使用深度上下文架构挖掘基于谓词的蕴含规则

郭茂生 张宇 赵德之 刘廷

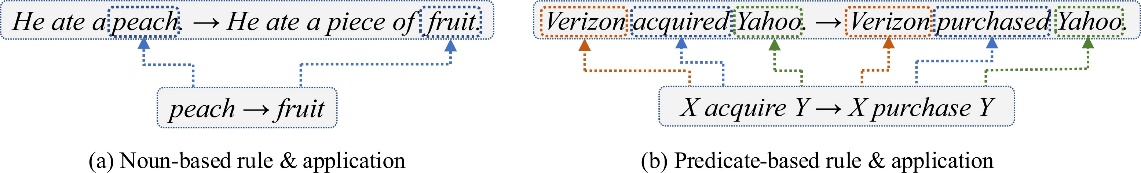
摘要

语义推断在自然语言处理任务中扮演着至关重要的角色，例如问题回答、机器阅读、文本摘要。自然语言中的推断与知识演绎是不可分割的，这通常是基于推导的隐含规则的一种表现形式。到现在，已经有许多工作致力于从文本集中抽取这样的隐含规则，这些工作使用了统计方法学论，包括分布式的假想和隐含狄利克雷分布。然而，这些研究不能平等地考虑覆盖范围以及挖掘规则的准确性，这给下游任务的应用带来了不稳定性。为了解决这个问题，这篇文章提出一个新的模型，叫做深度上下文架构。它由深度置信网络驱动，用于解决从文本中挖掘基于推导的推断规则。处理之前使用过的统计上下文信息，我们也通过将词嵌入到深度置信网络中去学习主题级别的表示来使用语义意义。通过结合两种信息的优势，我们提出的深度上下文架构模型在对谓词上下文进行建模显示处更好的潜力。在公共数据集上的评估展示了我们的方法比其他的强基线有更好的表现

1. 引言

由于推理现象在文本语料库中广泛分布，推理能力在许多NLP应用中起着至关重要的作用。对自然语言推理的研究被称为文本蕴涵，它提高了问答系统、机器阅读、文本照耀、解析器评估和其他NLP相关任务的表现。具体而言，文本蕴涵被定义为两个文本间有方向的的推理关系。给定一对文本片段，T和H，如果H的含义可以从T推断出来，那么T蕴涵H，记为*T* → *H*。例如，“Tom买了一本书“ → ”Tom 有一个书“。

此外，正确地处理推断关系和文本蕴涵知识的积累密不可分，这是日然语言推断的基础。关于蕴涵的知识通过以推理规则形式表示，如图1。例如，“桃子“ → ”果子“是”他吃了一个桃子“ → ”他吃了一个果子“的推断基础。并且”X得到Y“ →”X 购买了 Y“使得”Verizon 得到了雅虎“ → ”Verizon 收购了雅虎“变得合理。第一个例子中的”桃子 → 水果“是基于名词的蕴涵规则，通产才有不包含变量的”下位词 → 上位词“的形式。然而”X 得到Y → X购买了Y“是一种基于谓词的蕴涵模板，这通常包含两个可以更改的位置，例如主语(X)和宾语(Y)。在这篇文章中，我们重点集中与从文本中挖掘后一种形式的蕴涵规则。

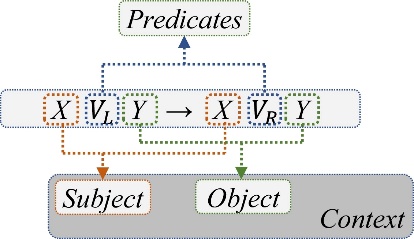


图一

更正式的说，基于谓词的蕴涵规则定义如下：

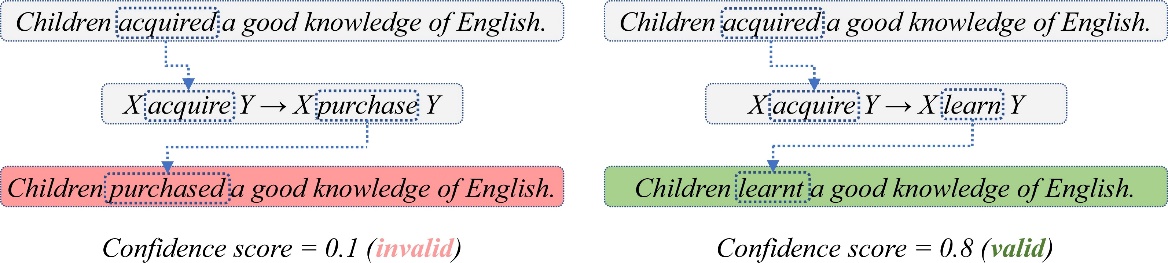
XVLY → XVRY

其中“VL“和”VR “是两个谓词，X是他们的主语，而Y是他们的宾语。元组<X,Y>形成了谓词的上下文，如图2所示。在上下文中填充插槽的单词实例称为填充词，例如，”Verizon“是图1(B)中示例中X的填充词。

图二

值得注意的是，基于谓词的雨涵规则所暗示的推理关系可能并非在所有的上下文中都成立

正如图3所示，我们无法从句子“儿童获得了良好的英语知识“中推断处”儿童购买了良好的英语知识“这一命题，因为规则”X获得Y → X购买Y“咋爱<儿童，英语>的上下文中无效。所以一个可能的蕴涵规则是”X获得了Y → X学习了Y“，即”孩子获得了良好的英语知识 “ → ”孩子学习了良好的英语知识“。这个问题可以通过为每个上下文引入一个置信分数来解决。该分数在3.1节中正式定义。例如”X获得Y → X学习Y“的置信度为0.8且高于其他的规则，暗示着它在上下文中更容易被接受



图三

许多工作已经为挖掘蕴涵规则做出了努力。根据置信度分数是否对谓词上下文敏感，这些工作可以分为两类：上下文敏感方法和上下文不敏感方法。

上下文敏感方法使用隐含狄利克雷分布作为主题模型来检测上下文，这通常比上下文不感知的的方法的准确率要高。然而，他们要求在训练集中能够注意到填充词，导致比上下文不感知的方法的覆盖范围要低。相反，上下文不感知方法可以覆盖更多的应用场景，而对填充词没有限制要求，但是由于缺乏上下文知识，它在规则应用阶段通常获得较低的准确性。换句话说，这些方法不能平等的考虑所提取规则的覆盖面和准确性。

