实验部分

数据集细节

The samples we used are designed for the online interactive scenarios. We randomly select a part of component services of a mashup as selected services, and build a positive sample by combining them with the mashup and a random unselected component service. A negative sample 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.

一对照算法：

现有的大部分方法都针对于the mashup development in static scenarios。我们挑选了一些能够在在线交互场景中工作的对照算法。这些方法涵盖了基于内容的方法，基于协同过滤的方法，混合算法。

* HDP-pop：它首先使用一种非常先进的主题模型——HDP算法，从内容信息中提取特征，然后基于mashup和服务之间的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.

It first uses the HDP algorithm, an advanced topic model, to extract features from the content information, and then predicts the rating of a service over a new mashup based on their HDP-based cosine similarity and the popularity of the service.

* *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mashup调用一个服务的概率，然后基于贝叶斯理论将这两个概率相乘，得到一个综合评分。最后根据这个评分和服务的流行度进行推荐。

*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。

The method first constructs a HIN using various objects in the service recommendation scenario, and then measure an overall similarity between two mashups based on the meta-paths between them. Then they 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 它提出了一个完整可行的服务推荐框架，它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。它在离线阶段将历史mashup聚类，然后在线将新mashup分类，以便加速近邻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。

We can find that CI performs much better than HDP, although the content information they use is the same. The reasons are twofold. First, as a bag-of-words model, HDP ignores word orders in the sequential content information and loses some semantic information, while the CNN based extractor CI employs can obtain high-quality and task-specific features.

It’s difficult for the cosine similarity of content features to measure the complex interaction between mashups and services. CI utilizes a DNN-based interaction layer to accomplish this task.

相对于HDP，SFTN在推荐时还引入了历史信息，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。

Compared with HDP, SFTN also introduces the historical information during recommendation, but the performance improvement is not as big as expected. On the one hand, its strategy to combine the user-based CF and PMF actually makes little use of the historical information. ~~It infers the interaction between a new mashup and a service through that between the neighbor mashups and the service.~~ Only the content information is used when searching for neighbor mashup, which hinders the quality of the neighbors. PMF cannot capture nonlinear interactions and performs weakly for sparse data. On the other hand, the method multiplies the two probabilities obtained from the content information and historical information, with the assumptions that they are independent of each other. But correctness of this assumption is doubtful because the way how these two information affect service recommendation is complex and unknown.

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NDCG@5, MAP@5 F1@5,上分别提高了22.6%,27.6%,13.1%,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。

IsRec and PasRec outperform SFTN significantly. They can find more accurate neighbor mashups by organizing various information using HIN and with the help of meta-path-based similarity measurement and word-embedding technologies. Compared with IsRec, the best method that apply to the online recommendation scenarios, NI utilizes the same information and find neighbors in the same way , but achieved quite better results. The NDCG@5, MAP@5, Precision@5, Recall@5, and F1@5 values of NI were higher than those of IsRec by 22.6%,27.6% and 13.1%, respectively. This is because IsRec directly estimates the scalar score of a service over a new mashup based on the invocation between its neighbor mashups and the service, without digging into the underlying interactions, which is not accurate enough. In NI, the representations of mashups and services obtained by node2vec contain rich interaction information between them. And the interaction is sufficiently exploited by an elaborate interaction layer equipped with MLP and attention block.

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。

Our hybrid example achieves better results than NI. Two kinds of interactions among m, the selected services, and s are first learned in terms of content information and historical information respectively, and then effectively integrated in the fusion layer.

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

二深度模型：

~~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。~~

深度学习模型广泛运用于推荐系统。它们首先基于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。然而在在线服务推荐场景中，这些模型不能直接运用，因为它们很难使用简单的embedding的操作从内容信息和调用信息中提取特征。内容信息需要先进的特征提取器处理才能转化为特征向量。新mashup在模型训练阶段没有出现，模型在serving stage也无法得到它的ID的embedding。而我们的两个实例中的特征表示策略，则使得这些基于深度学习的模型可行。

所以，在这一部分，我们将基于我们的策略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，使它做出预测。这样，可以更直观地比较我们的模型和它们在交互方式上的优劣。

Deep-learning-based models are widely used in recommendation systems. *They first get the feature vectors of users and items by apply embedding techniques to their various information, and then use DNN to learn their interaction.* However, these models cannot be directly applied to online service recommendation scenarios, because it’s difficult for their simple embedding layer to extract *effective* features from the available information, such as the content and the historical invocation. *An advanced feature extractor is necessary to transform the content information into real-value representations. These models could not get the embedding of the ID of a new mashup in the serving stage because it does not appear in the training phase.* And the feature representation strategies in our two examples make these deep-learning-based models feasible in the scenario we study in this paper.

Therefore, in this subsection, we will input the feature representations obtained from our two model examples into the interaction layer of three models and enable them make final predictions. In this way, we can more intuitively compare these models with ours in terms of feature interaction.

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，在这一部分，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

我们选用的深度学习模型主要包括以下几种：

* PNCF:这种双塔模型，首先分别提取和整合mashup和待测服务的各种特征，然后使用一个MLP学习二者之间的交互。

This model is indeed a variant of Deep Structured Semantic Models (DSSM). It first extracts and integrates various features of a mashup and a service, then captures their interaction using an MLP, and finally make a prediction.

