TabEL: Entity Linking in Web Tables

Chandra Sekhar Bhagavatula(✉), Thanapon Noraset, Doug Downey

Northwestern University, Evanston, IL 60201, USA
{csbhagav,nor.thanapon}@u.northwestern.edu, ddowney@eecs.northwestern.edu

摘要. Web表格中是关系型数据的一个很有价值的来源。Web包含大约1.54亿张关系型数据的HTML表格，仅Wikipedia就包含160万高质量的表格。抽取Web表格中的语义以生产机器可以理解的知识已经成为了一个热门的研究领域。

抽取Web内容中的语义的关键一步是实体链接 (EL)：任务就是将文本中的短语映射到其在一个知识库 (KB) 中的对应实体。在本文中我们提出了 *TabEL*，一个新的Web表格实体链接系统。*TabEL* 和过去的研究工作不同之处在于削弱了一个假设，这个假设认为一个表格中的语义能够映射到目标知识库中预先定义好的类型和关系。而，*TabEL* 将软约束条件施行在一个图模型上，这个图模型把高可能性赋给那些可能在 Wikipedia 文档和表格中同时出现的实体。在实验中，*TabEL* 跟当前最新的实体链接系统相比显著地降低了错误，包括在 Wikipedia 表格上减少了75%的错误和在 Web 表格上减少了60%的错误。我们同时也做了处理过的 Wikipedia 表格语料库和公开的开放的测试数据集用于将来的工作。

关键词: Web表格 · 实体链接 · 命名实体消歧 · 图模型

1 介绍

Web 表格 (*tables*)，换句话说 Web 上的 HTML 表格，是关系型数据的一个重要来源和信息抽取 (IE) 系统的一个重要输入。据估计 Web 上有超过141亿表格，1.54亿张表格包含关系型数据并且仅 Wikipedia 就是大约160万关系型表格的来源。不像普通文本，单张关系型表格就包含一系列高质量的关系实体和相关的元数据 (在表格的列头中)。Web 上关系型表格中蕴含的财富和实用性使得表格的语义解释 (*semantic interpretation*)，也就是将 Web 表格转换成机器能读懂的知识这一任务成为热门的研究领域。

抽取 Web 内容语义的关键一步就是实体链接 (EL)：将文本中的短语映射到给定知识库 (KB) 中其相对应的实体。举个例子，在表格 1，实体链接任务是链接在第二列的 "Chicago" 链接到其在知识库 (例如 YAGO) 中对应的实体 **Chicago** (城市)。短语的一词多义现象是 EL 系统面临的主要挑战。EL 系统必须对每一个给定的短语利用周围内容 (成为该短语的上下文) 中的线索来进行消除歧义。在表格 1中，"New York" 一词多次出现，但从上下文可以清晰地看出，它在第二列中指的是城市在第三列中指的是州。

我们提出 *TabEL*，这是一个在 Web 表格单元格中的短语上执行实体链接 (*Entity Linking*) 任务的系统。现有的表格语义解释系统通常采用图形化模型来对三个语义解释任务进行联合建模：实体链接 (*entity linking*)，列类型识别 (*column type identification*) 与关系抽取 (*relation extration*)。这样的联合模型

Table 1. 表格中包含一些美国的最高楼和其所在的城市和州。下划线表示知识库中的实体存在参考对象。

Building Name	City	State
One WTC	New York	New York
Willis Tower	Chicago	Illinois
⋮	⋮	⋮
MetLife Tower	New York	New York

是基于一个强大的假设的，该假设就是列的类型和表中表达的关系能够被映射到目标知识库中预先定义的类型和关系。因为由表格的结构传达的类型和关系信息是 EL 任务有价值的线索，依赖严格的映射到一个 KB 是很可能出错的因为知识库是不完整或者嘈杂的。

在本文中,我们研究了一个方案来替代严格映射到一个 KB。*TabEL* 通过图形化模型的软约束条件整合了类型和关系信息。这些约束条件对那些“连贯” 的参照实体集合进行编码偏好，这些“连贯”的实体往往同时出现在维基百科文档和表格中。虽然我们的图模型是紧密连接的(见第三节)，我们发现在实验中，我们可以使用迭代分类算法 (ICA) 更容易地达到消歧的目的。在实验中，我们发现 *TabEL* 比以前的工作更准确，在基准系统的 Web 表格上减少误差60% 。*TabEL* 在维基百科表格上表现尤其优异并且减少误差75%相对以前的工作来说。在消融研究中，我们分析了 *TabEL* 组件对精度的影响，证明了我们选取的特性导致了选择最常见短语意义作为其对应实体的系统提升了12%的准确率。

最后，我们公开了我们的包含超过160万个维基百科表格的语料库。我们也让包含已标注的实体的数据集维基百科和 Web 表格公开来供未来的表格 EL 系统使用。

2 序言

表格的语义解释的一般任务使用一个表和参考知识库 (KB) 作为输入，通常包括以下子任务：

1. 实体链接 (EL)：找到表格单元格中称为 *mentions* 的文本短语并将其与对应的候选实体关联
2. 列类型识别：将表格中的一列与其包含的实体的 KB 类型相关联
3. 关系抽取：根据两个表中列与列在给定的一行的实体对的关系将列们与知识库中的关系相关联