此外，只是用统计信息对于建模谓词的语义意义不够健壮，尤其是在数据源受限而不能覆盖所有的排列和上下文，导致在下游任务应用中的不稳定表现。仔细的说，如果一个填充词从来没有在源预料库中的谓词上下文中出现过，基于LDA的统计模型可能会给出一个低估的置信度。甚者，如果填充词在训练集中没有出现过，那么这些统计模型甚至根本不可能做出判断。

为了减轻不稳定性和获取更好的泛化能力，我们在我们的模型中引入了预训练词嵌入，因为通过神经网络从大量的语料库中获取的词嵌入能够同时获取单词的语义和语法意义，并且被证明能够从单词之间发现关系。另外，拥有数百万个单词的预训练嵌入能够填补训练集和应用场景间单词词汇的空白，实现更高的覆盖率。

深度神经网络已经被应用到了许多NLP任务中，显示出了许多可观的结果。然而，据我们所知，之前没有基于深度学习的方法被提出来且应用带蕴涵规则挖掘任务。在探索各种深度神经网络模型之后，我们选择了深度置信网络作为主题模型来学习任务中谓词的潜在上下文表示。选择该网络的原因有以下几点：首先深度置信网络有HintonHE Salakhutdinov提出来用于解决非线性降维和输入数据重建，非常适合在潜低维主题视角进行谓词上下文的建模。另外DBN可以以无监督的方式进行训练，这适合与蕴涵规则挖掘的任务。最近，DBNS作为主题莫ing在数字公布推荐系统上的表现要由于LDA。在这片文章中，DBNS通过获取统计和语义信息展示出了更好的建模谓词上下文的能力。在公开数据集上的评估展示了我梦的方法在准确率和覆盖标准下优于一些强基线。

这篇文章展示了一种新型的叫做深度上下文架构的模型，有深度置信网络驱动，从文本中挖掘基于谓词的推断规则。我们的主要贡献总结如下

* 提出的DCA模型不仅利用了常用的统计信息，而且利用了词嵌入带来的语义含义，这大大地提高了抽取出规则的准确性
* 我们的方法挖掘的蕴涵信息可以涵盖比基于LDA的方法更多的应用场景，LDA的方法有更严格的限制，即上下文中的所有填充词必须在训练数据集中出现
* 我们展示了使用DBNS作为主题模型队医挖掘基于谓词的蕴涵规则的任务是由帮助的。据我们所知，它是第一个在该任务上有着优秀表现的深度神经网络模型。

该文章的剩余部分组织如下。在第2节中，我们讨论相关的工作。接下来，第三节引入我们的用于解决蕴涵规则挖掘的新方法。第四节给出实验设置细节和进行结果分析。最后，第五节，我们总结该文章。

2. 相关工作

自2001年初依赖，学习如今二提取基于谓词的蕴涵规则已经发展成为NLP中为活跃的研究邻域之一。各种方法已经被提出。Lin和Pantel是第一批研究如何从原始文本中提取基于谓词的蕴涵规则的研究人员。他们提出DIRT（从文本中发现推理规则）方法，该方法基于著名的Harris分布假设。Harris认为，在相同预警出现的单词往往具有相似的含义。Lin等人将这一理论扩展到谓词之间的推理关系。即既有相似语境的谓词有着相似的含义，并且可能相互蕴涵。如果一堆谓词的相似性超过某个阈值，DIT将承认它们之间的蕴涵关系。

实际上，DIRT方法论存在三个缺陷：首先，即使两个谓词的上下文给长相似，两个谓词之间的推理 关系也可能不成立。这一现象通常存在与一对反义词中，例如“X解决Y → X恶化Y“。因此，DIRT优势会提取出蕴涵信息之外具有相似上下文的反义词。其次，DIRT不提供蕴涵信息的方向信息。例如，DIRT不会告诉我们”X买Y → X拥有Y“或者”X拥有Y → X买了Y“。然而，在大多数情况下，语境的蕴涵关系是一种无方向的关系。人工注释表明只有DIRT提取的20%-25%的规则可以双向保真。第三，DIRT在不同情况下对同一规则具有相同的置信。换句话说，DIRT认为提取的规则是普遍使用的，然而事实并非如此，因为我们可以很容易的给出反例，例如，”X得到Y → X购买Y“在<Verizon, Yahoo>的环境下是正确的，但在<儿童, 英语>中却不是。如果我们无区别的使用规则，他将给下游应用程序带来不确定性。

Szpektor等人提出另一种名为TE/ASE(模板提取/锚机提取)的蕴涵规则提取方法，该方法基于自举的想法。TE/ASE也无法提供有方向的蕴涵信息和上下文敏感的置信度。

