实验部分

数据集细节

The samples we used is designed for the online conversational scenarios. We randomly select a part of component services of a mashup as the selected services, and build a positive sample with a label of 1 by combining them with the mashup and a random unselected component service. A negative sample with a label of 0 is composed of the selected services, the mashup and a random service not invoked by the mashup. 负样本规模是正样本的6倍。

For a mashup, we randomly select the selected services multiple times to simulate the situations where developers have choose different component services in their sessions. Take a mashup composed of (a,b,c,d) as an example, we select (a),( b),( ab),( cd), (abc),(acd) as its selected services and build corresponding samples, respectively. The diversified samples empower our model to serve for developers in all circumstances.

对照算法：

现有的大部分方法都针对于the mashup development in static scenarios。我们这里只选取了一些能够在 在线交互场景中工作的对照算法。这些方法涉及到了基于内容，基于协同过滤，混合算法。

* HDP-pop：使用主题模型中的最先进的HDP算法从内容信息中提取特征，然后基于特征相似度和服务的流行程度做出推荐。

The content-based method make a prediction of a service over a new mashup using the TF-IDF-based cosine similarity between their content information.

* *AFUP* (Jain *et al.*, 2015) [11]. Two probabilities that a mashup invokes a service are first calculated according to their content similarity and the historical interaction of neighbor mashups and the service, and then integrated by Bayes’ theorem to . then multiplies them based on Bayes’ theorem, and finally ranks candidates according to their popularity.
* *SFTN*(Samanta *et al.*, 2017)[12]. The authors improve their previous work, AFUP, by using the hierarchical Dirichlet process (HDP) [36] and probability matrix factorization to process the content information and usage history.

首先基于内容信息和历史信息分别计算一个概率，然后基于贝叶斯理论将这两个概率的乘积作为评分的依据。最后，对上个阶段得到的推荐结果，再根据它们的流行度进行一次重排序，得到最后的推荐结果。

其中，在处理历史信息时，把user-CF的思想和PMF的方法结合，使得它具备了为新mashup服务的能力。

* PasRec

它们首先利用了服务推荐场景中的各种对象构造了一个HIN，然后基于两个mashup之间的各种meta-path去计算它们的综合相似度。Then then design a recommendation model based on user-based CF strategy and the overall similarity. They design a pairwise loss function and employ a Bayesian personalized ranking algorithm for model optimization.

* IsRec\_best 提出了一个完整的服务推荐框架，可以为本文的在线推荐场景服务。它的模型在PasRec基础上做出了一些优化。它使用了基于word-embedding的技术，改善了内容信息之间的相似度的计算。它基于主题模型对历史mashup进行预处理，将它们聚类。通过将在线mashup分类，可以加速近邻mashup的查找。

可以发现，CI的效果远远优于HDP+POP,虽然它们都是主要利用了内容信息。这是因为，主题模型本质上是词袋模型，忽略了词序列中的词汇顺序等语义信息，使用它得到的内容特征的质量不高；而我们的CI则使用了深度学习技术提取到高质量的，适应任务的特征。而且基于余弦相似度的相似度很难有效衡量mashup和服务在内容方面的复杂交互。而CI则基于DNN去处理内容的特征向量，捕捉功能方面的交互。

相对于HDP，SFTN在推荐时还融合了历史信息，但是效果提升很小。一方面，它利用历史信息的方式的效果一般。它通过近邻mashup与待测服务的交互估计当前mashup与该服务的交互。但是寻找近邻时只利用了内容信息，限制了相似度计算的效果。而且它基于矩阵分解模型估计交互，但PMF只能捕捉线性关系，在稀疏情况下质量不好。另一方面，它们将基于内容信息和历史信息得到的两个评分相乘的依据是两个概率独立的假设。但是这种假设没有证据，因为具体交互情况复杂未知。而实际上，在我们的混合实例中，在得到两种信息的交互之后，我们使用一个MLP去捕获这种交互。

The reasons are two-fold. First, their feature extraction methods ignored word orders and lost some semantic information. Second, the two probabilities derived from functionality and usage history were multiplied with the assumption that they are conditionally independent of each other. However, it is hard for the single multiplication to capture the way how the content information and usage history jointly affect service recommendation. Instead, MISR aims to capture such complicated interactions by a DNN.

相对于以上两种算法，IsRec\_best，PasRec的效果明显提升。它基于HIN，meta-path和word-embedding等技术更准确的找到了mashup实际的近邻。与这两种可以在在线推荐场景中工作的最好的方法相比，我们的NI使用了相同的历史信息和相似度技术，取得了明显更优的效果，在F1@5, NDCG@5, MAP@5上分别提高了13.1%,22.6%,27.6%。这是因为，它们直接用近邻mashup对待测服务的scalar形式的评分，去估计新mashup对它的评分，而没有深入挖掘它们的隐含潜在的交互关系，这种方式不够精确。而我们的NI实例中，基于node2vec得到的表示已经包含了丰富的交互信息，然后我们设计了基于DNN的交互层去捕捉交互信息。同时，我们还考虑了已选择服务的信息，并且使用attention机制更好地学到了不同服务对待测服务的影响。

而我们的混合交互的实例则在NI实例的基础上更进一步。它同时从内容信息和历史信息，从两种角度衡量m，selected services，s之间的交互，然后基于融合层有效地整合了这两种交互，做出了更好的预测。