参照实体，类型和关系都是建立在给定的 KB 中。举一个具体的例子，给定表格 1和知识库 YAGO，实体链接的任务之一就是将 “Chicago” 链接到知识库中的实体 Chicago。类型识别包括将表格第二列与知识库中的城市类型链接。关系抽取任务将包括识别实体 Willis Tower 和 Chicago 之间的关系 isLocatedIn。

在这篇文章中，我们专注于第一个语义解释任务，实体链接。我们现在正式定义表格的实体链接任务。我们还会介绍将用于本文的其余部分的符号。

正式定义

一个潜在的指称 (*potential mention*) 是文本中的一个短语，其对应的参考实体在给定的 KB 中是未知的。我们表示一个基于一个短语 s 潜在的 mention 为 $m_{s,?}$ (? 表示一个未知的实体)。一个已标注的指称 (*annotated mention*)，是一个短语其参考实体是已知并且被表示成 $m_{s,e}$，s 是文本短语其对应的参考实体是 e。

Web 上的一个表格被表示为一个矩阵，T，该表格包含 r 行 c列。使用行和列单位的表格很容易规范化成 $r \times c$ 的矩阵。$T[i,j]$代表了 T在 i^{th} 行和 j^{th} 列的单元格。

任务： 给定一个表格 T和一个知识库 $\mathcal{K}$ 的实体，实体链接 (*entity linking*) 的任务是识别 T 的单元格中每个潜在的指称并将其链接到其对应的参考实体 $e \in \mathcal{K}$。

3 系统描述

给定一个表格 T 和一个知识库 $\mathcal{K}$，*TabEL* 执行实体链接任务分为三步：

1. 指称识别 (*mention identification*)：在 T 的单元格中识别出每个潜在的指称，$m_{s,?}$
2. 候选实体生成 (*entitycandidategeneration*)：对于每个潜在的指称 $m_{s,?}$，生成其候选实体集合，$\mathcal{C}(m_{s,?})$ - 知识库 $\mathcal{K}$ 中的实体子集并且其中的实体都可能是 $m_{s,?}$ 的参考实体
3. 消歧 (*disambiguation*)：对于每个潜在的指称 $m_{s,?}$，从 $e \in \mathcal{C}(m_{s,?})$ (其候选实体集合) 中挑选出一个实体，根据指称的上下文作为 $m_{s,?}$ 的参考实体

TabEL 使用的是监督学习的方法，并使用表格中已标注的指称来训练其各个组成部分。就像大多数实体链接系统一样，*TabEL*也基于一个先验概率，那就是一个给定的字符串 s 指向一个特定的实体 e 的概率，i.e. $P(e|s)$。在先前的工作中，我们通过 Web 和 Wikipedia 上的超链接来估算这个概率分布 $P(e|s)$，就像在第4节中描述的那样。

尽管我们在实验中使用 YAGO 作为我们的知识库，但我们的方法是通用的并且可以使用任何知识库，只要为该知识库给定一些已标注的例子和一些在我们的系统中使用的合适的实体相似度的测量值。

3.1 指称识别

任何 EL 系统的第一步是找到潜在的指称，可以链接到知识库 $\mathcal{K}$ 中他们的参考实体。鉴于文本内容，t_q，来自输入表格的每个单元格，*TabEL* 将最长的短语识别为潜在的指称，t_q 中的 s 对于一些 e 来说有非零的概率 $P(e|s)$。如果 s 的长度小于 tq 的长度，*TabEL* 会在 s 后面发现长度最长的短语，并以此类推。例如，对于一个单元格的文本 “Barack Obama & Mitt Romney”，*TabEL* 会找到两个潜在的指称：一个是 “Barack Obama” 和另一个是 “Mitt Romney”。

3.2 候选实体生成

对于每个潜在的指称，$m_{s,?}$，*TabEL* 给指称设置了候选实体集合 $\mathcal{C}(m_{s,?})$，对于那些指称来说所有的 e 来说 $P(e|s)$ 都是非负的概率，也就是 $\mathcal{C}(m_{s,?}) = e|P(e|s) > 0$。举个例子，短语 “Chicago” 的候选实体集合会包含实体 Chicago, Chicago Bulls, Chicago (1927 film) 等等。

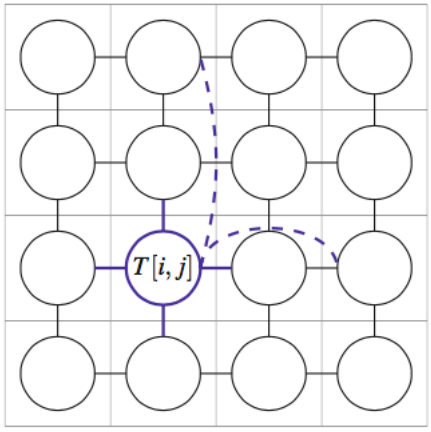

Fig. 1. 用于消岐的图模型。圆圈表示变量，边表示它们之间的依赖关系。为了简洁，此图只显示了单元格 $T[i, j]$ 的非相邻的依赖关系。

3.3 消岐

我们的消岐技术基于一个假设，那就是同一行或者同一列的实体应该是相关的。就像我们的实验中展示的那样，当同时消岐表格中的多个单元格时，我们可以获得更高的准确率通过偏好处理那些有关联性的消岐集合 (也就是集合中的实体是相关联的)。为了利用这个事实，我们使用了一个集体分类技术，其中的软约束条件鼓励同行同列的指称去相互关联来消岐。在给定的表格中的消岐是联合优化的，来使实体集合达到一个全局的连贯。

