CTEA：知识图谱的上下文和主题增强实体对齐

摘要

研究在多语言知识图谱（KG）中查找引用同一现实世界对象的实体的问题，即多语言知识图谱的实体对齐。最近，基于嵌入的实体对齐方法在该领域受到广泛关注。他们中的大多数首先通过实体的关系结构将实体嵌入到低维向量空间中，然后通过这些学习到的嵌入结合一些实体相似性函数来对齐实体。即使取得了有希望的性能，这些方法在利用实体上下文和实体主题信息方面存在缺陷。在本文中，我们提出了一种新颖的实体对齐框架CTEA（上下文和主题增强实体对齐），它集成了实体上下文信息和实体主题信息来帮助对齐。该框架通过专门设计的主题模型BTM4EA从实体的属性中学习实体主题分布，并且使用学习到的实体主题分布来过滤一些弱相关实体以用于每个要对齐的实体。同时，我们通过基于翻译的知识图谱嵌入模型将知识图谱嵌入到低维向量空间，并通过附加注意力的卷积神经网络（CNN）从这些向量中挖掘上下文信息。将实体嵌入、实体上下文和实体主题组合起来得到最终的对齐结果。扩展实验表明，我们的方法在大多数情况下都取得了有希望的性能。

关键词：

知识图嵌入

实体对齐

实体上下文

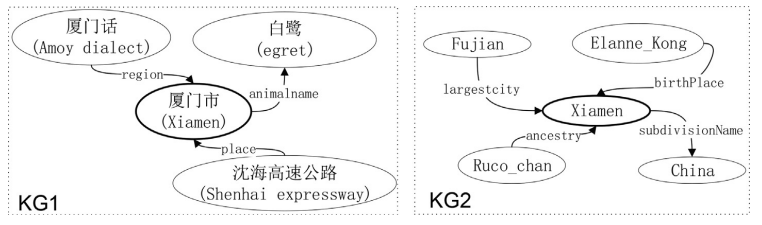
主题模型

# 1.Introduction

KG是将丰富的人类知识组织为结构数据的有向图。它们通常为搜索、推荐和问答系统等智能系统提供知识库。通常，KG编码数百万个实体和数十亿个事实的结构化信息，这些事实表示为两种类型的三元组：关系三元组和属性三元组。关系三元组，例如（AlanTuring，Education，PrincetonUniversity）表明头实体AlanTuring和尾实体PrincetonUniversity具有关系Education。属性三元组保存一个实体的属性，例如（艾伦图灵，出生日期，1912年6月23日）。

近年来，互联网使得越来越多的大规模多语言知识图谱的创建成为可能，例如DBpedia[1]、YAGO[2]和BabelNet[3]。虽然这些多语言知识图谱中的单语言知识可能是由不同的利益相关者或出于不同的意图从不同的数据源单独创建的，这导致引用同一现实世界对象的实体可能以不同的语言以不同的形式出现。例如，DBpedia中的实体http://zh.dbpedia.org/resource/林青霞和实体http://dbpedia.org/resource/Brigitte\_Lin均指中国的一位女演员，它们包含有关该女演员的互补内容，但出现在不同的子KG如果被表示为同一实体，我们可以获得关于该女演员的更全面的了解。不幸的是，并非所有等价物都已在跨语言知识图谱中得到利用。因此，越来越多的研究人员致力于跨语言实体对齐问题的研究，旨在自动匹配不同语言的实体。传统的跨语言实体对齐方法依赖于机器翻译技术或人工设计的特征来发现跨语言等价物。而机器翻译本身就是一个需要研究的复杂问题，而人为设计的特点不仅费时费力，而且带有主观性。最近，随着知识图嵌入模型的发展，针对此任务提出了许多基于嵌入的实体对齐方法（例如，MTransE[4]、JAPE[5]、BootEA[6]）。给定两个知识图谱和一组种子实体对，这些方法首先通过对单语知识进行编码将实体投影到低维向量空间中，然后借助这些实体向量，通过学习到的相似度得分函数计算实体之间的相似度得分。即使取得了可喜的性能，这些方法也存在以下两个局限性：

首先，当前的方法在为源实体找到真正的目标实体时没有对所有目标实体进行区分。我们认为，在精确对齐实体之前，通过一些实体高级语义过滤弱相关实体将有助于提高对齐性能，特别是对于长尾实体和远离预对齐实体的实体。例如，如图1所示，源厦门市和目标实体Xiamen没有对齐的邻居。在这种情况下，当前基于嵌入的实体对齐方法仅在对齐实体时利用实体嵌入，几乎没有证据支持这两个实体之间的等效性。由于预对齐实体提供的对齐信息在传播到两个实体时会丢失很多。因此，这两个实体的嵌入将相当不同，在通过实体相似度为厦门市寻找目标实体时，厦门将排名靠后。而源实体和目标实体可能具有一些相似的高级特征（例如主题信息）。如果我们使用这些高级特征过滤一些相关性较差的实体，则目标实体的排名将排在许多其他实体之前。



**如图1实体结构异质性的说明**

其次，当前基于嵌入的实体对齐方法没有明确地对实体的上下文进行建模。目前的方法，如MTransE[4]、IEAKE[7]和BootEA[6]，通过transE嵌入实体结构，对每个三元组分别建模。对于一个实体来说，即使其邻居的信息可以在迭代过程中传递到中心实体，它们也只能被隐式捕获。基于GCN的实体对齐模型，例如GCN-Align[8]和MuGCN[9]，试图将实体邻居信息集成到实体嵌入中，但它们未能考虑关系语义，而关系语义实际上在实体对齐中发挥着重要作用。当前方法学习到的实体的嵌入只能反映实体本身的语义。虽然相似的实体总是具有相似的上下文，但在对齐实体时应明确考虑实体上下文。

受上述观察的启发，我们提出了一个三步实体对齐框架。第一步是借助实体的一些高级特征为每个源实体生成候选集，第二步是使用实体结构特征生成实体嵌入和实体上下文表示，最后一步是基于实体获得对齐以上结果。

为了过滤弱相关实体，我们建议从实体属性中提取高级特征。由于实体的所有属性名称都是无序的，并且主题模型是词袋模型的簇，在文本聚类方面具有良好的性能。因此，我们采用主题模型来学习实体高级语义。实体的属性数量通常不会很大（在本文使用的一个数据集中，每个实体平均有16个属性），因此我们将BTM[10]（一种对短文本友好的主题模型）进行扩展，以学习主题分布对于所有实体。Jensen-Shannon(JS)散度用于评估两个实体主题分布之间的距离，根据该距离排名靠前的实体被聚类为候选者。