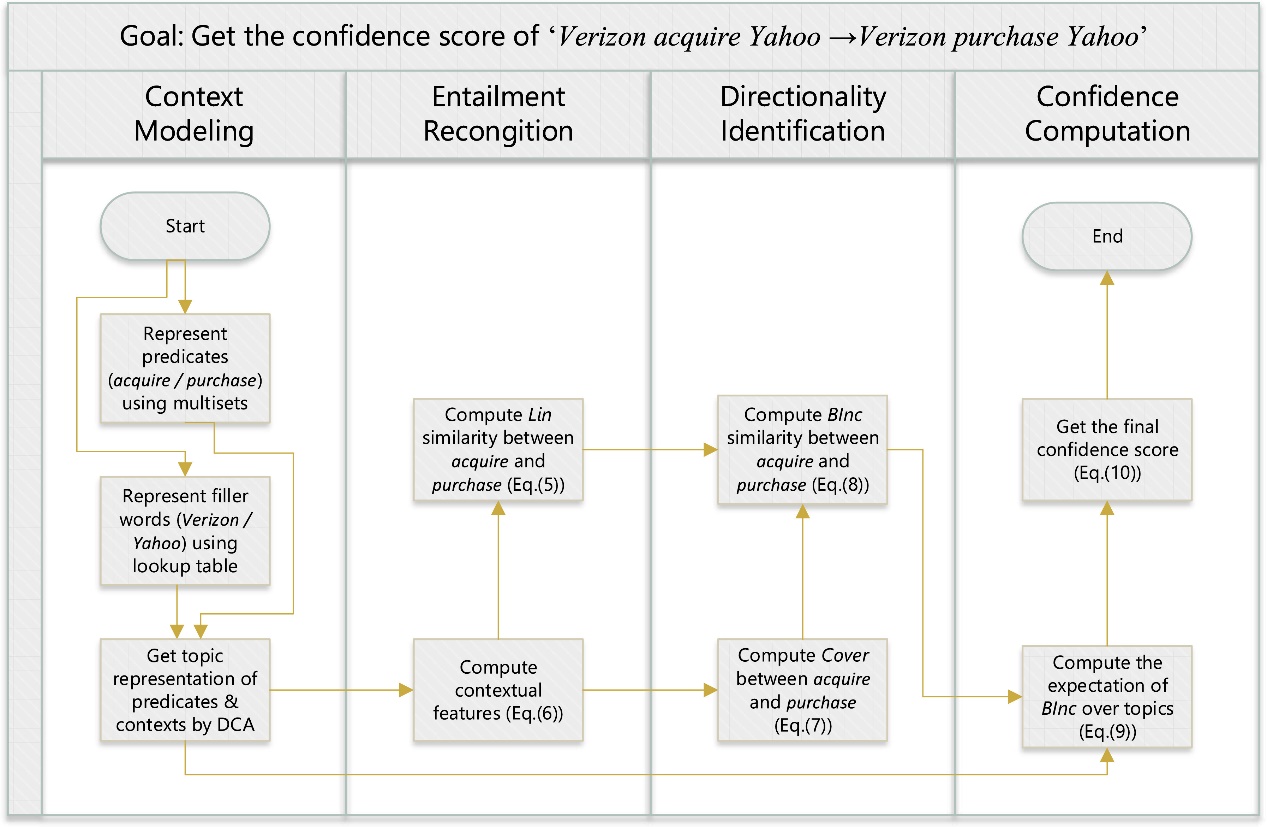
为了学习推理规则的方向性，Bhagat等人提出一种名为LEDIR(学习推理规则的方向性)的方法，该方法使用选择偏好来收集有方向蕴涵信息的证据。Szpektor等人提取一种称为BInc(平衡包含)的方向性相似性度量，来确定蕴涵规则的方向性。

Ritter、Dinu、Lapata和Melamud等人使用LDA(隐含狄利克雷分布)作为主题模型来描述谓词的上下文。通常，他们首先将谓词的上下文是为文档集合，并且使用LDA学习谓词的隐含主题分布。然后，让门利用给定上下文中的两个谓词的主题分布计算出的置信度来识别他们之间的蕴涵关系。这些基于LDA的方法为每个特定的规则应用提供一个上下文敏感的分数，从而比以前上下文不敏感的的方法具有更高的准确性。然而，这些LDA驱动的上下文敏感方法经常在规则应用阶段的覆盖面较低，由于对填充词的限制。

我们的方法收到先前工作。我们重新实现这些推理规则提取方法作为与我们的方法相同的数据集的基线对比，并在4.4小结进行分析。除之前使用的统计上下文信息之外，我们也通过词嵌入使用语义含义，这些语义已经证明能够解释单词之间的关系，从而显著地改善规则挖掘。更多的，我们使用DBNS而不是LDA作为我们的主题模型去学习谓词的隐上下文表示，DBNS在一些任务上的表现要由于LDA作为主题模型。我们的实验也展示了它拥有更好的建模谓词上下文的能力。

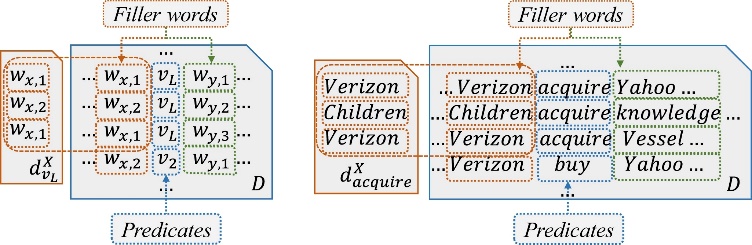
3. 方法说明

在本小结中，我们介绍了DCA模型以及用于基于谓词的推理规则挖掘的相应方法。给定两个谓词，需要解决四个问题才能完成从文本语料库中挖苦额蕴涵规则的任务。首先，我们需要对谓词的上下文进行建模(3.2小节)；第二，我们应该判断是否可以从另一个谓词中推断春一个谓词，即蕴涵关系识别(3.3节)；第三，我们应该识别蕴涵的方向，即那个谓词先出现(3.4节)；最后我们应该给出规则的置信度，即该规则在特定的情况下的使用性(3.5节)。图4通过“Verizon 收购Yahoo → Verizon收购Yahoo“为例演示了四部工作流程，其中每个方框中的子步骤将在下一节进行介绍。



3.1谓词的形式化表示法及其上下文

为了根号的描述我们的方法，我们将使用以下形式的符号：我们的目标是挖掘具有两个变量“X“的基于谓词的蕴涵规则，”XVLY → XVRY“，其中VL和VR表示两个不同的谓词，X是主语，Y是宾语。X和Y形成谓词的上下文用<X,Y>表示(图二)。文本预料库D是推理规则的知识源。多集dXVL表示一个包含所有出现在D中作为谓词主语(X)的词袋，如图五所示。同样的，我们有dYVL dXVR dYVL。例如，dXacquire是所有出现在数据集中作为”acquire“宾语的填充词，例如{Verizon，Children，Verizon，…}。值得注意的是，多集中的每个单词都可能重复出现，并且字序被忽略。此外，如果我们不区分”左“和”右“，即谓词在规则中的位置，我们可以使用单一的符号v来表示谓词。同样的，我们使用符号w来表示可以填充插槽X或Y的单词，或者使用wx和wy区别。此外，我们将谓词的上下文是为在t主题上的概率分布。最后，我们的方法应该给出置信分数F(VL,VR,wx,wy)来应用规则”XVLY →XVRY“在<wx, wy>的语境中。