在消岐的步骤中，一个 EL 系统需要从候选实体集合 $\mathcal{C}(m_{s,?})$ 选择一个参考实体作为一个给定的指称 $m_{s,?}$ 的参考实体。我们提出一个表格，T，作为一个图形化的模型，在这个模型中每个潜在的指称与一个离散的随机变量相关联，随机变量可能的值就是它的候选实体。每个变量都与所有其他同行同列的变量有一个直接依赖关系。模型可以画成一个马尔可夫网络 (Markov Network)，如图1所示，其中每一行 (和每一列) 形成全连接的团。

我们的图形化模型比以前的工作在这个任务中使用的模型更紧密连接。然而，我们发现一个迭代近似推理的方法对于这个模型是相当有效的。*TabEL* 使用迭代分类算法 (ICA) 共同给表格中的所有指称消除歧义。ICA 是一个迭代的推理方法，贪婪地重新分配每个变量的最大似然值，基于其他变量的当前值。在每次迭代中，我们使用一个训练过的本地分类器，$\mathcal{M}_{LR}$，计算每个变量的最

Algorithm 1. ICA for Disambiguation in *TabEL*

1: **function** *TabEL*-ICA($\mathcal{M}_{LR}$, T, maxIter) ▷ $\mathcal{M}_{LR}$: Local Disambiguation Model
▷ T: Input Table
▷ maxIter: Maximum number of inference iterations
2: **for all** $m_{s,?} \in T$ **do**
3: $m_{s,?} \leftarrow m_{s,e_0}$ ▷ where $e_0 \leftarrow \mathcal{M}_{LR}(\mathcal{C}(m_{s,?}))$
4: **end for**
5: $k \leftarrow 1$
6: **do**
7: **for all** $m_{s,?} \in T$ **do**
8: ▷ Re-calculate features according to current assignment to other variables
9: $reCalculateFeatures(m_{s,?})$
10: **end for**
11: $hasChange \leftarrow False$
12: **for all** $m_{s,?} \in T$ **do**
13: $m_{s,e_{k-1}} \leftarrow m_{s,e_k}$ ▷ Re-assign value. $e_k \leftarrow \mathcal{M}_{LR}(\mathcal{C}(m_{s,e_{k-1}}))$
14: **if** $e_{k-1} == e_k$ **then**
15: $hasChange \leftarrow True$
16: **end if**
17: **end for**
18: **while** $hasChange$ AND $k < maxIter$
19: **end function**

大似然值，该本地分类器本质上是一个基于逻辑回归的排名模型。算法3.2展示了ICA算法在图模型上执行迭代推理来找到一个可能性很高的参考实体集合对于给定测试表格中的指称。算法用一个使用 $\mathcal{M}_{LR}$ 的实体 (2到4行)初始化每个指称，然后迭代得重新计算特征和分值(6到18行)直到没有指称的分值变化或达到最大迭代限制次数。

$\mathcal{M}_{LR}$ 基于一组由其它变量的当前设置计算得到的一组特征来对给定的指称进行候选实体的排名。当地模型 $\mathcal{M}_{LR}$ 在一组已标注的指称上训练。$\mathcal{M}_{LR}$ 利用以下这些特征：

先验概率特征 $P(e|s)$ 是由 Web 和维基百科上的超链接估算得到。举个例子，在维基百科中 “Chicago” 这个短语作为锚文本出现了16884次。它链接到289个不同的页面，包括城市，电影，音乐乐队等等。但字符串 “Chicago” 最有可能指的是城市 $(P(ChicagoCity|\text{“}Chicago\text{”}) = 0.80)$。对于每个不同来源的超链接，我们给短语区分大小写的和不区分大小写的匹配计算特征值。此外，我们考虑了区分大小写的和不区分大小写的概率估计的平均值在所有数据来源上。

语义相似度特征 用来衡量表格中一个候选实体和其他实体的一致性。*TabEL* 有三个基于 SR 的特征：平均 SR 介于一个候选实体和所有实体在指称的 1)同行，2)同列，和 3)上下文也就是同行同列。

在 *TabEL* 中，我们使用定义在一对维基百科页面之间的 SR 基于他们的链入链接和链出链接的重叠率。我们使用的 SR 实现方法来自 Hecht et al。这是一个 Milne-Witten 语义相关度计算方法的修改版本，其中维基百科页面的第一段中的链接在计算相似度时被认为是比其他链接更重要的。一个候选实体和在

指称的上下文实体之间的平均 SR 值在表格的 EL 任务中是一个重要的特征，就像在第5部分中所示的实验那样。在维基百科表格上应用 *TabEL* 的特殊情况中，我们也包括了一个特征用于表示候选实体和包含表格的维基百科之间的相关度。