为了生成实体嵌入，我们有多种选择。基于翻译的模型，如TransE[11]、TransR[12]、TransD[13]使用简单的操作和有限的参数来学习实体嵌入，翻译模型速度更快，需要的参数更少，并且相对更容易训练；基于CNN的模型（例如ConvE[14]和ConvKB[15]）由于其参数效率和对复杂关系的考虑，可以学习更具表现力的嵌入。这里，我们使用TransE作为嵌入模型，因为它是最简单的，嵌入模型不是本文的重点。

为了获得实体上下文表示，我们使用注意机制聚合实体邻居信息。基于图神经网络的模型，例如R-GCN[16]和基于注意力的模型[17]可能是可用的选择。R-GCN[16]是将图卷积网络GCN[18]应用于关系数据的扩展。它收集每个实体的邻居信息并为它们分配相等的权重。其聚合模式为。基于注意力的模型[17]为实体邻域分配不同的权重。其聚合模式为。我们可以看到，在R-GCN[16]和基于注意力的模型[17]中，信息转换都是通过矩阵乘法来执行的。众所周知，CNN的表达能力比线性模型更强，并且卷积算子参数效率高。因此，我们将卷积算子应用于每个实体的邻居，并使用注意力机制聚合卷积结果以获得实体上下文表示。

一旦获得每个实体的候选实体集、嵌入、上下文表示，就可以获得对齐结果。实体嵌入用于生成实体嵌入相似度矩阵，其中每个元素表示两个实体嵌入之间的相似度。实体上下文表示用于生成实体上下文相似度矩阵，其中每个元素表示两个实体上下文表示之间的相似度。将两个相似度矩阵与一定的权重相结合，得到最终的相似度矩阵。我们将每行中的元素按降序排列，并将候选中的实体放在其他实体的前面。并获得每个实体的真实目标实体的排名。本文的主要贡献总结如下：

我们提出了一种新颖的实体对齐框架CTEA，它集成了实体嵌入、实体上下文和实体主题来对齐实体（第4节）。

为了对实体的主题信息进行建模，我们提出了BTM4EA，一种面向实体对齐的主题模型，用于从实体的属性中学习主题（第4.3节）。

我们建议利用实体的上下文表示来帮助对齐并设计一个CNN模型来从实体上下文中学习上下文表示（第4.4节）。

在三个真实数据集上的实验表明，CTEA的性能优于或与现有的基于嵌入的实体对齐方法相当。CTEA的每个模块都有助于改进（第5节）。

本文的其余部分组织如下。在第二节中，介绍了一些关于知识图谱嵌入模型和基于嵌入的实体对齐方法的相关工作。第3节阐述了形式化和问题描述。第4节详细阐述了CTEA的费率。我们在第5节中对三个基准数据集评估了我们的方法。第6节中提供了结论性意见和未来的工作。

# 2.Relatedworks

## 2.1.KG embedding

将KG嵌入到低维空间近年来引起了广泛关注。最近出现的知识图谱嵌入模型可以分为两类：（1）基于翻译，（2）基于神经网络。最著名和最典型的是一组基于翻译的模型，TransE[11]是第一个。对于三元组(h,r,t)，这意味着头实体h和尾实体t具有关系r，TransE期望h+r≈t，其中r、h和t分别表示h、r和t的嵌入。

TransE是样本，并且在1对1关系的链接预测任务中具有良好的性能。但是当涉及到1对n、n对1和n对n关系时，它会犯错误，因为它学习每个项目（即实体和关系）的单一表示，这将导致不同实体的相同表示。为了解决这个问题，TransH[19]引入了关系相关的投影向量，将实体投影到关系相关的超平面上，然后在每个特定的超平面上执行平移。TransR[12]为每个关系r分配一个转移矩阵，将实体向量映射到不同的关系空间，并在关系空间中进行翻译。TransD[13]通过为三元组中的每一项分配两个向量来扩展TransR，一个用于嵌入，一个用于投影。TranSparse[20]将TransR中的传输矩阵替换为自适应备用矩阵，考虑到不同的关系链接不同数量和不同类型的实体对（异构性），并且特定关系中的头实体和尾实体的数量也可能不同（不平衡）。由于实体在呈现给不同的三元组时具有不同的作用，TransAt[21]关注实体向量的不同维度以适应不同的关系，以确保转换仅关注特定维度。此外，DistMult[22]使用加权元素级点积来建模实体关系，而ComplEx[23]通过利用复杂嵌入和Hermitian点积来推广DisMult。这类模型需要的参数较少，相对更容易训练，因此受到许多基于KG的应用程序的青睐。

考虑到基于翻译的模型表达能力较差，提出了几种利用深度学习技术进行知识图谱嵌入的神经网络模型。ProjE[24]使用多层感知器，ConvE[14]使用嵌入的二维卷积来链接预测。有两种基于图神经网络的模型：R-GCN[16]和基于注意力的模型[17]，它们收集实体的邻居信息以增强实体表示。

虽然复杂模型具有相对更好的性能，但这里我们利用翻译模型作为我们的嵌入模型，因为它的简洁性。

## 2.2.Embedding-based entity alignment

大多数现有的KG嵌入模型都专注于对单个KG进行建模。然而，随着使用知识图谱的应用程序呈现出更多的多样性，单个知识图谱很难提供所需的所有知识。解决这个问题的一个有效方法是通过实体对齐来整合不同知识图谱之间的异构知识。目前基于嵌入的实体对齐模型可以分为两类：（1）基于翻译的实体对齐模型，（2）基于图神经网络（GNN）的实体对齐模型。

JE[25]利用TransE将不同的KG嵌入到统一的空间中，目标是每个预对齐对中的实体具有相似的嵌入。MTransE[4]结合TransE将KG结构编码到特定语言的向量空间中，并设计了三种不同的技术来对比对进行评分。大多数现有方法都受到缺乏足够先验对齐的挑战，为了解决这个问题，IEAKE[7]和BootEA[6]是两个自学习模型，它们通过参数共享和参数将两个KG嵌入到统一的低维向量空间中首先交换，然后迭代地将新实体对标记为监督数据。这些模型仅依赖于知识图谱中的关系三元组，而忽略其他字面特征。JAPE[5,26]在学习实体嵌入时使用实体属性。为了利用实体的文字描述，KDCoE[27]共同训练多语言KG嵌入模型和多语言文字描述嵌入模型，以利用实体的文字描述。AttrE[26]对实体属性进行字符级文字嵌入。MultiKE[28,29]集成实体的多个视图来学习实体对齐的嵌入。这些模型利用补充资源来增强实体表示，它们在某种意义上对实体的高级语义进行建模，但忽略了实体结构上下文。

CN-Align[8]使用GCN将不同语言的实体嵌入到统一的向量空间中。它结合了结构嵌入和属性嵌入来获得准确的对齐。MuGNN[9]和KEGC[30]共同完成KG并通过跨KG注意力对齐实体。然而，他们没有明确地建模关系语义。VR-GCN[31]集成了GCN的优势和知识图谱中的平移特性，以对关系信息进行建模。

