看一看实时相关推荐，满足你对同主题文章的“意犹未尽”

原创 谢若冰等 微信AI 11月26日

导语

在推荐系统中，用户在一个时间段经常会关注同一个主题。当用户读完一篇文章时，他往往会想要继续阅读和这篇文章相关的拓展文章。然而，传统的推荐系统feed流难以提供这种深度的拓展阅读（相关阅读）功能。这是由于考虑到推荐系统多样性和兴趣试探的要求，主推荐流中的文章往往是经过多种推荐/召回逻辑组成的，代表了用户的不同（潜在）兴趣，很少会出现同一个主题的文章连续出现的情况。

在这篇工作中，我们提出了一种新的任务——相关推荐（relevant recommendation suggestion）。这个任务的目标是：

（1）预测用户是否需要相关文章的拓展阅读；

（2）基于用户刚刚阅读的种子文章，实时推荐相关的文章。在看一看系统中，这些模型推荐的相关文章被组织在一个相关box（relevant box）中，并实时插入在上一次点击的文章后面，显式地给用户强烈认知。

这种相关推荐的主要挑战在于：

（1）它在基本的CTR等点击指标外，也需要额外考虑和种子文章的相关性以及信息增益；

（2）相关box的实时插入，也会导致在种子文章之后的本来可能曝光给用户的文章没有曝光。当用户并不想要拓展阅读时，这种实时插入的延迟成本（delay cost）会干扰用户，影响整体推荐效果。

为了解决这些问题，我们提出一种新的实时相关推荐框架（Real-time relevant recommendation suggestion (R3S)），包含了文章推荐（Item recommender）和Box触发（Box trigger）两个模块。具体地，我们从基本特征交互、语义相关性和信息增益等多个维度抽取特征作为专家网络，然后提出一种Multi-critic multi-gate mixture-of-experts (M3oE)模型，基于不同评论家的角度综合考虑这些专家网络的意见，判断是否以及推送什么相关文章。在实验中，我们在看一看系统中进行了离线、线上实验和消融实验，均获得显著提升。在文章维度和box维度的多项提升证明了我们的R3S框架的有效性。目前R3S模型已经上线，影响千万用户。

本文基于WSDM-2021论文《Real-time Relevant Recommendation Suggestion》，论文作者是来自腾讯微信的谢若冰、王瑞、张绍亮、杨智鸿、夏锋和林乐宇研究员。

一、模型背景与简介

随着信息的指数级增长，如何高效获取有效信息成为大众关注点。搜索和推荐是两种重要且互为补充的主动/被动信息获取方式。搜索通过用户主动输入的query理解用户需求，帮助用户主动获取信息。当用户不知道他们想要什么或者不知道怎么搜索时，推荐能够基于用户属性和行为预测用户的潜在兴趣和偏好，帮助用户被动获取信息。

推荐系统中，用户往往在一个时间段只关注一个主题，例如一个新闻事件或者一场球赛。当用户完成一个文章的阅读并退出到主推荐流时，如果对这个文章感兴趣，就很可能会想要主动地获取更多的相关文章进行拓展阅读。为了实现这项功能，推荐系统需要智能地认知到用户当前的实时偏好，并且显式而实时地把高质量的相关文章展示给用户。

然而，在传统推荐系统中，用户只能被动接受推荐内容，很难把自己在相关阅读上的实时偏好主动地反馈给系统（这点搜索中可以通过用户query实现用户-系统反馈）。另外，受限于推荐系统多样性和兴趣试探的要求，推荐系统在进行实时显式的相关推荐时也会投鼠忌器，担心扰乱原始主推荐流中的推荐结果。

