# Embedding-based News Recommendation for Millions of Users

Accepted by KDD'2017，作者团队来自日本雅虎新闻，属于将学术成果加以改造使用在产品中，能够适应大量用户的情境。

## 本文摘要

为了做好新闻推荐这件事情，就需要了解新闻的内容和用户喜好之间的关系。协同过滤等基于 ID 的算法局限性在于新闻这一类的对象，保质期很短，难以实时地更新。若利用新闻文本的内容，则是一种类似信息检索的方法，就需要为特定用户从阅读记录中生成个性化的 query。从这两种方法出发，本文提出了一种 3 层结构的基于 embedding 表示方法的新闻推荐方法：为新闻生成表示、为用户生成表示以及生成推荐新闻列表。这一框架在 Yahoo 日本新闻平台上进行了离线和在线测试，在推荐性能和推荐的效率上都取得了令人满意的结果。

## 本文简介

对用户来说，阅读所有的新闻时不可能的，因此需要推荐系统来筛选一些用户偏爱的。各大新闻网站会有一些编辑，精心挑选一些普遍的优秀新闻，却也不可能为每一个用户定制新闻。我们需要一个能够根据用户之前的阅读记录，生成对应的候选新闻来更好地服务用户。

基于 ID 的方法，如协同过滤、矩阵分解模型，具有以下两点缺陷：

- 新闻很快会过期，也会有很多新的新闻产生，这样的模型很难更新，存在冷启动的情况；
- 不能通过新闻的内容对用户进行推荐，太过死板，没有对新闻和用户的喜好做”理解“。

此外，在实际运行的新闻推荐系统中，还需要处理一些噪音数据，也必须适应大规模数据的场景，有较快的响应速度。

常见的一个基线系统的实现，是把新闻看成词语的集合，用户阅读过的所有新闻中包含的词构成的集合为一个用户的表示，通过候选新闻的词集合和用户的词集合之间的重合程度预测用户点击的可能性。这样的系统和模型，结合倒排索引的技巧简单且迅速，取得了一定的效果。因此，雅虎日本新闻选择这样一套系统提供新闻阅读和推荐的服务。但还是存在至少以下两点不足：

- 新闻中的同义词、相似语义的表达不能产生关联，这是词袋模型固有的问题，不同的 ID 则为完全不同的两个词；
- 用户的阅读记录是有时间顺序的，并且对于用户反复阅读的新闻应该有不同的处理。

词向量、深度学习方法在诸多领域都很有效，词向量可以保留词的语义信息而 RNN 擅长处理变长序列化的输入信息。本文提出了一种基于 embedding 的 3 步的分布式表示，从表示新闻，到为用户生成相关的推荐新闻。下面分别介绍新闻的表示、用户表示以及用到的一些技巧，而这里计算用户和候选新闻的得分，采用的是简单的向量内积。最后，给出离线、线上测试的结果。

## 系统流程

本文是依托在实际的新闻推荐系统上的，整个系统的流程如下：

- 识别：从用户的阅读记录中计算用户的特征表示；

- 匹配：计算用户特征与待推荐新闻之间的相关性；
- 排序：根据匹配的程度，对候选集进行排序；
- 去重：对于和用户已读新闻十分类似的，需要剔除；
- 广告：考虑到实际的收益，在新闻中穿插一些广告。

在本文的系统中，所有的新闻只具有 24h 的保质期，过期后则从候选集中删除，并且新闻的存活时间也是计算相关性的一个特征输入。在匹配计算时，新闻的文本长度也是考虑到的一个因素。这些特征工程（非算法相关）都是在实际中总结出的有效策略。对于去重，在实践中验证，新闻的分布式向量表示的余弦相似度如果查过某一个给定的阈值，则认为用户已经读过，会从候选列表中剔除。

# 模型——新闻表示

这一部分主要介绍如何将新闻编码成一个分布式表示。

## 生成式方法

模型中采用一个 DAE，即 Seq2Seq 模型。学习过程中引入了半监督信息，DAE 每次采样一个三元组 $(x_0, x_1, x_2)$，并且基于这样的条件：如果新闻 $x_0$ 和新闻 $x_1$ 是同一个类别，和新闻 $x_2$ 是不同的类别，那么 $x_0$ 与 $x_1$ 的相似度应该要比 $x_2$ 大，体现在中间的隐藏层表示，即 $Sim(h_0, h_1) > Sim(h_0, h_2)$。因此，在最小化损失时，把这部分监督信息也考虑进去了：