指称-实体相似度特征 捕捉一个潜在指称的上下文 (*context*) 和其每个候选实体的上下文 (*context-representation*) 表示之间的相似度。我们定义的一个指称的上下文就是其同行同列的单元格中的内容。一个实体的上下文就是在训练数据中它出现的地方周围的上下文的聚集。

例如，$m_{Chicago,?}$ 是一个潜在的指称在表格1中的单元格 $T(2,2)$。高亮显示的列被称为指称的列上下文，用 $\mathcal{X}^C(T[2,2])$。类似的，高亮显示的行被称为指称的行上下文，用 $\mathcal{X}^R(T[2,2])$。考虑表格1中 $T[1,2]$ 的实体 New_York_City。其上下文包含实体 Chicago, One World Trade Center, MetLife Tower 等等。为了构建一个 New_York_City 的上下文，我们聚集了语料库中所有链接到 New_York_City 的指称的上下文。

一般来说，单元格 $T(i,j)$ 中的指称的行上下文和列上下文由以下公式得到:

$$\mathcal{X}^R(T[i,j]) = T[i,\cdot]\backslash T[i,j]$$

$$\mathcal{X}^C(T[i,j]) = T[\cdot,j]\backslash T[i,j]$$

$T[i,\cdot]$ 指在 i^{th} 行的单元格，$T[\cdot,j]$ 指在 j^{th} 列的单元格。$\mathcal{X}^R_E(T[i,j])$ 表示在 $\mathcal{X}^R(T[i,j])$ 中找到的实体的多重集合。类似的，我们定义 $\mathcal{X}^C_W(T[i,j])$ 和 $\mathcal{X}^C_E(T[i,j])$ 来表示在 $\mathcal{X}^C(T[i,j])$ 中找到的短语和实体的多重集合。

一个实体的上下文可以从表格的语料库中得到，$\mathcal{T}$ 和已标注的指称。我们定义了2种实体的上下文：1) 单词上下文 (word-context-representation)，$\mathcal{R}_W(e)$ 是一个单词的聚集并且他们的频率从 $\mathcal{T}$ 中所有包含实体 e 的参考的单元格的上下文得到。2) 实体上下文 (entity-context-representation)，$\mathcal{R}_E(e)$，是一种类似的实体聚集及其频率。正式地，

$$\mathcal{R}_W(e) = \uplus_{T\in\mathcal{T}}(\mathcal{X}^R_W(T[i,j]) \uplus \mathcal{X}^C_W(T[i,j]))$$

$$\mathcal{R}_E(e) = \uplus_{T\in\mathcal{T}}(\mathcal{X}^R_E(T[i,j]) \uplus \mathcal{X}^C_E(T[i,j]))$$

其中，$m_{\cdot,e} \in T[i,j]$，也就是说单元格 $T[i,j]$ 包含一个指称其目标实体是 e 并且 $\uplus$ 表示一个多重集合的并集。

TabEL 使用了以下6个基于指称上下文和候选实体上下文之间的相似度的特征。

文本内容相似度特征	**实体内容相似度特征**
$S_C\Big(\mathcal{X}_W(T[i,j]), \mathcal{R}_W(e_c)\Big)$	$S_C\Big(\mathcal{X}_E(T[i,j]), \mathcal{R}_E(e_c)\Big)$
$S_C\Big(\mathcal{X}^R_W(T[i,j]), \mathcal{R}_W(e_c)\Big)$	$S_C\Big(\mathcal{X}^R_E(T[i,j]), \mathcal{R}_E(e_c)\Big)$
$S_C\Big(\mathcal{X}^C_W(T[i,j]), \mathcal{R}_W(e_c)\Big)$	$S_C\Big(\mathcal{X}^C_E(T[i,j]), \mathcal{R}_E(e_c)\Big)$

其中，S_C 表示两个多重集合间的余弦相似度。我们凭借剩余 IDF 值 (r-idf) 来衡量这些多重集中的单词和实体的多样性，剩余 IDF 值根据语料库预先计算好。

存在链接特征 和那些已经链接到它们在输入的表格中的参考实体的指称有关。我们在我们的系统用了两个布尔特征。第一个特性捕捉是否有现存的指称在 $m_{s,?}$ 的上下文中，并且有着外表字符串 s 链接到候选实体。第二个特性捕捉是否由字符串 s' 链接的候选实体与输入表格中的 s 不同。

表面特征 和短语相关，在潜在指称 $m_{s,?}$ 中的 s。我们有两个布尔特征。如果 s 在其单元格中是唯一的文本内容则第一个特征为真，否则假。如果 s 完全匹配一个在输入的知识库 $\mathcal{K}$ 实体的名称则第二个功能为真。

4 系统实现

表格语料库： 我们的表格数据集 $\mathcal{T}$ 有160万张维基百科表格。我们使用 Sweble parser 从维基百科中抽取所有带有类属性 “wikitable” (用于轻易识别数据表格) 的 HTML 表格，来自2013年十一月的英文维基百科 XML dump 文件。如第2部分所述，所有的 HTML 表格都表示成 $r \times c$ 单元格的矩阵。$\mathcal{T}$ 中的表格包含大约3000万超链接。这些超链接中的75%用于构建下面描述的其他资源。另外25%是专门用于本地消歧模型 $\mathcal{M}_{LR}$ 训练，验证和测试，在**??**节中描述。