为了用上下文信息来表示实体，GraphMatching[32]引入了主题实体图和公式化的实体对齐任务作为图匹配问题。它不仅利用实体和实体上下文的表示，还利用全局信息来查找对齐。但在对实体上下文进行建模时，它忽略了关系三元组中的关系标签。

CEAFF[33]提出了一种具有自适应特征融合机制的基于集体嵌入的EA框架，而不是在对齐模型上做出改变。它首先生成结构相似度、语义相似度和字符串相似度矩阵，然后将这些相似度矩阵与离线融合策略相结合。实体嵌入和实体上下文表示的组合模式在我们的方法中也是离线的，稍后将对此进行说明。

考虑到访问预先对齐的实体对的昂贵性，工作[34]以完全无监督或弱监督的方式在对抗性学习框架下对齐实体。

与之前的工作不同，我们将实体对齐分为三个步骤：候选生成、实体表示学习（包括实体嵌入和实体上下文表示）和实体对齐。正如我们将在本文后面展示的那样，这样做会在搜索其真实目标实体时排除许多与源实体全局不相似的实体。

# 3.Formalization and problem statement

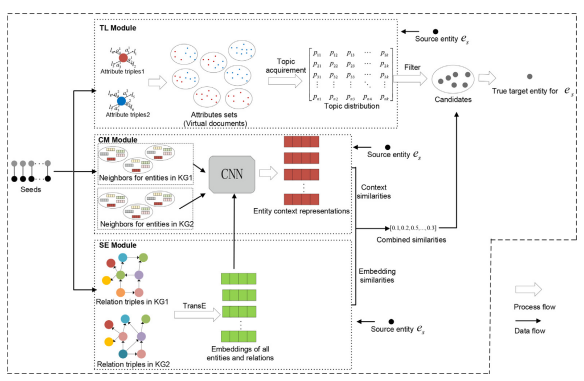
知识图谱可以形式化为KG={E,R,A,RT,AT}，其中E是实体集，R是关系集，A是属性集，RT={(h,r,t)∈H×R×T}为关系三元组，其中H⊆E为头实体集，T⊆E为尾实体集，AT={(e,a,v)∈E×A×V}为属性三元组v是属性a的值。在三元组(h,r,t)中，头实体h、关系r和尾实体t的嵌入分别用粗体h、r、t表示；三元组(h,r,t)的得分函数由fr(h,t)表示。不失一般性，我们只研究两个KG之间对齐实体的情况，并将本文中的实体对齐场景形式化如下：令KG1={E1,R1,A1,RT1,AT1},KG2={E2,R2,A2,RT2,AT2}表示要对齐的源KG和目标KG。Seeds=Sourtrain×Tartrain表示预对齐的实体对集，同时也是训练集的一部分，其中Sourtrain⊆E1称为训练源实体，Tartrain⊆E2称为训练目标实体。元素es∈Sourtrain称为训练源实体，其对应元素et∈Tartrain称为其真实目标实体。实体对齐就是找到一个映射m：Sourtrain→Tartrain，并根据这个映射来对齐更多的实体。在不同类型的实体对齐任务中，映射m可以是注入、满射或双射，本文重点关注双射情况，也就是说，我们假设测试源实体Sourtest和测试之间存在一对一的对齐目标实体Tartest。

# 4.ContextandTopicEnhancedEntityAlignment

在本节中，我们首先介绍CTEA的框架，然后详细阐述该方法的技术细节并讨论几个设计问题。

## 4.1.Overview

CTEA的框架如图2所示，它包含三个模块：实体主题学习（TL）模块、结构嵌入（SE）模块和上下文建模（CM）模块。在训练阶段，给定两个知识图谱，用不同的语言命名为KG1和KG2，首先使用两组属性三元组AT1和AT2构建虚拟文档（实际上是属性集），并将这些虚拟文档传输到TL模块训练主题模型，主题模型将输出每个实体的主题分布。另一方面，两组关系三元组RT1和RT2被输入到SE模块，其中每个实体种子中的两个实体被初始化为相同的嵌入。SE模块产生的实体表示和关系表示，加上预先对齐的实体对，被输入到CM模块并用于优化CM中的参数。



**图2CTEA的框架**

对于一个测试源实体es，为了得到其真实的目标实体，首先计算其与所有目标实体之间的主题分布距离，然后根据这些距离生成其候选实体。另一方面，它的嵌入用于计算它与所有目标实体的嵌入之间的嵌入相似度；学习到的上下文表示用于计算它与所有目标实体的上下文之间的上下文相似度。将结合这两种相似性，在生成的候选者的帮助下找到其真正的目标实体。

## 4.2.Topic learning for entities from attributes

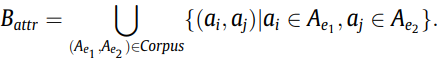
当前基于嵌入的实体对齐模型在为源实体查找真正的目标实体时，所有目标实体之间没有区别。然而，当源实体的邻居很少或远离预先对齐的实体时，实体嵌入不足以准确反映它是什么。因此，通过计算其嵌入与其他实体嵌入之间的相似性，将为其推荐不正确的目标实体。我们希望利用一些高级特征来过滤弱相关实体，以增加推荐真正目标实体的可能性。

实体属性是固有的特征，对于对齐实体非常重要，我们使用实体属性来建模实体的高级语义。由于实体的所有属性名称都可以视为词袋，主题模型是词袋模型的聚类，在文本聚类方面具有良好的性能。因此，我们采用主题模型来学习实体高级语义。由于单个实体的属性数量不是很多，我们改进了BTM[10]（一种不强调单词顺序的短文本友好主题模型）来对实体的主题进行建模，称为BTM4EA。

使用主题模型时必须解决的问题之一是如何定义文档。在这里，我们应用预对齐实体对中两个实体的所有属性来生成虚拟文档。从技术上讲，我们只利用属性名称及其范围类型，但由于其复杂性而忽略了特定的属性值。我们将实体e的属性集定义为Ae={a|(e,a,v)∈AT}。为了使预先对齐的实体对中的两个实体具有相似的主题分布，这两个实体的属性被收集到同一个虚拟文档中。也就是说，训练主题模型的整个语料库构建如下：



其中Ae1和Ae2分别是实体e1和实体e2的属性集。注意，当推断新到达的实体的主题信息时，可以让e1=e2。属性双项表示彼此相关的无序属性对。如果两个属性通常一起用来描述两个对齐的实体，我们称它们是相关的。所有属性位项Battr均通过以下方式获得：