$$
\begin{aligned}
\tilde{x}_n &\sim q(\tilde{x}_n|x_n) \\
h_n &= f(W\tilde{x}_n + b) - f(b) \\
y_n &= f(W' h_n + b') \\
L_T(h_0, h_1, h_2) &= \log(1 + \exp({h_0}^{\mathrm{T}} h_2 - {h_0}^{\mathrm{T}} h_1))
\end{aligned} \tag{1}
$$

$$
\theta = \underset{W, W', b, b'}{\arg\min} \sum_{(x_0, x_1, x_2) \in T} \sum_{n=0}^{2} L_R(y_n, x_n) + \alpha L_T(h_0, h_1, h_2),
$$

这里 $f$ 选用 sigmoid，$L_R$ 是经典 DAE 的损失，而 $L_T(h_0, h_1, h_2)$ 则是对不同类别新闻之间应该有较大差距的惩罚项，$\alpha$ 是平衡因子。这里的一个小技巧是，在训练过程中对 $x$ 进行 $p$ 常数级别的衰减，而不是随机衰减，即 $\tilde{x} = (1-p)x$。在这里生成的 $h$ 会用在：计算用户的表示、计算用户和文章的匹配程度、评价新闻之间的相似度。

# 模型——用户表示

这部分介绍了若干种从用户的阅读记录构建用户特征表示的方式。为了便于理解，先引出各个符号的含义：

使用 $a$ 表示新闻的分布式表示，$A$ 为新闻的全集。根据表示方法的不同，简单的可能是基于词典的 one-hot 表示，也可以是上一节中得到的 $h$。

定义 **浏览** 为用户 $u$ 的一个长度为 $T_u$ 的浏览记录：$\{a_t^u \in A\}_{t=1,\cdots,T_u}$。

**会话** 为用户具体的浏览、点击行为。具体而言，会话中包含的是用户 $u$ 在面对当前的推荐列表中，对候选新闻中的某一个新闻进行点击而其他的都被未被点击的行为。从用户 $u$ 在 $t$ 时刻点击新闻 $a_t^u$ 到 $t+1$ 时刻点击 $a_{t+1}^u$ 之间只可能有一个会话过程。这个会话 $s_t^u$ 长度为生成的推荐列表大小（这里设置为 $N$），记录了用户是否点击其中的某个新闻信息（只会有一个新闻被点击，并且成为 $a_{t+1}$），那么这个新闻就是正样本，其他的均为负样本，在模型生成过程中，需要保证用户对于正例新闻 $P_+$ 的得分高于负例 $P_-$。

定义用户的在 $t$ 时刻的表示为 $u_t = F(a_1, \cdots, a_t)$，用户 $u$ 对于新闻的得分函数 $R(u_t, s_t^u)$ 为了提高计算的效率，在每次用户发生新的动作，$u_t$ 都会更新，所以需要很快计算出来，这里就采用了向量内积。

最后，在用户表示模型中，需要确保以下条件成立：

$$\forall s_t^u \ \forall p_+ \in P_+ \forall p_- \in P_-, R(u_t, s_{t,p_+}^u) > R(u_t, s_{t,p_-}^u)$$

可以通过最小化这一损失来实现，小技巧是还加上了一个正例新闻和负例新闻之间的偏置项。

$$\sum_{s_t^u} \sum_{\substack{p_+ \in P_+ \\ p_- \in P_-}} -\frac{\log(\sigma(R(u_t, s_{t,p_+}^u) - R(u_t, s_{t,p_-}^u) + B(p_+, p_-)))}{|P_+||P_-|}.$$

### 基于词典的表示模型（BoW）

这一种方法即为基线模型，用户的表示最终表示为他读过的所有新闻的包含的词的 one-hot 向量。

存在的问题有两点：一是，表示向量较为稀疏，规模也很大，词之间丢失了语义信息，即近义词关系没有体现，必须要完全一致的词，从而损失了很多相关新闻的推荐；二是，用户的浏览顺序被忽略，那些用户反复查看的，频繁阅读带有某一关键词的新闻，没能比那些只读过一次的词语有更高的权重。