实体知识库： 我们使用知识库 YAGO，其中包含超过280万个实体，作为我们的参考知识库，$\mathcal{K}$。*TabEL* 将指称链接到 $\mathcal{K}$ 中280万个实体中的一个。YAGO 包含一个维基百科页面和实体之间的双向映射。我们利用这个映射来确定 $\mathcal{T}$ 中超链接的目标的 YAGO 实体。

已标注指称来源： 第3节提到过，我们利用包含已标注指称的数据集来训练 $\mathcal{M}_{LR}$ 模型和构造**??**节描述的上下文。正如上面介绍的那样，维基百科上的页面可以很容易地映射到 YAGO 知识库中的实体。因此，已标注指称可以从 Web 和维基百科上的超链接获得，通过考虑锚文本作为短语和链接目标作为知识库 $\mathcal{K}$ 中的参考实体。

为了可靠地估计一个字符串 s 指向一个实体 e 的概率，我们使用同时来自 Web 和维基百科的超链接。由 Spitkovsky 等人描述的基于英语维基百科概念的谷歌跨语言字典 (Google Cross-Lingual Dictionary) 数据集包含了所有 Web 上链接到维基百科页面的超链接的数据集。我们通过获取自维基百科的超链接增强了这个字典。从 Web 和维基百科上我们挖掘到1亿多个超链接并且获得包含已标注指称的大型数据集。有了像谷歌跨语言词典等优质资源，仅仅依靠先验概率，$P(e|s)$，的实体链接系统仍然可以表现得很好。我们在实验中开发了一个系统，$TabEL_{prior}$，对于一个给定的短语，该系统通过选择最频繁被链接的实体作为潜在指称的参考实体来进行消歧。

5 实验

在本节中，我们评估的 *TabEL* 的准确性并且与先前的工作在 (1) Web表 (2) 维基百科表格 两方面进行比较。在烧蚀研究中我们评估了 *TabEL* 中每组特征的功

能，并建立了基于实体共生率的特征权重。我们展示了集体推理方法，ICA，提高 *TabEL* 的准确率的有效性。

评估标准： *TabEL* 对所有测试指称进行消歧并且总是选择一个存在于给定的知识库中的实体。因此，跟以前在表格上做实体链接的研究一样，我们使用准确率作为我们评估与和其他表格 EL 系统比较的主要指标。我们定义准确率是一个分数，该分数表示测试集的指称在一个 EL 系统中链接正确的比率。与文本 EL 系统进行比较，我们使用宏观平均精度，召回率和 F1 值，该值普遍用于文本 EL 任务。

5.1 Web 表格

为了评估 Web 表格上 *TabEL* 的性能，我们使用先前工作中由 Limaye 等人创建的数据集 **WEB_MANUAL**，如下所述包括有修正和没有修正两种情况。**WEB_MANUAL** 数据集包含9000多个测试指称来自428张 Web 表格。使用 Gupta 等人的方法，这个数据集最初创建是通过从维基百科中寻找与一个包含36张 non-infobox 表格的种子集合相类似的 Web 表格。在 **WEB_MANUAL** 数据集的9036个测试指称中，我们发现大约5%的黄金标注是错误的。Mulwad等人同样已经在这个数据集中指出黄金注释中的错误，但是将修正留给了未来的工作。我们重新标注了这些错误的指称并且创建了一个新的数据集，**WEB_MANUAL_FIXED**，其中是已修正过标注的 Web 表格。

Table 2. 先前工作和 *TabEL* 在 Web 和维基百科表格数据集上的准确率比较

Dataset	**Limaye et al.**	$TabEL_{prior}$	$TabEL$
WEB_MANUAL	81.37	84.41	**89.41**
WEB_MANUAL_FIXED	-	87.56	**92.94**
WIKI_LINKS	84.28	91.27	**97.16**
WIKI_LINKS_RANDOM	-	87.83	**96.17**

表格2显示了 $TabEL_{prior}$ 和 $TabEL$ 在固定数据集 **WEB_MANUAL_FIXED** 上的准确率。为了完善和与先前的工作的比较，我们也显示我们的系统在原始 **WEB_MANUAL** 上的准确性。$TabEL$ 在 **WEB_MANUAL** 数据集上优于以前的工作并且在 **WEB_MANUAL_FIXED** 数据集上优于 $TabEL_{prior}$。一个包含我们在 **WEB_MANUAL** 上发现的错误标注的列表和修正过后的数据集 **WEB_MANUAL_FIXED** 都可以在我们的项目网页上找到。

5.2 维基百科表格

在表格2中我们显示了 $TabEL$ 在两个来自维基百科的数据集上的表现。我们采用了以前 Limaye 等人的工作中的 **WIKI_LINKS** 数据集，其包括超过140000个测试指称来自维基百科的大约3000张表格。$TabEL$ 优于以前的工作是通过减少 **WIKI_LINKS** 数据集上的误差超过75%。