假设a和b是整个语料库的主题分布和每个主题的属性分布的狄利克雷先验。BTM4EA中Battr的具体生成过程可以描述如下：

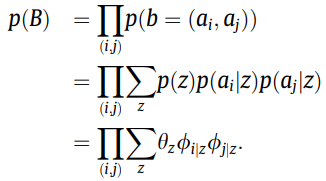
设置主题编号

绘制整个属性bitterms集Battr的主题分布

为每个主题绘制特定于主题的属性分布

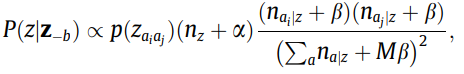
对于每个属性位项b，分配一个主题并选择两个属性

其中表示多项分布。经过上述过程，整个语料库的生成概率可以写为：

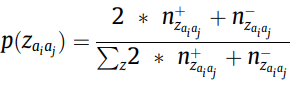


一旦获得了整个语料库的主题分布和每个主题的属性分布，就获得了每个属性位项的主题，进而获得了一个实体的主题分布（下文将进行描述）。这里我们使用吉布斯采样来推断和，就像BTM一样。吉布斯采样是一种简单且广泛使用的马尔可夫链蒙特卡罗算法。它比变分推理和最大后验估计等其他推理方法更准确，更节省内存。

吉布斯抽样过程可以保证通常同时出现的属性倾向于共享相同的主题。否则，我们希望具有相同类型的属性更有可能共享相同的主题。这里属性类型表示其值的范围类型。本文区分了四种属性类型：Integer、Double、Datetime和String。因此，当对特定属性位项b=(ai,aj)进行主题采样时，我们增加分配给与其具有相同类型的属性的主题的权重。条件分布计算如下：



其中



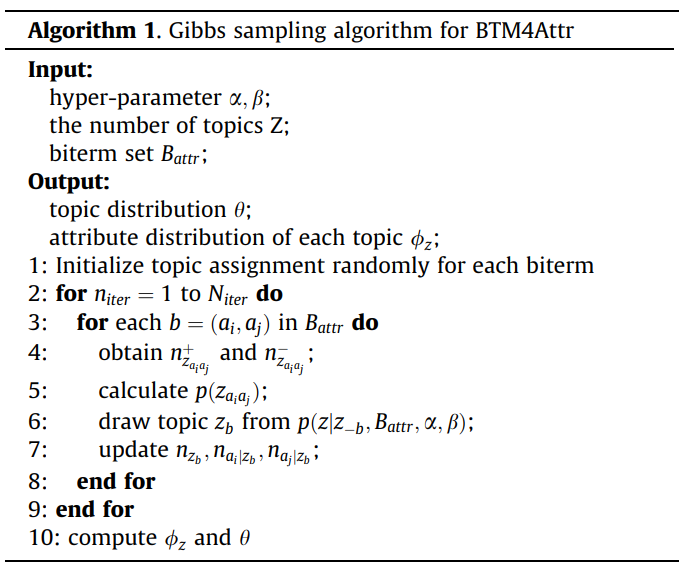
是根据ai和aj的类型附加到z的权重，定义为：



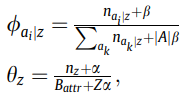
其中Sz是分配为主题z的属性集；Attrai是与ai同类型的属性集合，card(#)是集合的基数，Attraj是与aj同类型的属性集合，定义为：



其中Sz、Attrai和Attraj与上述相同，是某个集合的补集。吉布斯采样过程如算法1所示。



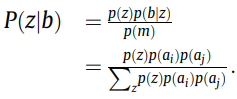
通过吉布斯采样过程，我们可以得到整个语料库的属性主题分布和主题分布：



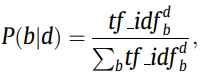
这里A是所有属性的集合。由于我们对整个语料库的生成过程进行建模，而不是对单个文档的生成过程进行建模，因此我们无法直接获取实体的属性集的主题比例，从而无法直接推导出实体的主题分布。为了推断实体的主题，我们假设属性集的主题比例等于属性集中包含的双项的主题比例的期望，计算如下：



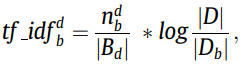
其中



在BTM中，假设文档中每个单词的主题具有相同的重要性，但是我们认为文档中越重要的双项，其主题在推断文档的主题分布时越重要。与BTM计算文档主题分布的方式完全不同，BTM完全计算所有双项的均匀分布，我们考虑了文档中双项的重要性。我们使用一个双项的tf\_idf来衡量该双项的重要性：



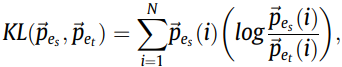
其中tf\_idfb是d中bittermb的tf\_idf值，代表b在d中的重要性，计算公式如下：



其中|Bd|是d中的双项数；|D|是所有虚拟文档的数量，|DB|是包含b的虚拟文档的数量。吉布斯采样收敛后，我们应用詹森-香农散度来计算两个实体的分布之间的距离。假设源实体es的主题分布为，目标实体et的主题分布为，则es和et之间的主题距离计算如下：



KL是两个分布之间的Kullback-Leibler散度，定义为：



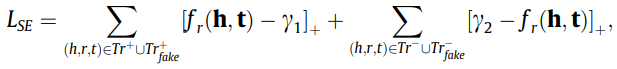
其中N是主题数。

## 4.3.Structure embedding

给定一个知识图谱KG，结构嵌入的目的是对其几何结构进行建模并学习其中每个实体和每个关系的表示。为了简化嵌入过程，我们使用基于翻译的嵌入模型。我们将实体和关系建模为向量，并将关系视为从头实体到尾实体的转换。给定关系三元组tr=(h,r,t)，我们期望h+r≈t。我们使用以下得分函数来衡量三元组tr=(h,r,t)的合理性：



其中||.||表示L2范数。我们更喜欢KG中三元组的低分，而通过用随机实体替换正三元组的头部实体或尾部实体来采样的负三元组的高分。正如BootEA[6]所说，对于实体对齐任务，正三元组的绝对低分有助于减少统一空间中嵌入的漂移，并更好地捕获不同KG的共同语义，我们在这里也应用了这种策略。由于我们认为可以更准确地捕获具有丰富邻居的实体的语义，因此我们像[35]那样为零传入或零传出实体添加假三元组。最后结构嵌入的损失函数为：



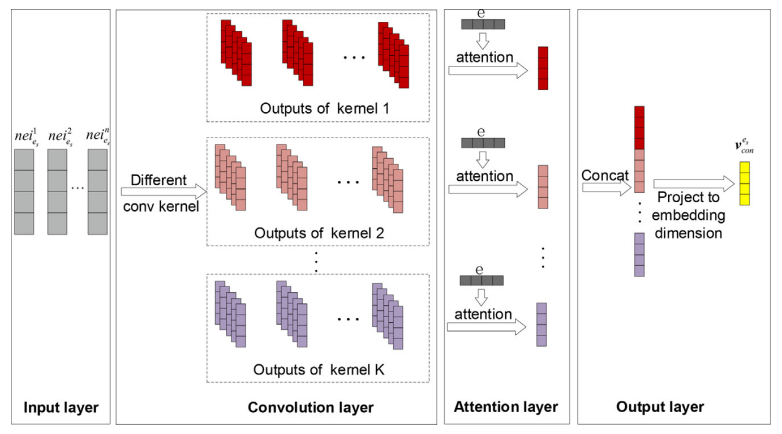
其中Tr+是KG中所有正三元组的集合；Tr+fake是我们构造的假三元组，Tr-和Tr-fake分别是通过替换Tr+和Tr+fake中三元组的头或尾实体采样的所有负三元组的集合。是两个超参数，用于限制正三元组和负三元组的分数。我们让预对齐实体对中的两个实体共享相同的嵌入，以弥合两个知识图谱之间的差距。