在这个工作中，我们尝试赋予推荐系统相关推荐的能力，使用户能够进行深度拓展阅读，提升用户体验和黏性，而一个直观的想法就是参考搜索引擎中的主动信息获取。

在搜索中，query suggestion需要基于用户已经输入的query，预测用户意图，根据语义相似度输出符合用户意图的一组query候选。query suggestion可以看作是在搜索中的相关query的推荐。类似query suggestion，在推荐中，我们提出了一个新的任务——推荐建议（recommendation suggestion），基于用户刚刚点击过的文章，预测用户是否需要相关推荐以及在相关推荐上的偏好，输出合适的相关文章。这个任务可以看作是在推荐中的搜索，用户刚点击过的文章即为query suggestion中的query，可以看做是用户潜意识中给系统的重要反馈。图1给出了在微信看一看中的相关推荐建议系统示意图。

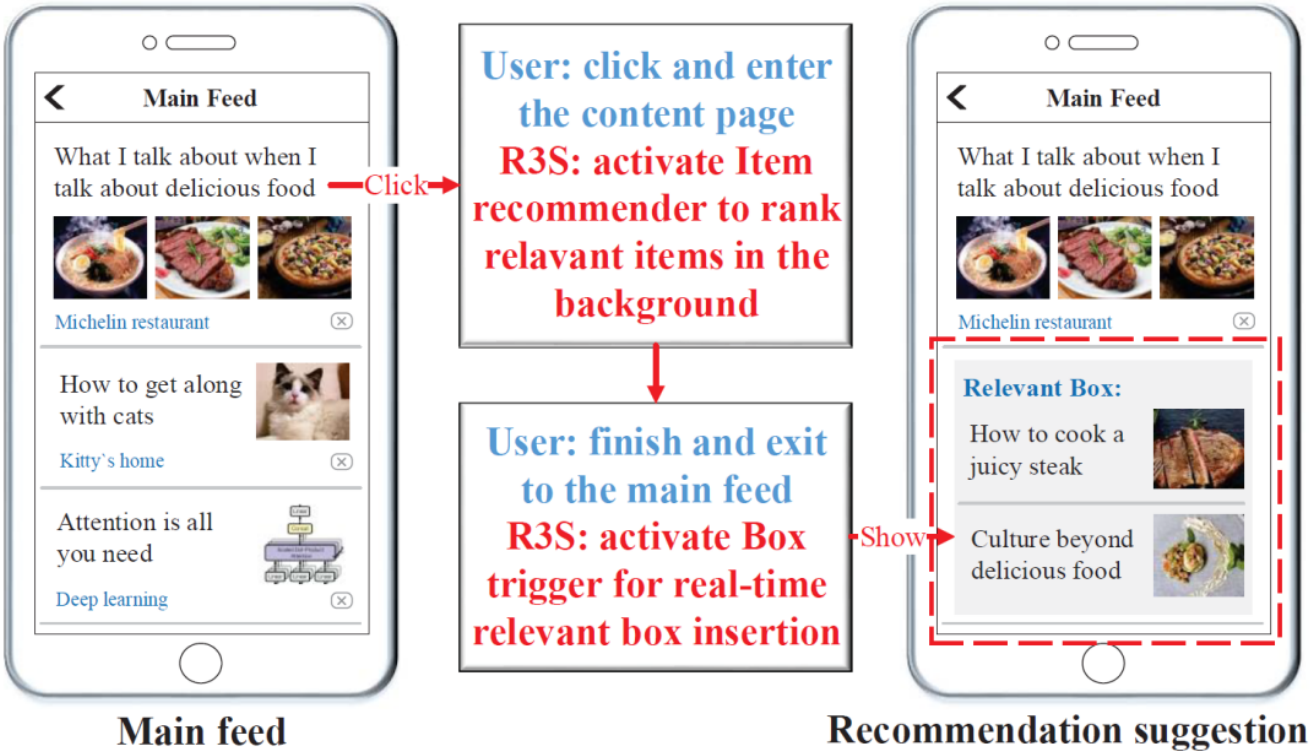

图1：相关推荐系统示意图

右屏展示了实时相关推荐的产品形态和效果。相关文章都被组织在relevant box中，实时插入至紧靠主推荐流里刚点击的文章的后面。

当用户点击文章，读完文章并退出到主推荐流中时，相关推荐模型会计算是否以及哪些相关文章应该推荐给用户。为了提升用户对于拓展阅读的感知，我们把相关文章组织在一个显式的相关box中，同时实时地把相关box插入到主推荐流里用户刚点击过的文章下面。这样，相关推荐框架能够及时地对用户想要拓展阅读的需求做出显式响应，实现拓展阅读功能。