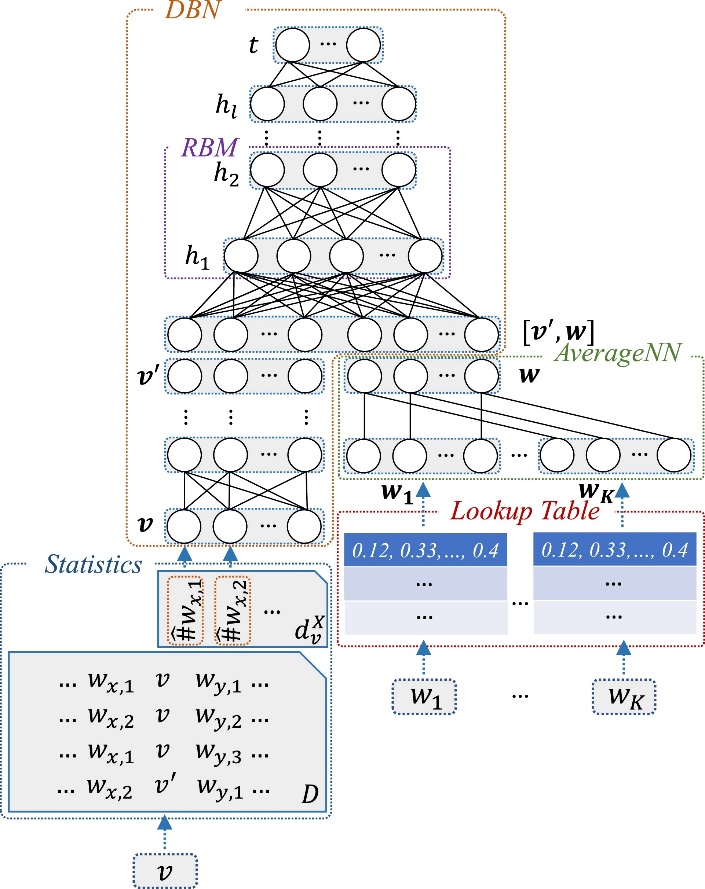


3.2通过深度上下文架构进行上下文建模

一个谓词在不同的上下文可有多个含义。例如，动词“acquire“至少有两个含义，一个用于购买某些东邪，另一个用于学习能力。我们将含义视为谓词的主题。例如，如果单词”acquire“的主题是(购买，学习)，则在<Verizon，Yahoo>上下文中概率分布可能是(0.9, 0.1)，而在<Children，English>的上下文中的概率是<0.2，0.8>。

更正式的说，对谓词上下文进行建模的任务是描述条件概率P(t|v,wx,wy)。由于上下文<wx,wy>的对偶性和对称性，它可以分解为两个子任务，即计算P(t|v,wx)和P(t|v,wy)。并将他们合并获得最后的置信分数。在本小节中，我们首先尝试对只有一个变量(X和Y)的谓词进行上下文建模，如P(T|t,v,w)，它表示给定谓词v及其填充词w的主题t上的概率分布。

DCA模型利用DBN作为其主题模型来计算P(T|v,w)。如图6所示，该模型自下而上由四个不同的部件组成：一个统计部分用来表示谓词v，一个查找表用于获取填充词w的词嵌入，一个AverageNN用于计算填充词的表示，和一个DBN用于计算给定v和w在主题t上的概率分布。



当给统计部分输入谓词v时，它首先提取出在知识源中出现过所有的填充词(Wx,1, Wx,2,…)来形成多集dXV(不失通用性，假设我们认为w来自插槽X)。然后计算每个wx,i的归一化频率，如下：

(1)

其中i，j ∈ 1,…,N并且N时D中填充词汇量的大小。最后，它生成一个固定长度的向量v = (, )去表示谓词v。

同时，查找表用于查找向量**wk**表示每个填充字wk，将其输入到AverageNN中，以获取填充字固定长度的表示如下：

其中K时填充到插槽X的单词数。大多数的插槽由单个单词填充，例如图三中“Children“，同时有些插槽可能由多个单词组成的短语填充，例如”良好的英语知识“，这是在这里引入AverageNN的原因.

如图六所示，DBN由RBM(受限泊尔兹曼机)堆叠而成，其中RBM的每个隐藏层服务于下一个输入层。RBM是一个无方向的基于生成能量的模型，具有输入层，隐藏层以及层之间但不是层内部的连接。例如，h1是输入层h2是RBM的隐藏层，并且h2可以看作是和h1在低维度的重现。

DBNDE 输入是向量**v**和**w**，它们不仅包含知识源中的统计特征，而且包含预训练词向量中的语义信息。由于**v**的维度远远大于**w**的维度，DCA模型首先使用一堆RBMs来减小**v**的维度以生成**v’**，它与**w**有相同的维度。然后将它们连接成一个单一向量[**v’**, **w**]，用来输入到另一堆RBMs来输出在t上的主题分布。最后，我们有：

其中 表示DBN的参数，使用对比发散算法进行训练。

3.3 蕴涵关系识别

为了识别语料库中谓词之间的文本蕴涵关系，我们用DCA捕获的上下文信息对DIRT进行扩展。

DIRT背后的主要思想是分布假设，即具有相似上下文的的两个谓词有着相似的含义并且肯能相互蕴涵。关键是如何去测量这两个谓词之间的相似性。由于一个谓词有两个变量，X和Y，我们可以通过对插槽X和变量Y的相似分数进行集合平均来计算两个变量的相似度。实际上，DIRT采用了Lin相似度。方程(4)用一个变量描述了Lin相似度的定义：

其中是上下文特征函数由 计算得到。是由DCA建模的主题分布。注意，在补充上下文信息之后，不同上下文的相似性会发生变化，换句话说，扩展的Lin相似度是上下文敏感的，这适合于蕴涵关系的抽取。