* Embedding+MLP：除了新mashup和待测服务，这种模型加入了mashup在线选择的组件服务的信息，并使用平均池化处理所有已选服务的信息。

In addition to mashups and candidate services, this model believes that the information of user behaviors is a complement of user profile and incorporates the selected services into the interaction learning.

* DIN:这种模型首先将mashup和服务的两种特征表示拼接，作为它的整体表示。然后使用attention机制和MLP去学习三者之间的复杂交互。

Considering the diverse user interest, the model designs an attention-based unit to activate a few related interests from various user behaviors. In this scenario, the model employs MLP and attention mechanism to learn the complex interaction among the mashup, the selected services and next service.

通过对比，我们可以发现。这些方法中，PNCF的效果最差，这是因为它的模型中没有加入已选择服务的信息。而这种信息对下个服务的选择有一定影响。Embedding+MLP相对于PNCF，加入了这种信息，效果有一定提升。但是它把所有的已选择服务赋予了相同的权重。DIN相对于这种方法，引入了注意力机制，对已选择服务和待测服务之间的关系更准确地建模，所以效果又有提升。

Compared with other models, PNCF does not process the information of the selected services and performs the worst, indicating that the information has an impact on the selection of next service. The major difference between DIN and Embedding+MLP lies in their methods to integrate the selected services. DIN outperforms Embedding+MLP, demonstrating that each selected service has a different influence on the selection of next service and the attention mechanism can models their complex interaction better.

相对于以上模型，我们的混合模型效果明显最优。这说明相比于DIN，我们的混合模型中对交互关系的建模更有效。

这是因为，DIN（和其他模型）中，首先将Mashup和服务的各种种类的特征拼接，作为整体表示。然后后续的交互建模都在整体表示的层次上进行。

一方面，它使用一个attention block，它综合多种特征的考虑，对每个已选择服务赋予一个综合的影响权重。然而，当对象的多个特征在不同的空间时，**从不同特征的角度考虑的一个已选服务对候选服务的影响可能不一致。DIN的单个attention的机制不能很好的解决这个问题。**而我们的混合模型，在多个空间都应用注意力机制，分别考虑已选择和待测服务之间的各种交互，增强了模型对交互关系的建模能力。

另一方面，DIN是为电商推荐设计的，它考虑了各种特征之间的充分交互。但是在服务推荐的场景中，内容特征和历史特征在完全不相关的空间中，不需要对它们进行跨领域的交互。我们的模型在不同的空间分别学习一种交互，然后最后将它们融合，可以提高交互学习的精准度，降低学习的难度。

Secondly, DIN is designed for e-commerce recommendations and implements the sufficient interaction among various features. However, in the service recommendation scenario, the features extracted from the content and historical invocation are in completely unrelated spaces, so there is no need to combine them across domains. Two internal interactions in the two spaces are learned and fused in our hybrid model, which increases the accuracy and reduces the difficulty of interaction learning.

一方面，DIN的激活单元综合了两种特征表示的考虑，为每个已选择服务对next服务的影响赋予一个整体的权重。然而，基于内容信息和历史信息得到的特征表示在不同的空间内，所以**从这两种特征的角度考虑的一个已选服务对候选服务的两种影响可能不一致。DIN的单个attention的机制不能很好的解决这个问题。**而我们的混合模型在两个空间分别运用注意力机制去考虑已选择和待测服务之间的不同交互，增强了模型对交互关系的学习能力。

It synthesizes the considerations from the content and the historical information

Our hybrid model achieves better results than DIN across all ranking positions. The reasons are twofold. First, the activate unit of DIN measures the impact of each selected service on the next service holistically and assigns an overall weight to an impact. However, the representations of the content and the historical information are in different spaces, so the two effects of a selected service on the next service from the perspectives of these two features may be inconsistent with each other. Our hybrid model applies the attention mechanism to the two spaces and measures the two different impacts respectively, so it has stronger ability to learn complex interactions. *While the single attention in DIN can hardly tackle the problem.*

当然，DIN的处理可能是由于电商推荐中丰富的特征会增大计算负担。但是，在服务推荐领域，针对可用的，有限的，更显著的内容和历史信息，还是我们的交互方式更优。

DIN能够发挥作用的原因，是将离散特征embedding技术和交互方式的结合。这意味着为了效率，交互方式没那么精确，但是embedding技术能够和它互补，生成task specific的embedding。但是服务推荐任务中，基于内容信息和历史信息得到的特征已经相对固定，那么就要求交互方式更智能，能够充分利用已有的特征的特点。

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。

我们的混合模型和DIN的关系:

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。

三 已选服务整合方式的对比

我们的框架中，为了更高效地学习已选服务和待测服务之间的交互，需要先整合每个已选服务的特征，得到一个整体的表示。我们的模型使用attention机制为不同待测服务分配了不同的权重。这里我们使用不同的整合方式替代attention机制，提出了几种模型变体，比较整合方式对模型的推荐效果的影响。

可以发现，几种加入了已选择服务的信息的整合方式的效果要比不加入的效果好。这说明了已选择服务的信息的作用。基于attention的整合方式的效果，是表现最好的。这是因为它能识别到不同选择服务的影响不同，对待测服务和已选择服务之间的关系更好的建模。但是attention相对于average的提升效果并没有预想的突出，这可能是服务推荐领域，mashup调用的组件服务一般不多。

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。