这一方面源于我们的基于深度学习的特征提取技术能够得到更高质量的特征表示，另一方面源于。

NI：

我们历史信息空间的模型实例中，设计的基于近邻和node2vec表示新mashup的策略，是创新，效果很好。

混合

深度模型：

~~在 在线服务推荐场景中，可以运用的信息主要是内容信息和历史信息。~~

~~在利用历史信息时，现有的基于深度学习的推荐模型大多将user和item的ID作为输入特征，然后经过embedding层处理之后，得到二者的表示，然后基于DNN学习它们之间的交互。在本场景中，新建的mashup的ID在模型训练阶段没有出现，所以也无法模型服务时无法利用ID信息。我们的NI实例提供了一种处理历史信息的例子，可以得到新mashup和服务在同一空间的向量表示。~~

~~内容信息方面，需要使用特征提取技术处理，得到一个特征向量。~~

深度学习模型可以用于推荐和CTR预测问题。首先基于user和item的各种信息和embedding技术，得到特征向量，然后在此基础上使用DNN学习它们之间的交互。

在 在线服务推荐场景中，可以运用的信息主要是内容信息和*（无历史）*调用信息。为了利用历史信息，现有的大多数模型一般在模型训练阶段将ID，例如mashup ID，服务ID等映射到一个特征向量。但是这种ID+embedding的方式不适合本场景中的新mashup，*它的ID在模型训练阶段没有出现，在serving stage也就无法得到它的ID的embedding。*而为了使用内容信息，也需要使用单独的特征提取技术处理，得到其特征向量。我们的内容信息和历史信息的两个实例，提供了从这两种信息中提取特征的例子。*我们的特征表示策略使得基于文本和历史信息的基于深度学习的在线对话式推荐可行。*

所以，这一部分，我们将我们的从内容信息和历史信息中得到的特征表示，输入到现有的推荐模型的交互层中，最终做出预测。这样一来，一方面可以使得这些深度学习的推荐模型能够应用到这种在线服务推荐场景中，另一方面也可以更直观地比较不同模型的交互方式的优劣。

不同模型的推荐结果如下：

~~相同的特征（相当于backbone）；不同的交互方式~~

~~现有的deepCTR等工作主要针对多种离散特征的情况，离散特征先进行embedding操作，多种特征之间再进行充分的交互。Embedding和交互方式的定义（模型结构）是统一的整体，相互影响。~~

1. ~~我们的特征只有文本和历史信息。文本只能提取特征；对于历史信息，可以使用其ID形式，但是新ID没有embedding，所以只能使用我们提出的表示方法。所以相当于特征已经确定。总之，我们的特征表示策略使得基于文本和历史信息的基于深度学习的在线对话式推荐可行。~~

~~所以使用相同的特征表示，突出交互模式的对比。~~

2.同种特征之间交互即可，对于文本和历史的特征，交互意义不大，反而增加交互学习的难度。

PNCF m和s的信息输入到模型中 相当于不加选择服务信息

Embedding+MLP 加入行为数据，平均池化

DINRec

我们的三种

PNCF:

双塔模型，首先将mashup和待测服务的各种特征整合，然后使用一个MLP学习交互。

1模型部分，也没有考虑已选择服务的信息。

2 全交互。

* *PNCF* (Chen *et al.*, 2018) [14]. The framework compresses all sparse features of users and items in an embedding layer and then uses an MLP to model their interaction. However, its feature extraction component does not apply to extract textual features, and we implement two variants for this scenario: PNCF-HDP, which applies HDP adopted in SFTN, and PNCF-Deep, which uses our feature extraction strategy.

Embedding+MLP 加入行为数据，平均池化

1.没有突出已选择服务的重要性不同。

2. 全交互。各种特征整合后，统一交互。增大了学习难度。见DIN分析。

PNCF借鉴了双塔模型的设计，它首先得到mashup和api的总体的特征表示，然后在交互层使用一个MLP学习它们的交互，最终做出预测。无脑交互。

这里，我们把基于内容信息的实例得到的mashup和api的表示作为它们的内容特征，把基于历史信息的实例中采用的表示方法得到的表示，作为他们的信息的表示。然后两种特征的拼接作为mashup/服务的表示。

我们的效果优于PNCF。因为它没有考虑到用户新选择的服务。同时这种所有特征拼接之后再交互的方式，常用于特征和任务高度相关，基于embedding的模型。而我们这里的两种特征，内容和历史之间，不需要交互，只需要同种信息的交互。

DIN:

1.DINRec中使用的全部是离散特征，需要embedding化，比如ID。在线对话式场景中无法使用。我们设计了特征表示方法。

2.Mashup，已选择服务和待测服务的各种种类的特征拼接，之后做一次attention激活处理。相当于对每个已选择服务对候选只有一个影响权重。当特征比较复杂繁多时，**从各个空间内的多个特征出发考虑的影响可能不一致**，也**增加了交互学习的难度**。我们的混合模型相当于在多个空间都进行了attention，从各个特征空间考虑已选择和待测，mashup之间的交互，增强了模型对交互关系的建模能力。全交互。

3.相对于DIN，我们扩展性更强。当引入新的特征时，只需要在新特征空间学习交互，将交互加入到原来的交互向量，然后重新学习MLP即可。不fine-tuning效果也很好。而DIN则需要整个模型，所有参数更新一遍。

1. 加入已选择的服务的效果（不加，以及几种不同的方式） 在1 2 3上分别对比
   1. 加入有提升
   2. Attention的优势。当选择的服务越多时，相对提升越高
2. 相似度计算方法的对照：
   1. PasRec全
   2. IsRec全
   3. IsRec\_best（实际上只使用了文本和tag特征）
   4. 只使用文本和tag特征：PasRec两条路径
   5. 只使用文本和tag特征：我们的特征
3. 加入已选择信息的，整体的DIN?

Case study

加入了已选择的服务之后，能推荐之前没有hit的next服务；

各个服务的权重不同。Eg:从内容角度。