3.4蕴涵关系方向性识别

容易得出结论，Lin相似度和我们的上下文扩展是关于()对称的。如果，我们交换它们两个的位置，那么相似度测量将会保持不变，即。换句话说，我们不能区别蕴涵规则的方向性只通过使用Lin相似度。

我们从Blin中获取一个想法，利用填充词覆盖面来作为方向相似度：

与Lin相似度对比，是不对称的，且能够捕捉到有方向的信息。然而，它倾向于选择不常见的规则vL。如果一个罕见的谓词vL与其他的谓词vR由常见的填充词，那么其特征的覆盖面通常很高，即使它们之间没有任何的蕴涵关系。因此，只使用也不能有效的提取出蕴涵规则。同vanilla BInc值相似，我们的上下文BInc相似度使用和通过几何均值：

*.*

3.5 置信度计算

到目前为止，给定蕴涵谓词VL和他的填充词w，我们有主题分布，即P(T|vt,w)和每个主题t中两个谓词间的上下文相似度，即，计算P(t|vL,w)对应的相似度数学数学期望值非常直观的。那就是概率加权平均所有可能主题的BInct，如公式(9)所示

*sim(vL, vR, w)* = E*t*∼*P(t*|*vL,w)*[*BInct(vL, vR)*] = sum(t)[*P(t*|*vL, w)BInct(vL, vR)*]*.*

我们只用一个插槽就得到了两个谓词之间的相似度，即*sim(vL, vR, w)*，事实上，这里有两个变量，即X和Y。通过插槽X和Y的对称性。‘*X VL Y*→*X VR Y*’规则在<WX，WY>环境下的置信分数*F*(*VL, VR, wx, wy*)计算如下：

*F (VL,VR, wx, wy )* = *sim(vL, vR, wx )sim(vL, vR, wy ).*

4. 实验和结果分析

进行了比较实验，以评估我们方法的有效性。

在本小节中，我们介绍实验的设计并分析结果来展示DCA驱动方法在推理规则挖掘任务上的优越性。

4.1 数据集构建

4.4.1 训练集

为了评估我们提出的蕴涵规则挖掘方法，我们选择了ReVerb ClueWeb Extractions语料库作为数据源D。它有1500万个元组<Wx，V，Wy>，都是从英文网页上摘取下来的。

为了减少谓词的数量并丰富它们的上下文，我们过滤掉了停止词来规范化谓词。例如句子“X can accommodate up to Y”，“X will accommodate up to Y”，“X accommodate up to Y”中的谓词被认为是相同的，都是“accommodate up to”。另外，我们移除了数量少于30的不同填充词的谓词。

此外，我们也预处理填充词，通过删除停止词，稀有字和非ASCII字符。最后，我们的数据集有6607个不同的谓词和30000不同的填充词。

4.1.2 测试集

测试集是冲Zeichner的数据集中提取的，它包含6567个人类标注的正确的或不正确的推理规则应用，如表1所示。例如，"绝育伤害了我的宠物。绝育伤害了我的宠物→"被注释为"X伤害 Y → X 伤害 Y"的正确注释，而"大多数用户需要使用此功能。→大多数用户需要的不仅仅是此功能"，则"X 需要使用 Y → X 需要超过 Y"的规则被标记为不正确。

| **蕴涵规则** | **相应的应用** | **注解** |
| --- | --- | --- |
| *X 伤害 Y* → *X 伤害 Y* | *绝育伤害了我的宠物*→*绝育伤害了我的宠物* | 正确 |
| *X 需要使用 Y* → *X 需要超过 Y* | *大多数用户需要使用此功能*→*大多数用户需要的不仅仅是此功能* | 不對 |

我们实现了第二节中描述的所有推断规则挖掘方法。由于Zeichner数据集不能覆盖这些方法抽取的所有规则，并且这些方法不能保证抽取数据集中的所有规则，我们只选择它们的交集作为我们的测试数据，有214个正推理规则应用和397个负推理应用组成，共611个实例。形式如下

4.1.3 发展集

虽然我们的方法可以用无监督的方式进行训练，但DCA中存在一些超参数，例如，DBNS的深度和每一层的维度，应该使用发展集进行调优。我们丢弃了交集中存在与测试集中的所有规则，使用剩余的作为发展集，有243个正例和463个反例组成：

4.2 评估指标

为了便于与先前的工作进行比较，我们采用了Melamud等人使用的评估指标，即MAP(平均精度)。我们按照它们的评估设置将测试机随机的划分成30个子集{*Sn*}30*n*=1，并且根据我们的的方法或者基线方法给出的置信分数进行排名，然后计算每组的平均精度，最后计算平均精度的平均值，如下：

其中N=30.

在理想情况下，正例样本的置信分数应该要高于负例。因此，所有正例趴在负例之前。如果存在相反的情况，那么MAP分数将贵下降。总之，MAP适合于蕴涵规则挖掘方法的评估。

出了MAP之外，我们引入Zeichner数据集的覆盖面作为我们测量每个方法适用性的准则：

实际上，Coverage适用Zeichner数据集去模拟实际应用场景来评估方法的泛化能力。英雌，该分数越高越好。

4.3实验设置

为了保证抽取出推断规则的准确性和减少大量的计算，我们只保留每个谓词最有信心的1500个规则。另外，为了减少输入到DCA的向量维度，我们只保留知识源D中频率最高的前30000个填充词来表示模型中统计部分的谓词v。我们适用300维预训练的GloVe单词向量，他在之前的文本蕴涵识别任务中已经广泛适用，作为查找表中填充词w的表示。DCA和LDA的主题数量固定到50，与原来的配置相同。如图7描述了DCA中DBNs的结构，在发展集中进行调优。

4.4 结构分析

4.4.1 DBNs深度的影响

我们首先调整DCA中DBNs的深度以发现深度在该任务上的影响。为了评估我们提出的蕴涵规则挖掘方法，我们选择了ReVerb ClueWeb Extractions语料库作为数据源D。它有1500万个元组<Wx，V，Wy>，都是从英文网页上摘取下来的。

为了减少谓词的数量并丰富它们的上下文，我们过滤掉了停止词来规范化谓词。例如句子“X can accommodate up to Y”，“X will accommodate up to Y”，“X accommodate up to Y”中的谓词被认为是相同的，都是“accommodate up to”。另外，我们移除了数量少于30的不同填充词的谓词。