和传统推荐任务不同的是，相关推荐有两个额外的挑战：

（1）相关推荐需要联合考虑多种因素，包括CTR导向的特征交互、种子文章和相关文章的语义相关性和信息增益等。由于不同用户对于不同因素的优先级不同，个性化考虑多因素变得困难。

（2）这种显式实时的相关box插入带来了额外的机会成本。在点击的种子文章之下的本来能曝光给用户的文章，可能会由于这种实时插入导致延迟甚至最终无法曝光（例如图1左侧的猫和attention的文章，在相关box插入后被挤到了更下方）。这种延迟成本（delay cost）在相关推荐模型中也需要考虑，从而使得相关box的实时插入对于整体效果影响尽可能小。

为了解决这些挑战，我们提出一种新颖的实时相关推荐框架（Real-time relevant recommendation suggestion (R3S)），希望能够通过实时插入的方式提供相关推荐功能。

R3S系统包含了文章推荐（Item recommender）和Box触发（Box trigger）两个模块。

文章推荐模块需要基于种子文章（被看作query）召回语义相关的文章，并对他们进行排序。我们提出一种Multi-critic multi-gate mixture-of-experts (M3oE)模型，从基本特征交互、语义相关性和信息增益等多个维度抽取特征构建专家网络，然后训练不同评论家综合考虑这些专家网络的意见。

Box触发模块则通过M3oE模型判断是否应该实时插入相关推荐的相关box。它作为相关推荐的质量检察官，能够避免相关推荐内容的过召回，减少对主推荐流的影响。和文章推荐模块不同的是，Box触发模块还考虑了用户对于种子文章的满意度，以及相关推荐文章实时插入带来的延迟成本。

这些设计解决了以上两项挑战，使得R3S能够同时提升文章和Box相关指标。我们在真实世界的微信看一看系统上进行了大量的离线、线上实验，证明R3S框架的有效性。R3S模型在文章、box和整体指标上均有显著提升。消融实验也证明了模型各个模块的有效性。这篇工作的主要贡献点如下：

1、我们提出一种新的相关推荐recommendation suggestion任务，采用了相关推荐文章实时插入的产品形态，实现了推荐中的拓展阅读功能。

2、我们设计了一种R3S框架，包含文章推荐和Box触发两个模块。我们设计了神经网络表征种子文章和相关文章之间的语义相关性和信息增益的特征，也提出一种M3oE模型，使用不同评论家综合考虑不同专家信息。

3、离线和线上实验中的显著提升证明了R3S模型的有效性。消融实验也证明了模型各个模块的有效性。

4、我们已经将R3S框架部署于微信看一看线上系统，服务千万用户。模型的实用性得到了验证。

二、模型结构

在介绍R3S模型细节前，我们首先介绍一下R3S中一些重要概念：

种子文章（seed）：表示当前用户点击阅读过的文章。

相关文章（relevant item）：相关文章指的是和种子文章语义相关的文章，通常是共享相同的主题或实体。这些文章是相关推荐的候选集。

相关box（relevant box）：相关文章被组织在相关box中（参考图1右屏）。当用户完成种子文章阅读时，如果R3S系统判断应该进行相关推荐，相关box会实时插入在主推荐流的种子文章之下。

延迟成本（delay cost）：相关box实时插入时会导致原推荐流中种子文章之下的文章曝光延迟或者无法曝光。这种曝光损失被称为延迟成本。

R3S系统包含了文章推荐（Item recommender）和Box触发（Box trigger）两个模块。首先，在文章推荐模块中，输入特征被划分成四个特征组，分别是种子文章特征、候选文章特征、用户属性特征以及推荐上下文特征。前两个特征组内拥有相同数量的特征域。具体地，我们使用了

