邻居的评论，得到的近邻特征记作 v_N

商品建模

与上述 **用户建模** 部分是一致的，只是选取的评论是该商品的所有评论。不同的是，此时我们会保留 LSTM 得到的隐状态，而不是通过 attention 得到一个向量。（如模型图中下面中间的框所示）

个性化商品建模

使用一个动态的记忆神经网络，将用户特征 v_U 、近邻特征 v_N 以及商品模型各个隐状态 $h_T = \{h_{T_1}, h_{T_2}, \dots, h_{T_{n_t}}\}$ 融合在一起，做法是典型的 attention Memory Network 模型，模型的输出是 v_C ，它是由 h_T 中的每一个 hidden state 由 attention 定义的加权和： $v_C = \sum_i^{n_t} \beta_i h_{T_i}$ 。类似的，这里运用多跳（hop）的形式，模拟了信息的多层抽象（推理），上一跳的 v_C^{t-1} 影响下一跳的 v_C^t 生成过程

β 即为我们认为的 attention 权重，具体的计算是：

$$u_i^t = \tanh(W_T h_{T_i} + W_C v_C^{t-1} + W_U v_U + W_N v_N + b)$$
$$\beta_i^t = \frac{\exp(u_i^t)}{\sum_j \exp(u_j^t)}$$

这样的生成过程，最终得到的 v_C 融合了用户的个人特征、目标商品的特征。

个性化评论生成

类似 NMT 的 Encoder-Decoder 模型的 Decoder 部分，此处解码的初始状态即为 v_C 。

个性化打分生成

有些意外，他的做法是融合了已有的打分做 attention 加权平均和用 v_C 生成的打分。

$$\hat{s} = \sum_i^n \alpha_i \times s_i + \mu \times \tanh(W_S(v_C \oplus h_{R_n}) + b_S)$$

还有一个小Trick，把在个性化评论生成的最后一个隐藏状态拼接在 v_C 之后，算是做到了不浪费所有的信息。

模型训练与损失计算

打分部分的损失即为传统的 RMSE 损失，个性化评论的生成则是使用最大化词语生成的似然概率：

$$\log P(r|v_C) = \sum_j P(r_j | r_1, r_2, \dots, r_{j-1}, v_C) = \text{softmax}(h_j)$$

这里使用一个 softmax 层计算， h_j 即为 LSTM 解码器端在 j 时刻的隐藏状态。整体的训练过程是联合训练，即同时最小化两部分损失，可以体现出这样设计的优越性：达到两者的互相促进与提高。

实验

实验设置

数据集来自 Yelp.com，是一个学术领域的标准数据集，但是做了一些筛选和训练、验证、测试集的划分。关于生成的个性化评论质量使用 ROUGE-1.5.5 工具评价，指数越高，证明所蕴含的信息越充分；个性化打分使用的 MAE 评价，是因为它对于那些预测偏差大的结果具有更大的惩罚效果（本质和RMSE差不多）。

对照实验分析

本文提出的模型融合了用户的评论、用户的近邻评论，在实验结果中体现出这部分信息都是有正向收益的。

	Rating	Generation
Joint	0.904	0.267
-user	1.254	0.220
-neighbor	1.162	0.245
-user,-neighbor	1.342	0.205
-rating	-	0.254
-generation	1.042	-

Table 2: Feature ablation tests.

表中显示，提出用户评论模块或者是用户的近邻模块，性能都有所下降。

另一方面，本文提出的联合训练方式，也收到了两者协同训练与学习时能互相促进，共同提高的效果。这在模型损失函数的定义中体现在预测的个性化打分与生成个性化评论时的隐藏状态也是有关的，如表中最后两行的结果。

至于其他超参的影响，记忆神经网络的跳数使用 3 层就比较合适，过多则过拟合了，符合经验；预测个性化打分的超参 μ 取 1 时效果最好，表示两部分信息的重要性基本相近。

最终结果

	Rating	Generation
RS-Average	1.280	-
RS-Linear	1.234	-
RS-Item	1.364	-
RS-MF	1.143	-
Sum-Opinosis	-	0.183
Sum-LSTM-Att	-	0.196
Joint	1.023	0.250

Table 3: Final results.

与其他 baseline 的比较，在预测个性打分上，与简单平均、线性加权的结果有显著的提升，显示这样的模型更加精确的抓住了用户的个性化特征。相比基于商品的最近邻、协同过滤（矩阵分解）模型，性能也有显著提升，显示出商品的评论，无论是从用户还是商品的角度，都是有很大影响的。最后，在生成个性化评论，与基于图的观点抽取和传统 LSTM+attention 模型相比，获得提升，表明个性化的用户特征在生成该用户的评论具有重要意义，也同时显示出联合模型的优越性。

结论与我的看法

本文的亮点主要有如下三点：

- 提出了一种新的联合模型，解决既生成个性化评论、又预测打分的多任务学习问题；
- 以多角度方式，从评论中对用户、商品分别建模，灵活运用了 attention 机制；
- 用一个深度记忆网络，完成了多角度信息的融合

与我而言，本文的价值主要在于

- 模型中各处蕴含着 attention，从直觉上能够感受到这样做的意义，虽然文章对于这部分的可解释性阐述不够，每一步做法都很直接，即便看起来这么做也是水到渠成，无可非议。
- 更大的收获在于通过 Memory Network 进行融合，通过 hop 机制完成了层层抽象，这是我之前听关于这方面报告时没有想到的，把每一种视角下的向量表示都看成是 memory，很有新意。
- 这里的联合模型与多任务学习也是我一直在探索的方向，这里取得了较好的效果，得益于损失函数的设计和每一步隐藏变量的生成，需要有直觉上的解释可能才会有效果。