此外，我们也预处理填充词，通过删除停止词，稀有字和非ASCII字符。最后，我们的数据集有6607个不同的谓词和30000不同的填充词。

4.1.2 测试集

测试集是冲Zeichner的数据集中提取的，它包含6567个人类标注的正确的或不正确的推理规则应用，如表1所示。例如，"绝育伤害了我的宠物。绝育伤害了我的宠物→"被注释为"X伤害 Y → X 伤害 Y"的正确注释，而"大多数用户需要使用此功能。→大多数用户需要的不仅仅是此功能"，则"X 需要使用 Y → X 需要超过 Y"的规则被标记为不正确。

| **蕴涵规则** | **相应的应用** | **注解** |
| --- | --- | --- |
| *X 伤害 Y* → *X 伤害 Y* | *绝育伤害了我的宠物*→*绝育伤害了我的宠物* | 正确 |
| *X 需要使用 Y* → *X 需要超过 Y* | *大多数用户需要使用此功能*→*大多数用户需要的不仅仅是此功能* | 不對 |

我们实现了第二节中描述的所有推断规则挖掘方法。由于Zeichner数据集不能覆盖这些方法抽取的所有规则，并且这些方法不能保证抽取数据集中的所有规则，我们只选择它们的交集作为我们的测试数据，有214个正推理规则应用和397个负推理应用组成，共611个实例。形式如下

4.1.3 发展集

虽然我们的方法可以用无监督的方式进行训练，但DCA中存在一些超参数，例如，DBNS的深度和每一层的维度，应该使用发展集进行调优。我们丢弃了交集中存在与测试集中的所有规则，使用剩余的作为发展集，有243个正例和463个反例组成：

4.2 评估指标

为了便于与先前的工作进行比较，我们采用了Melamud等人使用的评估指标，即MAP(平均精度)。我们按照它们的评估设置将测试机随机的划分成30个子集{*Sn*}30*n*=1，并且根据我们的的方法或者基线方法给出的置信分数进行排名，然后计算每组的平均精度，最后计算平均精度的平均值，如下：

其中N=30.

在理想情况下，正例样本的置信分数应该要高于负例。因此，所有正例趴在负例之前。如果存在相反的情况，那么MAP分数将贵下降。总之，MAP适合于蕴涵规则挖掘方法的评估。

出了MAP之外，我们引入Zeichner数据集的覆盖面作为我们测量每个方法适用性的准则：

实际上，Coverage适用Zeichner数据集去模拟实际应用场景来评估方法的泛化能力。英雌，该分数越高越好。

4.3实验设置

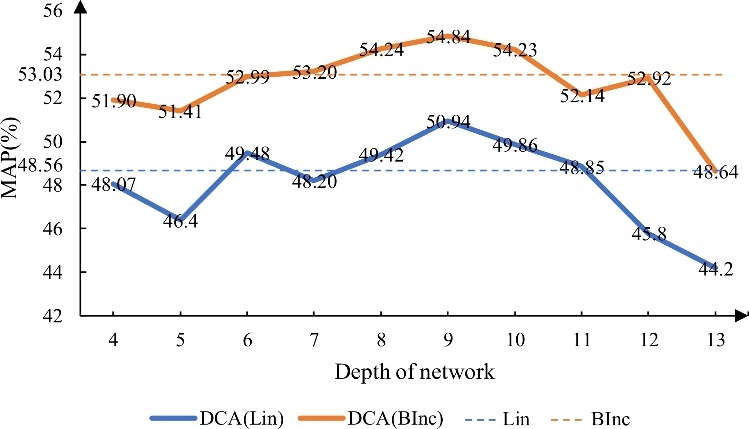
为了保证抽取出推断规则的准确性和减少大量的计算，我们只保留每个谓词最有信心的1500个规则。另外，为了减少输入到DCA的向量维度，我们只保留知识源D中频率最高的前30000个填充词来表示模型中统计部分的谓词v。我们适用300维预训练的GloVe单词向量，他在之前的文本蕴涵识别任务中已经广泛适用，作为查找表中填充词w的表示。DCA和LDA的主题数量固定到50，与原来的配置相同。如图7描述了DCA中DBNs的结构，在发展集中进行调优。

4.4 结构分析

4.4.1 DBNs深度的影响

我们首先天正DCA中DBNs的深度来寻找深度在该任务上的影响。表2展示我们子啊发展集上进行评估的结构，为了找到最好的DSNs深度来对谓词的上下文进行建模，其中第二列的数字表示DCA模型中DBNs每层的维度。加号表示将谓词v和他的填充词w进行拼接。正如表所示，这里的数字设计为统一的减少。例如，在DBNs的第11层，每层的单元数量几乎是前一层的一半。

不同深度的DCA的性能如图[8](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "fig0008)所示，其中橙色实线表示不同深度下具有扩展BInc相似性的DCA*的MAP*，而蓝色实线表示具有扩展Lin相似性的DCA的性能。橙色和蓝色的虚线展示了不同BInc在发展集上的表象和原始Lin相似度。我们注意到DCAs在从7到10比BInc有更高的MAP，其中深度9工作的最好。因此我们固定深度为9，并调整每层的维度，最后得到如图7的结构。具有该结构的DCA在开发集上获得了*MAP*的55.93（具有BInc相似性）和52.13（具有Lin相似性）。如图所示，性能在9 之前随着深度的增长而增加，在添加更多层后性能下降。我们怀疑模型中深度较低的 DBN 部分可能会降低维数的速度太快，无法从 *w* 和 *v* 的表示形式中捕获有用的信息，而如果我们向模型添加太多层，则错误级联的影响将导致性能下降。