我们评估第二个数据集，**WIKI_LINKS_RANDOM**，来自维基百科表格，以提供一个更全面的表现评估。**WIKI_LINKS** 数据集，来自 Limaye 等人创建，最初的构建是通过选择那些有至少90%的单元格包含链接的维基百科表格。我们相信这个数据集可能是有结果偏倚的因为表格中高密度的链接表明表很重要而且在它们的单元格中可能包含常见实体。这种偏倚在 $TabEL_{prior}$ 在 **Wiki_LINKS** 数据集上的表现和在 **Wiki_LINKS_RANDOM** 数据集上的表现的对比中十分明显。$TabEL_{prior}$ 系统，其对于一个给定的文本指称会选择最常见的参考实体，相比于 **Wiki_LINKS_RANDOM** 它在 **Wiki_LINKS** 上执行得更好。因此，我们创建了 **Wiki_LINKS_RANDOM** 数据集包含随机选择的维基百科表格，无论表格中存在的链接的密度。
Wiki_LINKS_RANDOM 包含大约50000测试指称来自3000张随机挑选的维基百科表格。每个表格中的链接被用作为一个测试指称，它的目标实体被视为一个黄金标注。

表格2表明，$TabEL$ 在 **Wiki_LINKS** 数据集上和在
Wiki_LINKS_RANDOM 数据集上都达到非常高的准确率。高准确率得在维基百科表格上执行表格实体链接很重要，因为许多系统利用维基百科表格中的链接来创建 RDF 三元组或是来支持表格搜索系统 (见第7节)。

5.3 丢失的维基百科链接消歧

表格 EL 任务的一个有趣的变种是识别并消歧未链接的指称为实体，同时保留现有的链接，与上述5.2节中的实验不同，其删除所有现有的链接。为了在这个任务中评估 $TabEL$，我们创建了一个数据集，**TabEL 35K**，包含35000个随机选择的维基百科已标注指称。这些指称不是用于估计先验概率而是用于构建上下文表示。$TabEL$ 在这个任务上执行特别好并且在这个数据集上的准确度是98.38%，而 $TabEL_{prior}$ 的准确性为88.13%。有趣的是,我们发现由 $TabEL$ 造成的16%的错误实际上不是 $TabEL$ 中的错误，而是在维基百科超链接中的错误。另外22%的错误是这样的情况，对于指称黄金注释和 $TabEL$ 的标注都可以被认为是正确的。所有由 $TabEL$ 在这个数据集上造成的错误的细节可以从我们的项目页面中找到。

5.4 与其他的表格实体链接系统的比较

Zhang 等人介绍了一个表格 EL 系统，其能够联合执行三个语义解释的任务。直接与该系统相比较是困难的因为这项工作是在 **WEB_MANUAL** 和 **WIKI_LINKS** 数据集的并集上给出结果并且使用了 F1 值作为衡量标准。与这项工作的 83.7 F1 值相比，$TabEL$ 的 F1 值达到96.92。这和我们的系统在 **WEB_MANUAL** 和 **WIKI_LINKS** 数据集的并集上的准确率是一样的，因为 $TabEL$ 没有忽略任何测试指称。

5.5 与文本实体链接系统的比较

EL 技术对于自由文本输入是完全确立的，并且我们可以说它们也可以应用于表格数据。这里，我们在 **WEB_MANUAL_FIXED** 数据集上评估许多现有的文本 EL 系统的性能并且显示 $TabEL$ 在表格 EL 任务上优于所有文本 EL 系

Table 3. 与6个文本实体链接系统在 **WEB_MANUAL_FIXED** 数据集上的宏观平均精度，召回率和 F1 值比较。结果的 GERBIL 链接：`http://gerbil.aksw.org/gerbil/experiment?id=201507180000`

	AGDISTIS	**Babelfy**	**Dbpedia Spotlight**	**KEA**	**NERD-ML**	**WAT**	*TabEL*
Macro-Precision	0.7773	0.9464	0.8248	0.9209	0.7611	0.9490	**0.9855**
Macro-Recall	0.3587	0.3431	0.1086	0.369	0.6907	0.3442	**0.9237**
Macro-F1	0.3835	0.3663	0.1637	0.4008	0.697	0.3695	**0.9237**

统。我们的结果表明表格 EL 任务能够更好得被像 *TabEL* 这样专门设计来处理表格数据的系统来解决。

我们使用了 GERBIL 框架来比较文本 EL 系统。测试数据集中的每个表格和它们中识别出的指称都转换成文本格式并作为 GERBIL 框架的输入。表格3显示了 *TabEL* 与其他六个 EL 系统相比较的文本宏观平均准确率，召回率和 F1 值。*TabEL* 在宏观平均准确率，召回率和 F1 值上显著优于所有文本 EL 系统。

5.6 烧蚀研究

我们在 **TabEL_35K** 数据集上执行一个消融研究来评估每组特征在 $\mathcal{M}_{LR}$ 的有效性。表格4显示了当特征组从 $\mathcal{M}_{LR}$ 移除时按照错误百分比的增长降序排列的特征组。$\mathcal{M}_{LR}$ 中的所有特征组对系统都有积极的影响，其中 SR 特征是最宝贵的。基于上下文的特征对整个系统的准确性也有很高的影响。