## 4.4.Context modelling

如上所述，Structure Embedding模块中获得的实体嵌入仅反映一个实体本身的语义。虽然邻居的信息可以在训练过程中转换为中心实体，但实体嵌入不能明确包含其邻居的语义。众所周知，一致的实体往往具有相似的上下文。因此，我们回顾实体的上下文来帮助对齐。实体的实体上下文定义如下：

实体的上下文：对于关系三元组RT={(h,r,t)∈H×R×T}和实体e，设置Cone={(h,r)|(h,r,t)∈G∧==e}称为实体e的上下文，Cone中的每个元素称为e的邻居，记为neie。

我们提出了一种采用注意力机制的多通道CNN来对实体的上下文进行建模。图3说明了网络的架构。



**图3为源实体es生成上下文表示的过程。是es的邻居表示，最终输出ves是es的上下文表示。**

该网络包含四层：输入层、卷积层、注意力层和输出层。

输入层给定一个源实体es∈Sourtrain，输入层将其所有邻居收集到一个上下文集合中，并输出其邻居的表示，其中card(.)是集合的基数，是的原始表示，其定义如下：



值得注意的是，实体的邻居是无序的，因此上面的i仅显示邻居的数量，但不包含任何顺序信息。

卷积层卷积层用于从邻居表示中提取特征。为了提取不同方面的特征，我们使用K个不同大小的卷积核，每个卷积核有Ω个输出通道。对于源实体es的邻居，卷积核w的通道x的输出为



其中✳是卷积算子，是的偏差，ReLU是激活函数。卷积层中内核k的输出是该内核中所有通道的所有邻居的输出的聚合：

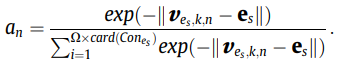


由于我们不区分中的元素来自哪个邻居，因此为了方便起见，我们将重写如下



请注意，每个内核的这些输出都是无序的，原因与上述相同。

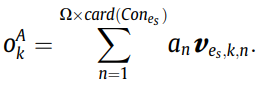
注意力层注意力层用于从卷积层提取的特征中过滤重要信息。在注意力层中，每个卷积核生成的特征向量被转换为与实体嵌入具有相同维度的向量。每个内核都采用注意机制。与源实体嵌入越相似的特征向量对于源实体来说越重要，注意力机制可以增强它们的重要性。特征向量的权重计算如下：



因此，注意力层OA的最终输出由：



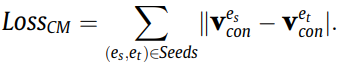
其中是中特征向量的加权和：



输出层输出层使用注意力层OA的输出作为输入，输出与es维度相同的上下文表示。我们在这一层上不使用任何激活和丢失。该层的输出为



其中是输出层的权重矩阵，bo是Mo的偏差，这个向量代表实体es的上下文的整体语义。我们希望相似实体的上下文表示是相似的。为此，我们最小化了Seeds中每个实体对中源实体和目标实体的上下文表示之间的距离。CM模块的损失函数为：



我们用整个损失函数联合训练SE和CM：Loss=LossSE+LossCM：

## 4.5.Entity alignment based on the three modules

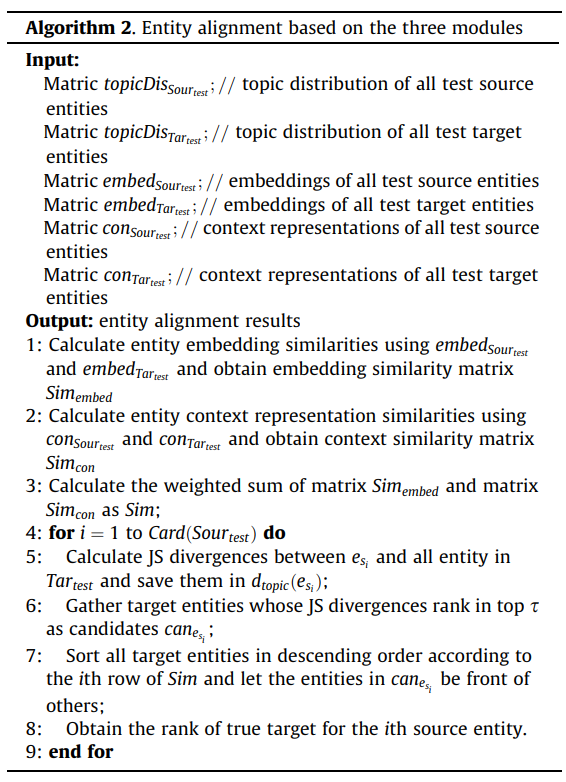
我们使用TL模块为每个测试源实体选择一个候选集，并使用这个候选集来帮助找到其真正的目标。具体来说，给定一个测试源实体es，我们计算它与测试目标集中每个实体之间的JS散度，并将这些JS散度按升序排序，然后选择前t个（t是经验参数）实体为候选者，这有助于SE和CM的结合寻找其真正的目标实体。

从SE获得的实体嵌入用于生成实体嵌入相似度矩阵Simembed，其中每个元素代表两个实体嵌入之间的相似度。从CM获得的实体上下文表示用于生成实体上下文相似度矩阵Simcon，其中每个元素表示两个实体上下文表示之间的相似度。将两个相似度矩阵与一些权重组合起来，得到最终的相似度矩阵Sim：



对于每个测试源实体，我们将每行中的元素按降序排列，并让候选中的实体位于其他实体的前面。并获得每个实体的真实目标实体的排名。

作为一种基于嵌入的实体对齐方法，CTEA并不十分强调正确对齐的精度，而是追求所有测试源实体的较低等级。在对目标实体进行排名时，我们会考虑源实体的组合表示和主题分布。具体过程如算法2所示。