4.4.2 定量分析

我们实现了第 [2 节中](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "sec0002)介绍的几种蕴涵规则挖掘方法作为我们的基准。这些方法的测试集上的表现如[图9](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "fig0009)所示（由*MAP*评估）和[10](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "fig0010)（由 *Coverage* 评估），其中 DIRT 表示普通的 DIRT 推理[规则提取](https://www.sciencedirect.com/topics/computer-science/rule-extraction" \o "Learn more about rule extraction from ScienceDirect's AI-generated Topic Pages)方法 [[5]](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "bib0005)，其相似性由方程(4) 计算。DIRT无法识别蕴涵的方向，并且对上下文不敏感。BInc 代表没有上下文扩展的原始 BInc [[9]](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "bib0009) 方法，这是一种方向性感知但上下文不敏感的方法。DC(双调节)[[10]](https://www.sciencedirect.com/science/article/pii/S0925231218311196#bib0010)，SC(单条件反射)[11]，LDA(Lin)和LDA(BInc)[[12]](https://www.sciencedirect.com/science/article/pii/S0925231218311196#bib0012)是几种LDA驱动的上下文相关方法。DC将[等式(9)](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "eq0009)替换为[等式(15），](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "eq0015)这也也无法确定蕴涵规则的方向。而SC用[方程(](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "eq0009)[16)代替方程(9)，](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "eq0016)从而产生方向性感知方法。LDA（BInc） [[12]](https://www.sciencedirect.com/science/article/pii/S0925231218311196#bib0012) 将[方程 （3）](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "eq0003) 替换为 LDA 主题模型。DCA(Lin)和LDA(Lin)都使用*Lint*(*vL, vR*)而不是*BInct*(*vL,vR*)。

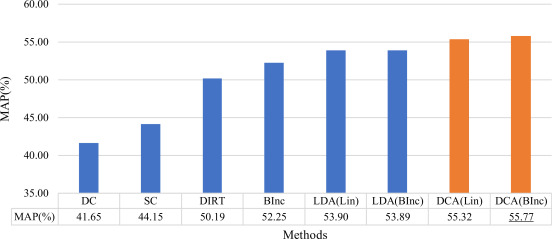


图 9.

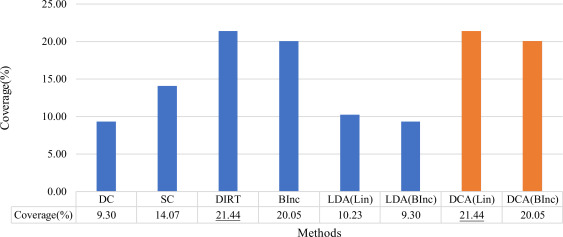


图 10.

从这些图示可以看出，我们基于DCA的推理规则挖掘方法在*MAP*和*覆盖范围*指标下的表现优于在相同数据集下测试的所有基线方法。

我们可以从这些结果中得出以下结论。首先，大多数方向性感知方法(例如BInc，SC，DCA(BInc))比其相应的无方向性感知方法（例如DIRT，DC，DCA（Lin））表现得更好，因为文本蕴涵实际上是一种单向关系。其次，使用LDA或DCA的上下文建模可以提高推理规则挖掘的正确性，因为推理规则并不普遍适用。第三，DCA在该任务中对谓词的上下文进行建模能力比LDA高2%的*MAP*。第四，通过基于LDA的方法提取的规则可以覆盖Zeichner数据集中样本量的两倍以上，这意味着我们的方法具有更广泛的应用范围。

除了DCA与这些基线方法之间的比较外，我们还研究了基于填充词的词向量表示的有效性。该实验的结果如图[11](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "fig0011)所示，其中BInc和LDA是基线，其他DCA(\*)s 表示在各种词向量设置下提出的方法。

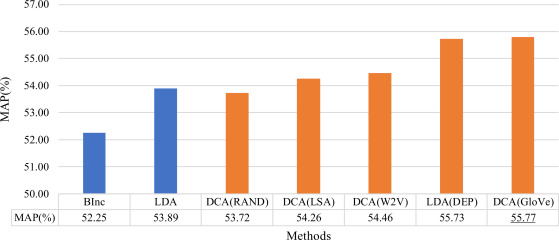


图 11.*不同*设置词向量表示下DCA的MAP（%）。

我们首先从预先训练的词嵌入中去除了所有[语义知识](https://www.sciencedirect.com/topics/computer-science/semantic-knowledge" \o "Learn more about semantic knowledge from ScienceDirect's AI-generated Topic Pages)，方法是根据高斯分布用随机初始化的词向量给DCA，以评估DBN本身的上下文建模能力，结果为53.72（MAP%），高于原始的BInc方法，与LDA竞争，并考虑到引入的噪声。这种现象证明了DBN在DCA模型中的有效性。接下来，我们探讨了不同词向量的性能。通过使用Indra [[23]](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "bib0023)的架构，我们获得了在维基百科上训练的三个词嵌入，即LSA（潜在语义分析[[24]](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "bib0024)，这是使用word-document矩阵上的[奇异值](https://www.sciencedirect.com/topics/computer-science/singular-value" \o "Learn more about singular value from ScienceDirect's AI-generated Topic Pages)分解获得的），W2V（Word2Vec [[25]](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "bib0025)，使用连续词袋和跳过语法架构训练）和DEP（基于依赖关系的词嵌入[[26]）。](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "bib0026)这与 W2V 类似，但上下文是根据依赖关系[解析](https://www.sciencedirect.com/topics/computer-science/parsing" \o "Learn more about parsing from ScienceDirect's AI-generated Topic Pages)树定义的）。尽管 LSA 在这些预训练的词向量中获得了最低的 MAP 分数，但它仍然比 LDA 和 RAND 都表现得更好，这表明了来自词向量的语义知识的有用性。与W2V相比，DEP涉及额外的依赖信息并获得更好的性能，甚至与GloVe相当，GloVe被证明能够在[自然语言](https://www.sciencedirect.com/topics/computer-science/natural-languages" \o "Learn more about natural language from ScienceDirect's AI-generated Topic Pages)推理任务中捕获蕴涵关系[[20]](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "bib0020)，[[21]](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "bib0021)，[[22]](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "bib0022)。