Multi-critic multi-gate mixture-of-experts (M3oE)模型抽取有用的特征，M3oE模型示意图如下图2。

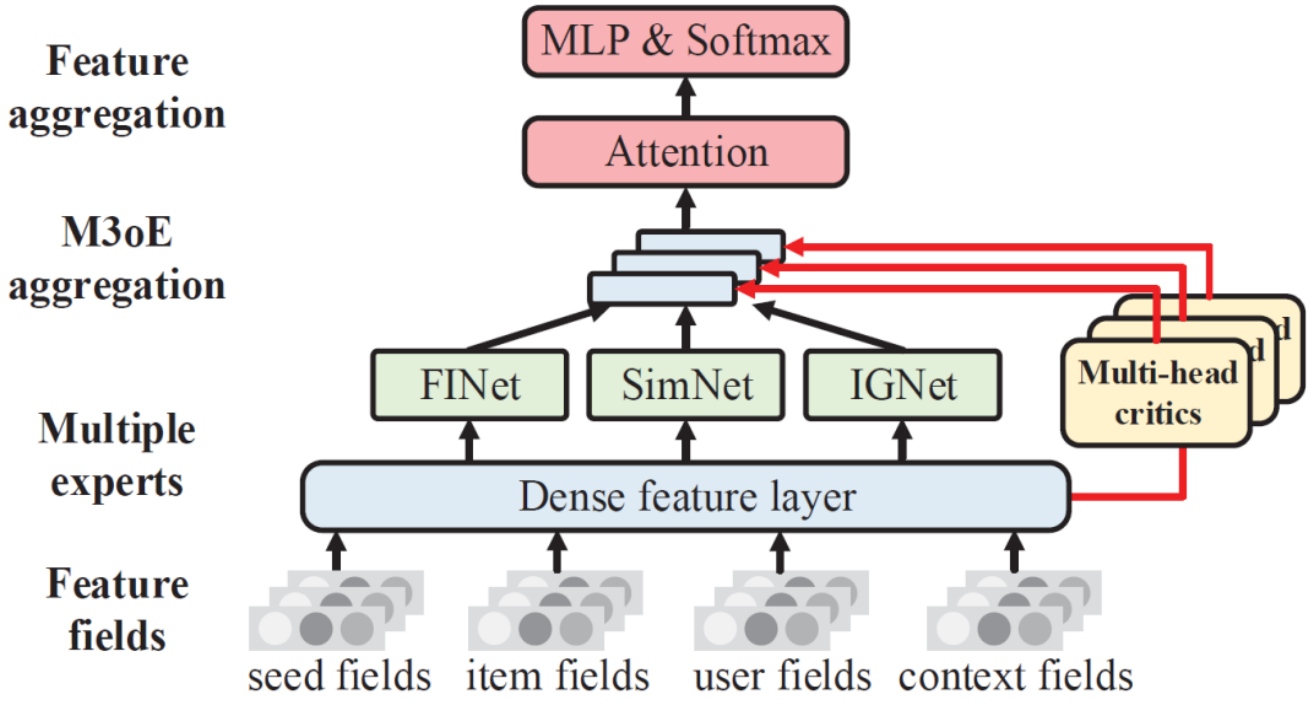

图2：M3oE模型示意图

我们设置了FINet，SimNet和IGNet作为三个专家网络，分别专注抽取（1）基于self-attention的特征交互、（2）种子文章和候选相关文章的语义相似性，和（3）候选相关文章对比种子文章的信息增益。三个专家网络的结构图如图3。