### 衰减模型（Dec）

比上面的方法，改进了两点：使用了分布式表示的方法，即 DAE 得到的结果，带有隐藏语义信息；使用了词向量的加权和，且权重根据阅读时间进行衰减。

### RNN 模型

考虑到 RNN 的天然优势，将序列化的新闻浏览记录，压缩成一个隐藏表示，既能把新闻的语义信息融合进来，还考虑到了浏览的顺序。目前常用的 RNN 单元有 LSTM 和 GRU，通过对这两种变体的分析，比较他们在计算 $h_t$ 使用不同的门，发现 LSTM 对于那些很长的输入，会导致 $h_t$ 的无穷范数，即分量的最大绝对值会很大，正比于 $t$，所以会带来梯度爆炸的问题，需要引入 gradient clipping 的方法。

## 离线实验

训练集包括雅虎日本新闻的 12 million 用户，每个用户历时两周的新闻浏览数据。测试数据取自在此之后的数据，并且把其中最后 20 天的用户阅读记录作为测试。评价指标使用 AUC，MRR（平均倒数排名），nDCG，把点击行为视为正例。

**Table 2: Results from offline experiments. Values indicate average of metrics and 99% confidence intervals in ten split test sets.**

| | AUC | MRR | nDCG |
|---|---|---|---|
| *BoW* | 0.582 ± 0.003 | 0.300 ± 0.003 | 0.446 ± 0.002 |
| *BoW-Ave* | 0.579 ± 0.004 | 0.310 ± 0.003 | 0.452 ± 0.002 |
| *BoW-Dec* | 0.560 ± 0.004 | 0.297 ± 0.004 | 0.442 ± 0.003 |
| *Average* | 0.608 ± 0.003 | 0.313 ± 0.003 | 0.457 ± 0.002 |
| *Decay* | 0.596 ± 0.003 | 0.302 ± 0.002 | 0.449 ± 0.001 |
| *RNN* | 0.612 ± 0.004 | 0.309 ± 0.004 | 0.455 ± 0.003 |
| *LSTM* | 0.648 ± 0.004 | 0.344 ± 0.004 | 0.481 ± 0.003 |
| *GRU* | **0.652 ± 0.003** | **0.347 ± 0.004** | **0.484 ± 0.003** |

通过多种方法的对比得出：GRU 是最优的模型，$t$-检验的 $p$ 值小于 0.01。BoW 方式、简单的衰减模型则因为它们各自的缺点，性能不佳。

## 在线实验

### 实验评估

在模型设计完成、离线实验取得性能提升以后，于 2016 年 12 月在实际的新闻服务中上线该系统，其中 1% 的用户仍然通过基线系统给出推荐（BoW）。在线评价的指标，有四个，以前两者优先：

- Sessions，用户每天浏览新闻的平均次数；
- Duration，用户浏览新闻的总体平均时间；
- Clicks，用户在推荐列表中的平均点击次数；
- CTR，平均点击率

对于用户也分为重度、中度和轻度使用者，对后面两种用户的提升更为明显。

**Table 3: Average metric lift rates in 7th week and those by user segments.**

| Metric | ALL | Heavy | Medium | Light |
|---|---|---|---|---|
| Sessions | +2.3% | +1.1% | +1.0% | +1.8% |
| Duration | +7.8% | +4.9% | +13.3% | +17.4% |
| Clicks | +19.1% | +14.3% | +26.3% | +42.3% |
| CTR | +23.0% | +18.7% | +29.8% | +45.1% |

### 上线过程中的挑战

由于是一个实际上线的系统，采用了神经网络模型，出于以下四个原因，不能进行频繁的模型更新：

- 当前部署的模型在 GPU 工作站上训练了一周时间；
- 一旦更新新闻的文本表示（DAE），那么后面的所有模型（用户表示、RNN 等模型参数）都要重新训练和更新；
- 更新用户表示模型，也需要重新训练 RNN/GRU 模型，并且之前的模型信息只能被丢弃；
- 用户的表示信息和新闻的表示信息、用户的搜索信息涉及到的模型参数都必须是同步的。

由于新闻处在不断更新的状态中，3 个月过后这套新的系统性能下降到和基线系统持平，所以每三个月更新一次为佳。

## 总结

本文针对新闻推荐这一比较专精的领域，结合实际的需求和限制条件，提出了一种清晰的基于 embedding 表示的新闻推荐系统，涉及到使用 DAE 进行新闻的表示学习，在此基础上，利用RNN 进行用户特征的表示学习，在最后利用简单且高效的内积方式计算候选新闻和用户之间的关联性，是一次比较好的学术研究和工业系统碰撞的产物。

文中对过去模型的相关总结比较到位，指出了各种传统算法的局限性，并且有（尽可能）对症下药的策略来优化和提升整体模型。此外，作为实现一个金融资讯推荐系统是一个很好的参考借鉴。