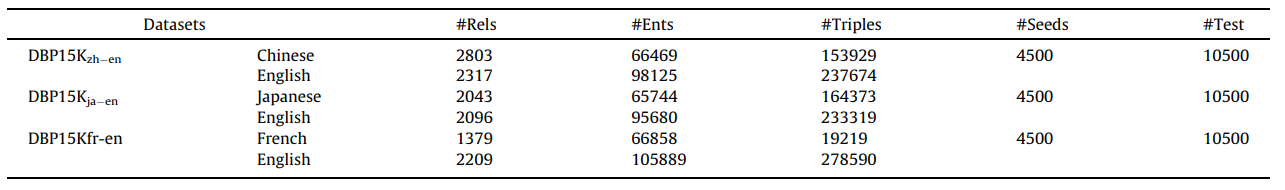
# 5.Experiment

CTEA已在三个开放数据集上得到验证。我们使用Tensorflow来开发我们的方法。实验在配备IntelCorei7-8700K3.70GHzCPU、NVIDIAGeforceGTX1080TiGPU和64GB内存的个人工作站上进行。我们的源代码、数据集和实验结果将在线提供。

## 5.1.Datasetsandbaseline

我们在实验中使用DBP15K作为数据集，该数据集由Ref[5]构建。DBP15k包含三个基于DBpedia多语言版本构建的跨语言数据集：DBPzh\_en（汉译英）、DBPja\_en（日译英）、DBPfr\_en（法译英语）。每个数据集包含15,000个对齐的实体对。与大多数使用这些数据集的作品一样，我们使用这些数据对中的30%作为种子，剩下的作为测试数据。表1列出了三个数据集的统计数据。

**表1实体比对中使用的数据集**



我们将CTEA与七种最先进的基于嵌入的实体对齐方法进行比较，包括四种基于翻译的模型IEAKE[7]、MtransE[4]、JAPE[5]和BootEA[6]以及三种基于GCN的模型GCN-Align[8]、MuGCN[9]和CEAFFw/oC，它是Ref[33]的变体，没有集体对齐策略。

## 5.2.Trainingdetails

在SE模块中，每个KG首先基于TransE学习其嵌入以捕获其结构信息。因此每个KG都有一个独特的嵌入空间。然而，我们在统一空间中计算SE模块中实体嵌入之间以及CM模块中实体上下文表示之间的相似性。我们必须将两个KG嵌入到一个统一的向量空间中。通常有三种方法可以做到这一点：（1）将KG嵌入到单独的空间中，并将一个KG的嵌入转移到另一个KG的嵌入空间中；(2)让种子对中的两个实体共享相同的嵌入；（3）使用预先对齐的实体对生成新的crossKG三元组，在三元组级别上构建不同KG之间的桥梁。我们在本文中采用第二种方式。

由于知识图仅包含正三元组，因此需要提前构造负三元组以满足SE模块中基于边际的目标函数的需求。一种广泛使用的生成负三元组的采样方法是均匀负采样，其中通过随机替换正三元组的头实体或尾实体来获得负三元组。但正如BootEA所说，采样的替换器可能与原始实体完全正交，因此对嵌入学习的贡献不大。在本文中，我们应用邻居抽样。给定要替换的实体e，我们从嵌入中选择其前s个最近邻居作为候选者，并从候选者中随机采样一个实体。我们基于正态分布初始化所有实体和关系的嵌入，并使用梯度下降优化算法Ada-Grad[36]来优化我们的模型。所有嵌入的长度限制为1，以避免通过增加嵌入范数来微不足道地优化目标。实验中使用以下超参数。对于TL模块，主题数Z=30，狄利克雷分布的超参数α=0.1；β=1。对于SE模块，实体和关系的嵌入大小dim=75；s=10为负采样，即每个正三元组采样10个三元组；正三元组的最大值γ1=0.1，负三元组的最小值γ2=2。对于CM模块，核数K=3，卷积核大小分别为2×1、3×1和4×1，每个核的通道数Ω=8。用于平衡实体嵌入和实体上下文的u设置为0.7。

我们训练SE500个时期，然后进行4000个时期的CM训练来学习CM中的参数。值得注意的是，我们在训练CM时不更新SE模块中的参数，并且在训练SE时不更新CM模块中的参数。TL模型在SE和CM之前训练。测试CTEA时将TL的结果作为离线数据。

我们使用hit@10、平均排名（MR）和平均倒数（MRR）作为评估指标来评估CTEA的性能，其定义如下：

正确对齐的MR平均等级，推导如下：



其中rankes是实体es的真实目标的排名。

MRR正确对齐的平均倒数，推导如下：



hit@10排名前10的正确比对比例，计算公式为：



较高的hit@10、MRR和较低的MR表示更好的性能。在测试阶段，对于测试源实体集中的每个实体，我们计算它与测试目标实体集中所有实体之间的距离，然后记录真实目标的排名。当我们借助TL获得每个测试源实体的候选集时，我们将让候选实体排名领先于其他实体，以增加其被选为真正目标实体的概率

## 5.3.Overallperformance

表2显示了CTEA与其他基于嵌入的实体对齐方法的比较结果。我们发现CTEA优于除CEAFFw/oC之外的其他方法[33]，尤其是在MR方面。主要原因是CTEA在对齐之前通过TL模块过滤了一些弱相关实体，并显式地利用实体上下文信息。

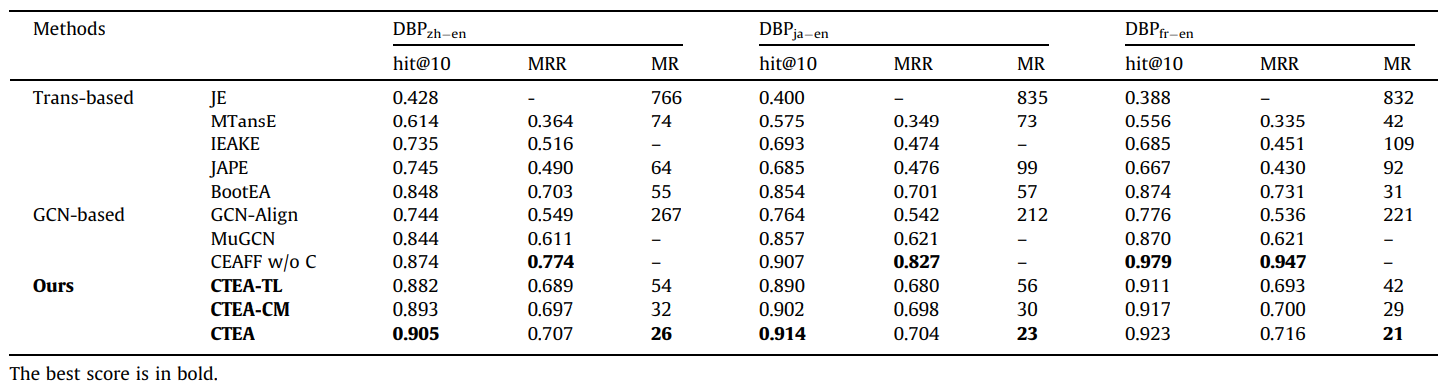
更具体地说，与其他基于反式的模型相比，CTEA在有关hit@10和MR的所有数据集上均优于所有基线。与除CEAFF之外的第二好方法相比，CTEA关于hit@10的平均增益为5.6%，MR的平均降低为21。这是由于CTEA在寻找实体的真实目标时探索了实体的上下文和主题特征，TL生成的候选可以过滤源实体和上下文表示的许多外围目标实体，而其他基于反式的模型（如JE[25]、MTransE[4]和BootEA[6]）仅依赖于KG结构，而忽略其他有价值的特征（如属性）。JAPE[5]利用实体属性，但它使用word2vec来建模属性共现，这可能不适合短文本，因此不适合建模实体属性。关于MRR，CTEA在DBPzh\_en和DBPja\_en上优于其他方法，但在DBPfr\_en上不如BootEA[6]，这可能是因为BootEA[6]是一种迭代方法，其中将添加新对齐的实体对作为种子。与基于GCN的模型相比，CTEA也取得了优异的结果。即使基于GCN的方法GCN-Align[8]和MUGCN[9]尝试将实体邻居聚合到实体嵌入，它们也没有显式地建模实体上下文，也没有过滤源实体的弱相关实体。