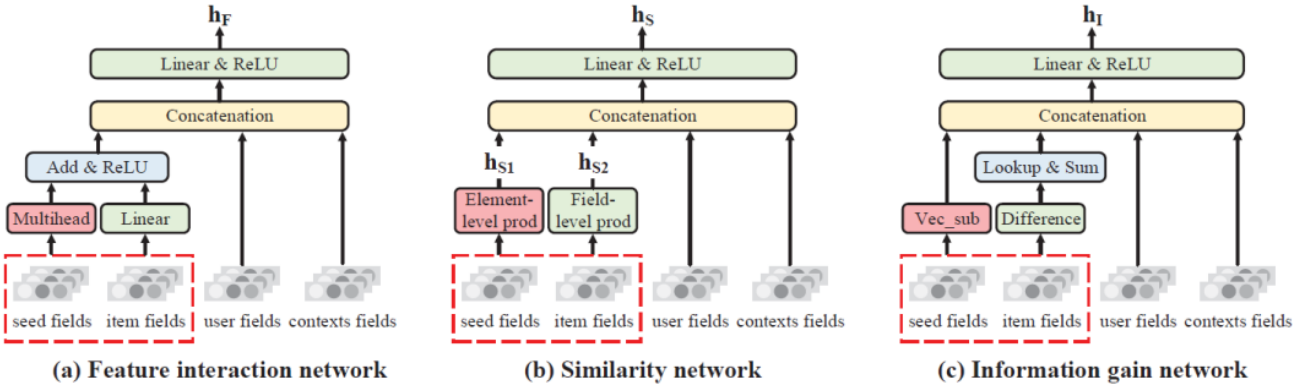

图3：FINet，SimNet和IGNet专家网络示意图

具体地，FINet关注基本特征交互，我们使用multi-head attention计量特征交互：

$$\hat{\mathbf{F}} = \text{MultiHead}(\mathbf{F}) = \text{Concat}(\text{head}_1, \cdots, \text{head}_h)\mathbf{W}_F^O.$$

对于SimNet（负责语义相似度），我们重点关注了种子文章特征组和候选文章特征组之间的相似性，使用向量点乘和按位乘计算语义相关性：

$$\mathbf{h}_{S1} = \mathbf{f}^S \odot \mathbf{f}^I, \quad \mathbf{h}_{S1} \in \mathbb{R}^{kd}.$$

$$\mathbf{h}_{S2} = \{\mathbf{f}_1^S \cdot \mathbf{f}_1^I, \cdots, \mathbf{f}_k^S \cdot \mathbf{f}_k^I\}, \quad \mathbf{h}_{S2} \in \mathbb{R}^k,$$

$$\mathbf{h}_S = \mathrm{ReLU}(\mathbf{W}_S^H \cdot \mathrm{Concat}(\mathbf{h}_{S1}, \mathbf{h}_{S2}, \mathbf{f}^U, \mathbf{f}^C)).$$

对于IGNet（负责信息增益），我们关注了候选文章特征组比种子文章特征组新增的信息，根据特征组的类别设置了两种计算方式。对于类别型的特征域（如用户感兴趣的tag），我们关注种子和候选文章的差集；对于连续型的特征域（如用户年龄、性别等），我们关注特征向量的差。具体形式化定义如下：

$$\mathbf{g}_i = \mathrm{IG}(\mathbf{f}_i^I, \mathbf{f}_i^S) = \begin{cases} \mathrm{Sum}(L(F_i^I - F_i^S)), & f_i^I \in F_{cat}. \\ \mathbf{f}_i^I - \mathbf{f}_i^S, & f_i^I \in F_{con}. \end{cases}$$

$$\mathbf{h}_I = \mathrm{ReLU}(\mathbf{W}_I^H \cdot \mathrm{Concat}(\mathbf{g}_1, \cdots, \mathbf{g}_k, \mathbf{f}^U, \mathbf{f}^C)).$$

对于这些专家的特征聚合，我们参考MMoE模型，设计了M3oE模型。具体地，我们使用multi-head策略产生不同的评论家（critic），然后每个评论家分别对各个专家计算权值，最终进行加权聚合。具体如下：

$$\mathbf{c}_j = g_1^j(\mathbf{x}_j)\mathbf{h}_F + g_2^j(\mathbf{x}_j)\mathbf{h}_S + g_3^j(\mathbf{x}_j)\mathbf{h}_I.$$

$$\mathbf{h}_0 = \sum_{i=1}^{d_c} \alpha_i \mathbf{c}_i, \quad \alpha_i = \frac{\exp(\mathbf{c}_i \mathbf{W}^M \mathbf{f}_a)}{\sum_{j=1}^{d_c} \exp(\mathbf{c}_j \mathbf{W}^M \mathbf{f}_a)}.$$