6 分析

在上面的实验中，我们表明了 *TabEL* 始终优于之前的最先进的系统。原因之一是，联合方法，比如在 [4]，使用了常见的最具体类型 (CMST) 假设：所有其他条件不变的情况下，一个 EL 系统应该更偏好链接中列中的指称到知识库中共享 CMST 的实体。原则上，这种假设能够被杠杆化来偏好消歧，导致列实体共享相同的数据类型，从而提高精确度。然而,这种假设在实践中往往是违反了规则因为现有的知识库，其中类型是有限制的，是不完善的和嘈杂的。事实上，当在 DBpedia 本体库映射到类型时，我们发现维基百科表格中只有24.3%的列满足 CMST 假设。进一步，当一列确实存在 CMST，它通常不够具体以至于不能帮助 EL：在一列超过50%的实体仍然保持有歧义即使即使限制实体为列的 CMST。于是，我们不基于知识库中严格的类型限制 EL 目标，二是我们使用一个基于实体共生数据的特征软类型约束，即 SR 和实体上下文相似度特征。同时，不使用一个联合模型来一起解决 EL 和类型识别的任务，我们的系统分别独立解决表格上的 EL 任务。这允许 *TabEL* 回避 CMST 假设的风险。这也由 Venetis 等人在之前的工作中被发现。独立地解决表列类型识别任务比联合解决三个表格语义解释任务要表现更好。

6.1 ICA 的效率

表格5显示了 ICA 达到收敛的迭代次数和两个测试数据集上在多个迭代中执行推理导致的准确率的提升。结果表明集体推理对这项任务很有效。

Table 4. 烧蚀研究：当从 $\mathcal{M}_{LR}$ 中移除各组特征时错误增加的百分率，在 **TabEL_35K** 数据集上

Feature Group Removed	Accuracy	Percent increase in error
-SR	95.31	189.28
-Prior	96.33	126.31
-Existing Link Features	96.55	113.08
-Text Context	96.70	103.73
-Entity Context	96.93	89.27
-Surface	98.32	3.54
Full Wikifier	98.38	0.0

Table 5. 在2个数据集上 ICA 的效率。在2个数据集上，收敛时的准确率比第一轮迭代完的准确率要高。

	WEB_MANUAL_FIXED	WEB_LINKS_RANDOM
No. of iterations for convergence	6	18
Accuracy after iteration 1	88.50	95.85
Convergence Accuracy	92.94	96.17

6.2 实体流行性偏倚分析

EL 任务对于显著的实体是容易的但是对于长尾的不太常见的实体就特别困难。在这里，我们分析维基百科表格中的实体的重要性分布并且显示我们的系统在长尾的不突出的实体上同样表现良好。另一方面，$TabEL_{prior}$ 系统的准确率在不太突出的实体上低，在常见的实体上高。

我们使用一个实体在维基百科中的链入链接的数量来代表其重要性。图2 (a) 展示了一个直方图来统计 (在对数尺度上) **TabEL_35K** 数据集中的指称的目标实体的链入链接数量 。有趣的是，指称的目标实体的链入链接数量呈对数正态分布。估计的正态分布曲线也显示在该图中。

我们把 **TabEL_35K** 数据集根据相等的链接计数 (在对数尺度) 间隔分成5份。图2 (b) $TabEL$ 的性能表现对于目标实体的链入链接的数量变化的变化。我们的系统的准确率在这5份数据集上几乎一样。

6.3 运行时间分析

为了估计我们系统的可扩展性，我们测量了在每个数据集上进行指称消岐所需的时间。表格6显示了每张表格的消歧时间和集体推理达到收敛的迭代次数。Limaye 等人报告的消歧时间为0.7秒/表格。

WIKI_LINKS 数据集中的指称非常密集，因此运行时间较长。有很多参数，比如候选实体的数量，推理迭代的最大数量，收敛标准，都可以调整来进一步缩短我们系统的消歧时间。

$TabEL$ 上一个可能的优化点是候选实体集合中的实体数量。我们改变了一个全局参数来设定每个指称的候选实体数量的阈值并且分析其对于 $TabEL$ 在 **TabEL_35K** 数据集上的影响。我们发现改变该参数从5到20导致准确率有一个可观的提升。增加这个阈值是收益递减的。表格7显示了这些结果和每个指称的候选实体的平均数量根据这个阈值变化时的情况。