至于CEAFFw/oC[33]，根据大多数指标，其性能远远优于CTEA和其他基线。例如，在DBPfr\_en中，根据hit@10和MRR，它达到了0.979和0.947，分别比第二好的高了5%和21%。我们将这一结果归因于它对预训练实体名称嵌入的利用，这是一种非常具有歧视性的信息，这在很大程度上有助于其良好的性能。例如，英文实体名称“China”和中文实体名称“中华人民共和国”在词嵌入空间中已经具有非常相似的嵌入，这使得拥有这两个名称的两个实体之间的相似度相差很大，而无需任何训练过程。有理由相信，将实体名称嵌入集成到CTEA或其他模型中也会带来更好的结果。

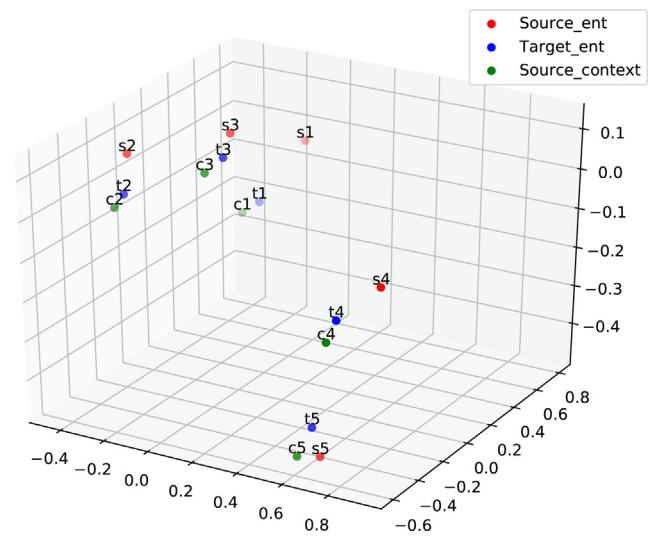
## 5.4.Effectivenessofmodules

为了验证CM模块和TL模块的有效性，我们分别去掉CM和TL来构建两个子框架，并将结果报告在表2中。这两个模块都有助于实体对齐。

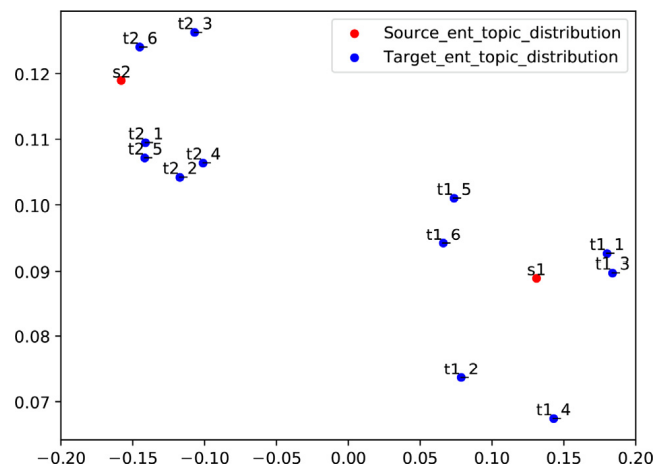
**表2与其他方法的实体对齐结果比较**



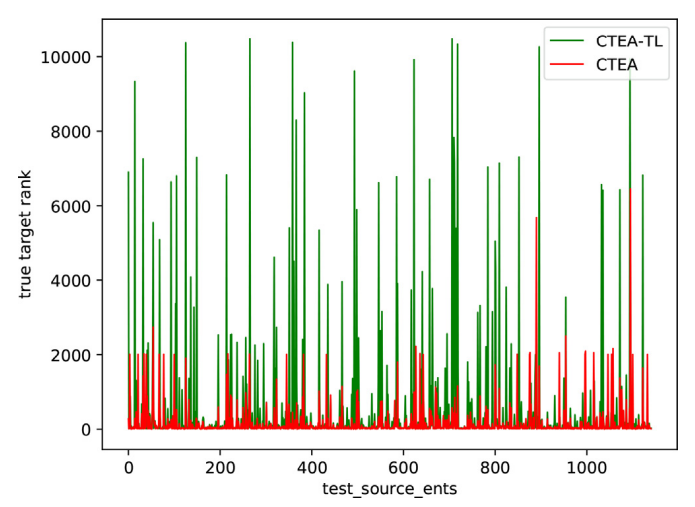
CM的有效性当去掉CM后，性能会略有下降，例如hit@10时性能下降约1%。当两个要对齐的实体有一些对齐的邻居时，CM补充SE。我们从DBPzh\_en的Sourtest中选择了5个实体，并使用PCA将它们的嵌入、上下文表示和真实目标的嵌入投影到三维空间，如图4所示。我们发现，对于一些测试源实体，它们的嵌入距离很远。与目标的嵌入相比，而它们的上下文表示更接近目标的嵌入，这解释了实体上下文表示的有效性。



**图4实体上下文影响的说明。(s1:米尔沃尔足球俱乐部,s2:乌尔奇人,s3:Ipad\_(第四代),s4:伯明翰,s5:珍珠港,t1:Millwall\_F.C.,t2:Ulch\_people,t3:IPad\_(4th\_generation),t4:Birmingham,t5:Kim\_Jin\_su,ci(i=1,2,3,4,5):contextrepresentationofsi(i=1,2,3,4,5),respectively).**



**图5按实体主题获得的候选者。（s1：艾哈迈德.马田.达夫塔里，t1\_1：Lindsey\_Graham，t1\_2：Ahmad\_Matin-Daftari，t1\_3：Samuel\_Osgood，t1\_4：Christine\_Lagarde，t1\_5：突尼斯总统，t1\_6：爱德华，\_the\_Black\_Princes2：，t2\_1：Takumi\_Minamino、t2\_2：Marc\_Overmars、t2\_3：Vagner\_Love、t2\_4：Kasper\_Kusk、t2\_5：Paul\_Pogba、t2\_6：Fabio\_Quagliarella）。**