最后，加权聚合特征加入到具体MSE loss中训练模型：

$$L_{RR} = \frac{1}{N} \sum_{N_a} (y - \mathbf{w}_{RR}^\top \mathbf{h}_f)^2$$

Box触发模块和文章推荐模块类似，也是使用M3oE和三个专家网络进行特征交互。除此之外，Box触发模块在特征计算上，还考虑了用户对于种子文章的满意度，以及相关推荐文章实时插入带来的延迟成本。在最后的loss function中，Box触发模块还在传统交叉熵loss的基础上增加了一项延迟成本相关loss，具体如下：

$$L_{BT} = -\frac{1}{N}(\lambda_p \sum_{N_p} \log p(x) + \lambda_n \sum_{N_n} \log(1 - p(x)) + \lambda_d \sum_{N_d} \log(1 - p(x))).$$

我们还在论文中详细介绍了线上部署细节和我们对于多种可能的实时相关推荐产品形态的思考。

三、实验结果

我们针对相关推荐场景设计了离线和线上实验，基于微信看一看的真实系统对R3S模型的效果进行评测。用户相关数据和行为数据均经过了脱敏处理。图4给出了文章层级的CTR预估结果，图5

给出了相关box层级的CTR预估结果，图6给出了线上实验的结果。我们发现R3S在离线和线上实验中文章和相关box相关指标上都有显著提升。

model	AUC	RelaImpr
FM (Rendle 2010)	0.6949	0.00%
AFM (Xiao et al. 2017)	0.7002	2.72%
NFM (He and Chua 2017)	0.7012	3.23%
Wide&Deep (Cheng et al. 2016)	0.7191	12.42%
DeepFM (Guo et al. 2017)	0.7248	15.43%
AutoInt (Song et al. 2019)	0.7220	13.90%
AFN (Cheng et al. 2020)	0.7294	17.70%
R3S	**0.7419**	**24.11%**

图4：文章层级的CTR预估结果

model	AUC	RelaImpr
FM (Rendle 2010)	0.7658	0.00%
AFM (Xiao et al. 2017)	0.7704	1.73%
NFM (He and Chua 2017)	0.7724	2.48%
Wide&Deep (Cheng et al. 2016)	0.7866	7.83%
DeepFM (Guo et al. 2017)	0.7901	9.14%
AutoInt (Song et al. 2019)	0.7899	9.07%
AFN (Cheng et al. 2020)	0.7982	12.19%
R3S	**0.8101**	**16.67%**

图5：相关box层级的CTR预估结果

model	DT	BCTR	BUHR	BIV
R3S (Item)	+1.68%	-7.62%	+3.14%	+10.68%
R3S (Item+Box)	+1.64%	+9.90%	+16.62%	+24.26%

图6：线上实验结果

四、总结

我们在这个工作中探索了实时相关推荐这个新的推荐任务场景，提出了一套R3S框架解决了实时相关推荐任务，在离线和线上实验中均取得显著提升效果。R3S框架已经部署于微信看一看推荐系统中，服务千万用户。

我们认为相关推荐能够辅助用户进行深度拓展阅读，增加用户时长，提升用户的阅读体验，是推荐系统未来很值得研究的课题。在专家网络设计、专家融合设计等方面，模型还有较大提升空间；对延迟成本的建模也是一个好的研究方向。

微信AI

不描摹技术的酷炫，不依赖拟人的形态，微信AI是什么？是悄无声息却无处不在，是用技术创造更高效率，是更懂你。

微信AI关注语音识别与合成、自然语言处理、计算机视觉、工业级推荐系统等领域，成果对内应用于微信翻译、微信视频号、微信看一看等业务，对外服务王者荣耀、QQ音乐等产品。