总之，我们基于DCA的方法的优越性来自三个方面：首先，DCA[的网络结构](https://www.sciencedirect.com/topics/computer-science/network-structure)，其中DGN学习谓词及其上下文的潜在低维主题表示，经过精心设计，从数据集的统计信息和存储在预训练的词向量中的语义知识中获取输入。其次，由于基于词嵌入的填充词表示，我们基于DCA的方法可以克服LDA驱动方法的限制，即所有填充词都必须出现在训练集中。第三，由分布假说（即BInc）驱动的方向性感知方法从文本中捕获蕴涵现象。综上所述，所提方法作为一种具有高准确性的上下文感知方法，保留了上下文感知方法的高覆盖率优势。

#### 4.4.4定性分析和误差分析

除了上述定量分析外，我们还研究了提取的规则，特别是我们的方法所犯的错误。其中一些列在[表3](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "tbl0003)中，其中假阳性意味着示例被Zeichner标记为不正确，但获得了相当高的置信度分数，而假阴性表示标记为正确的示例，置信度分数非常低，而真阳性和真阴性表示我们的方法报告的置信度与Zeichner的注释一致。

表 3.正确预测以及假阳性和[假阴性错误](https://www.sciencedirect.com/topics/computer-science/false-negative-error" \o "Learn more about false negative errors from ScienceDirect's AI-generated Topic Pages)的示例。

| 蕴涵规则 | 相应的应用 | 评论 |
| --- | --- | --- |
| *X 击败 Y* → *X 击败 Y* | 理查德打败了马库斯。→理查德击败了马库斯。 | 真阳性（释义） |
| *X 加入 Y* → *X 成为 Y 的成员* | 安德森加入美国空军。→安德森是美国空军的一员。 | 真阳性（推理） |
| *X 射击 Y* ↛ *X 投掷 Y* | 詹娜射杀弗兰克。↛ 詹娜扔弗兰克。 | 真正的否定（无关紧要） |
| *X 影响 Y* ↛ *X 干扰 Y* | 法官影响结果。↛ 法官干预结果。 | 真负数（向后蕴涵） |
| *X 说 Y* ↛ *X 不说 Y* | 工作人员会讲英语。↛ 工作人员不会说英语。 | 误报（否定） |
| *X 对 Y 更好* ↛ *X 对 Y 更好* | 大麻对我们的星球更好。↛大麻对我们的星球有害。 | 误报（反义词） |
| *X 需要保持 Y* ↛ *X 得到保持 Y* | 微软需要留住开发人员。↛ 微软可以留住开发人员。 | 误报（预处理） |
| *X 油漆 Y* → *X 现在 Y* | 一幅画画画千言万语。→ 一张图片呈现千言万语。 | 假阴性（隐喻） |

大多数提取的蕴涵规则是可解释的，例如，"*X*击败*Y*→*X*击败*Y*"，以及推理，例如，"*X*加入*Y*→*X*是*Y*的成员"，这在[表3](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "tbl0003)中表示为"真阳性"，而真负样本由不相关组成，例如，"*X shoot Y* ↛ *X throw Y*"和向后蕴涵， 这意味着两个谓词是相关的，但蕴涵关系只存在于向后方向，例如，"*X*影响*Y* ↛*X*干扰*Y*"，但"*X*影响*Y* ←*X*干扰*Y* "在<判断，结果>上下文中*，*。这些样本表明，我们的方法能够捕获蕴涵关系并有效地检测其方向。

许多误报错误来自反义词或谓词，具有相反的含义，但具有共同的上下文。例如，"说话"和"不说话"（[表3](https://www.sciencedirect.com/science/article/pii/S0925231218311196#tbl0003)中的第一个示例）具有几乎相同的上下文和相似的统计表示形式。因此，它们具有相似的*P*（*t*|*v, w*）， 因此有很高的置信度得分 *F*。对于所有基于分布假设的方法，这是不可避免的，它认为相似的上下文导致蕴涵关系。我们计划在今后的工作中解决这个问题。

此外，[第 4.1 节中](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "sec0010)引入的谓词规范化导致了另一个罕见但存在的误报错误，例如[表 3](https://www.sciencedirect.com/science/article/pii/S0925231218311196#tbl0003) 中的第七个示例。"need to keep"和"get to keep"都规范化为“keep"，以丰富上下文的多样性，从而更好地进行置信度计算。事实上，我们试图摆脱谓词规范化，但由于缺乏上下文的多样性，它导致了性能下降。这个问题可以通过补充更多的培训语料库来解决。

假阴性错误可能是由谓词的隐喻或谓词的扩展用法引起的。例如，" *X paint Y* → *X present Y*"在大多数情况下都不是真的，除了在*< a picture, a thousand words >* 条件下。因此，这些谓词的上下文可能会有很大差异，从而导致置信度分数较低。

**5. 结论和今后的工作**

在本文中，我们提出了一种名为Deep Contextual Architecture（DCA）的基于DBN的新型模型，以从文本语料库中挖掘蕴涵规则。一方面，我们的方法结合了来自预先训练的词向量的语义意义和来自文本语料库的统计信息，这极大地改进了推理规则挖掘。另一方面，通过利用DBN的低维数据表示能力，我们基于[神经网络](https://www.sciencedirect.com/topics/neuroscience/neural-networks" \o "Learn more about neural networks from ScienceDirect's AI-generated Topic Pages)的方法通过MAP和 Coverage指标实现了这项任务的最先进的性能。对公开可用数据集的评估证明了我们在蕴涵规则挖掘方面的优势。

同时，该模型仍有改进的空间。首先，DCA使用 AverageNN（[图 6](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "fig0006)）来表示包含多个标记的填充词，这些填充词可以用预先训练的短语或句子级 LSTM 编码器（如 [[27]）](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "bib0027)替换。其次，可以将注意力机制纳入该模型，例如，自注意机制可能有助于对包含多个标记的谓词或填充词进行建模，而谓词与其上下文之间的相互注意可以更好地建模它们之间的交互。第三，将DCA挖掘的单个推理规则合并到知识图谱中，并使用与[[28]](https://www.sciencedirect.com/science/article/pii/S0925231218311196" \l "bib0028)类似的导航机制，根据它推断出更多的规则，这是合理和可能的。最后，还存在一些挑战，例如当谓词的上下文相似时，如何区分反义词和蕴涵。我们计划在未来沿着这些方向进行研究。