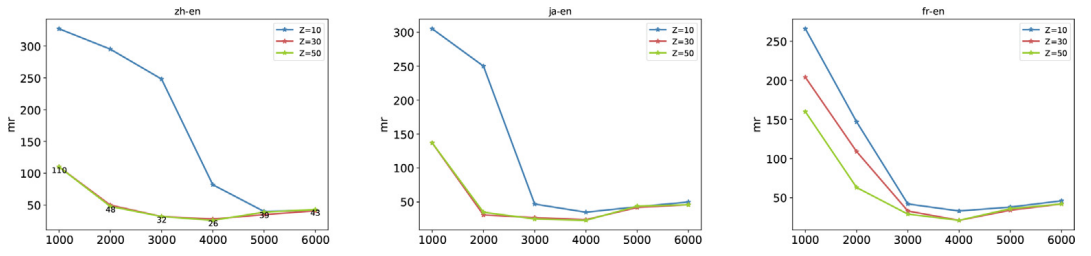


**图6DBPzh\_en中源实体的CTEA和CTEA-TL之间的真实目标排名比较。**

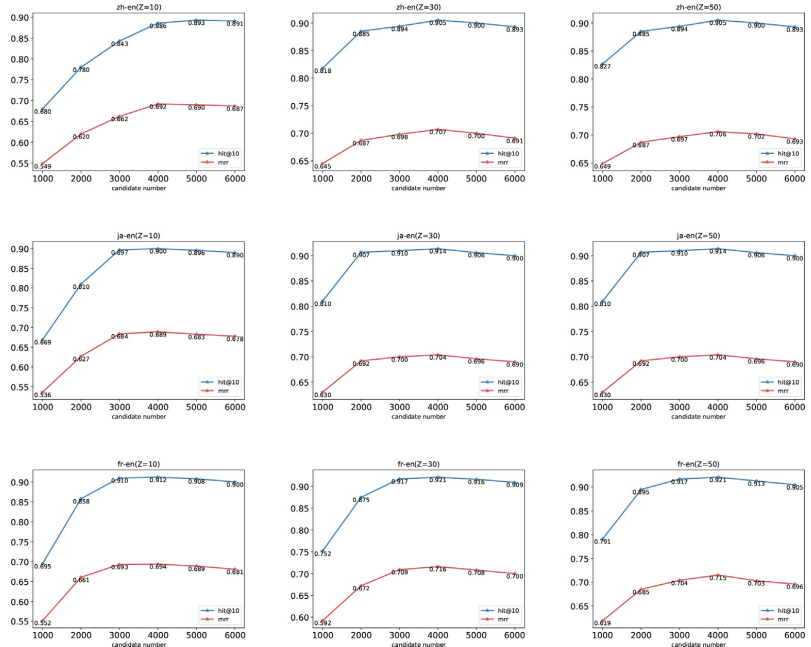
TL的有效性当去掉TL后，性能明显下降，特别是对于MR。这表明TL模块在寻找特定源实体的真实目标时确实过滤了具有不同全局语义的实体。如图5所示，TL帮助源实体从目标实体集中找到与其主题分布相似的候选实体。例如，对于DBPzh\_en中的源实体s1：艾哈迈德.马田.达夫塔里（英语为Ahmad\_Matin-Daftari），TL推荐一些实体作为候选实体，例如t1\_1，t1\_2，t1\_3，t1\_4，t1\_5，t1\_6都是政客，因为这些实体具有通常在s1的属性中同时出现的属性。从这些候选者中，以下模块可以更容易地找到真正的候选者。为了进一步解释TL精确过滤了一些外围实体，我们收集了DBPzh\_en的Sourtest中真实目标排序大于从CTEA-TL得到的10的源实体，并在图6中显示了CTEA-TL和CTEA的排名可以看到红线总体比绿线低，添加TL时大Rank的实体较少，这符合我们设计TL模块的初衷。

## 5.5.Impactofcandidatenumber

我们框架中TL的职责是过滤一些实体并为每个源实体生成候选集。由于TL粗略地生成候选集，因此如何确定候选集的数量是一个棘手的问题。如果候选数量太少，可能会排除部分测试源实体的真实目标；如果数量太多，候选者可能无法达到我们想要的效果。我们考察了考生人数与CETA整体表现之间的关系。



**图7不同主题数下候选数对MR的影响**



**图8不同主题编号下候选编号对Hit@10和MRR的影响**

除了主题编号Z=30报告整体结果外，我们还选择Z=10和Z=50来测试候选编号对CTEA整体性能的影响。我们在{1000,2000,3000,4000,5000,6000}中选择候选编号t，并在图7和图8中报告这些候选编号下CTEA的性能。如图7和图8所示，在Z=30下，当t≤4000，整体性能随着s的增加而提高，因为随着s的增加，候选集中包含真实目标的概率也更大。请注意，当t≤2000时，CTEA的性能低于CTEA-TL，因为在TL从候选集中排除一些真实目标的情况下，候选数量太小。当t>4000时，整体性能逐渐下降，因为候选集太大，无法过滤出与源实体语义相似度不大的实体，导致CTEA最终退化为CTEA-TL。请注意，当选择Z=30和Z=50时，我们可以获得几乎相同的结果。当Z=10时，整体性能具有相似的趋势，但结果不如Z=30和Z=50，因为主题数量太小，TL无法学习实体各自的高级语义，因此无法生成合适的候选人。

# 6.Conclusion

在本文中，我们提出了一种新颖的实体对齐框架CTEA，它合并实体上下文和实体主题信息以帮助对齐实体。我们首先通过改进的主题模型BTM4EA从实体属性学习实体主题。然后使用实体之间的JS分歧来生成每个源实体的候选者。为了利用实体上下文来帮助对齐实体，我们将知识图谱嵌入到低维空间中，并使用多通道CNN从中心实体的邻居中挖掘实体上下文。实体嵌入和实体上下文相结合，在候选者的帮助下找到真正的目标实体。对三个真实世界数据集的实验和进一步分析证明了我们方法的有效性。我们在本文中分别训练TL和其他两个模块，然而，实体的主题分配尽管受到实体表示的影响。未来我们会在线优化主题模型，并用SE和CM来训练TL。我们将寻求其他模型，例如GAT，来对实体的上下文进行建模。在本文中，我们使用实体结构信息来生成实体嵌入。虽然还有其他功能，例如参考文献使用的实体名称。[33,28]，参考文献使用的实体描述。[29]，它可用于生成更具代表性的实体嵌入和更具代表性的实体上下文表示。我们将来会考虑应用这些功能来完善我们的三步实体对齐框架。