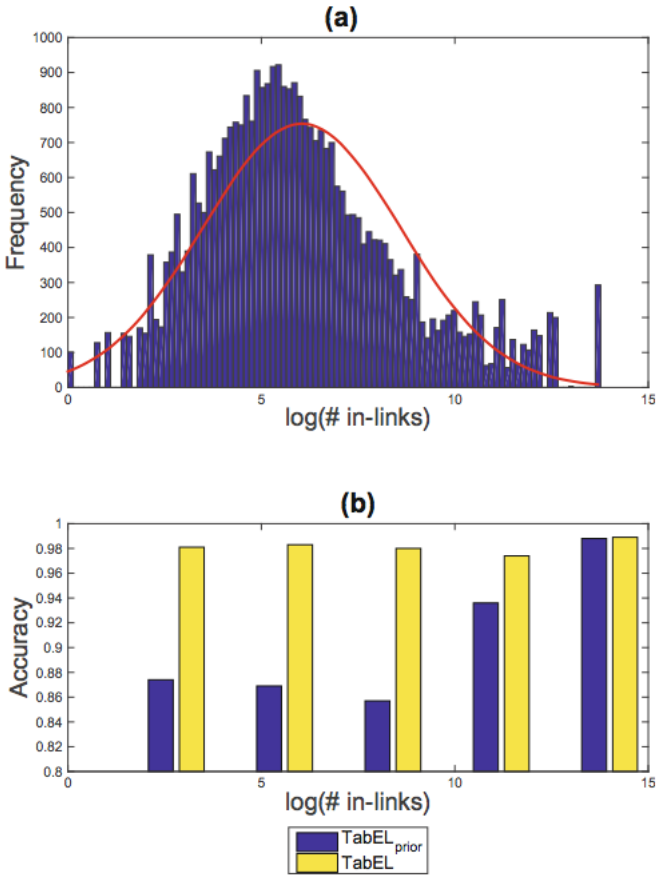

Fig. 2. (a) 链入链接的直方图统计 (在对数尺度上) 维基百科表格中的指称的目标。一个正态分布曲线由红色显示 (b) $TabEL$ 的性能表现对于目标实体的链入链接的数量变化的变化

Table 6. 每张表的消歧时间和在不同数据集上 $TabEL$ 推理迭代的次数

Dataset	Average Time (s/table)	No. of iterations
WEB_MANUAL_FIXED	1.12	6
WEB_LINKS_RANDOM	2.32	18
WIKI_LINKS	31.9	27

Table 7. 改变候选实体的最大数量对于 *TabEL* 的准确率影响，在 **TABEL_35K** 数据集上

Max No. of Candidates	**Average No. of Candidates**	*TabEL* Accuracy
5	1.68	0.73
10	2.32	0.83
15	2.97	0.90
20	3.54	0.94
25	4.04	0.96
40	7.18	0.98

7 先前工作

Cafarella 等人开创了 Web 表格上的研究，并且发现在 Web 上有1.54亿个包含关系型数据的表格。自那时以来，人们尽了各种努力来从网页中抽取语义。Muoz 等人描述了一种方法，依靠现存的链接来将维基百科表格转换为 RDF 三元组。他们使用了现有的知识库 (KB) 像 DBpedia 中的事实，来找到表格中现有的列间关系，然后再为实体提取新的列间关系。Sekhavat 等人提出了一种概率的方法来扩充知识库，使用了 Web 文本语料库中表格化数据中的事实和与知识库中的关系有联系的自然语言模式。这些依靠表格中现存的链接抽取 RDF 的方法可以从我们的系统 *TabEL* 获益显著，我们的系统比以前的系统在 Web 表格上的实体链接任务有更好的精度并且在维基百科表格中效果也特别好。

Syed 等人描述的方法是用 Wikitology 自动推断 Web 表格中的部分语义模型，Wikitology 是一个由维基百科文章和相关页面建立的主题本体。他们的系统解决表格的语义解释的三个任务。Mulwad 等人使用一个图形化的模型关系来对实体链接，列类型识别和关系抽取进行联合建模。对我们的系统最近的工作是由 Limaye 等人和 Zhang 等人做的。他们二者的系统都对实体链接，列类型识别和 Web 表格上的关系抽取进行联合建模。就像在第1节中讨论的那样，由于知识库的正确性和完整性之间的严重依赖性，联合模型运行的风险在于负面影响实体链接的表现。Venetis 等人展示了一个系统来只处理类型识别任务比在这个任务上的联合模型表现要好。在 *TabEL*，我们专注于实体链接这一个任务，并展示 EL 可以通过隔离地解决它来提高表现，而不是通过一个联合的方法。

最后，以前的工作已经研究了建立在已提取 Web 表格上的应用程序。表格的展开已经被 Das 等人，Gupta 等人，Fan等人 和我们之前的工作研究过。Das 等人和我们之前的工作也研究了表格搜索 (*table search*)，该任务就是对于一个给定的文本查询返回一个列表的基于与文本查询的相关性排名的表格。所有这些系统以不同的方式利用表格中现有的参考实体，并且使用 *TabEL* 给表格添加更多的链接可以提高应用程序的准确性。

8 结论和未来工作

在本文中，我们描述了我们的表格实体链接系统 *TabEL*。*TabEL* 使用了集体分类技术来联合消除给定表格中的所有指称的歧义。不使用严格的类型和关

系映射到参考知识库，TabEL而是使用软约束的图形模型，来避开不完整的或嘈杂的知识库带来的错误并且在多个数据集上优于以前的工作。我们还显示，*TabEL* 在长尾的非频繁实体上表现得也相当好 - 对其来说 EL 的任务显得尤其艰难。烧蚀研究证明我们选取的语义相似度特征十分有效。

我们制作了公开开放的表格语料库，包含160万维基百科表格连同已标注数据集可以供未来的 table-EL 系统做比较。

在未来的工作中，我们计划将 *TabEL* 和一些系统集成，这些系统能够识别列类型和表格中的列间关系并将数据转换成机器可理解的形式像 RDF。最后，我们计划在未来发布我们的代码。

致谢 本项研究由 NSF grant IIS-1351029 和 the Allen Institute for Artificial Intelligence 赞助支持。