跨知识图谱实体对齐

唐晓彬 2019101407

随着计算机计算能力的不断提升与可获取数据量的不断增大，机器学习与人工智能技术在工业界的应用也越来越多。比如运用于推荐系统、问答系统、搜索引擎等领域。通常知识图谱的信息越全面，下游应用的效果也会随之提升。

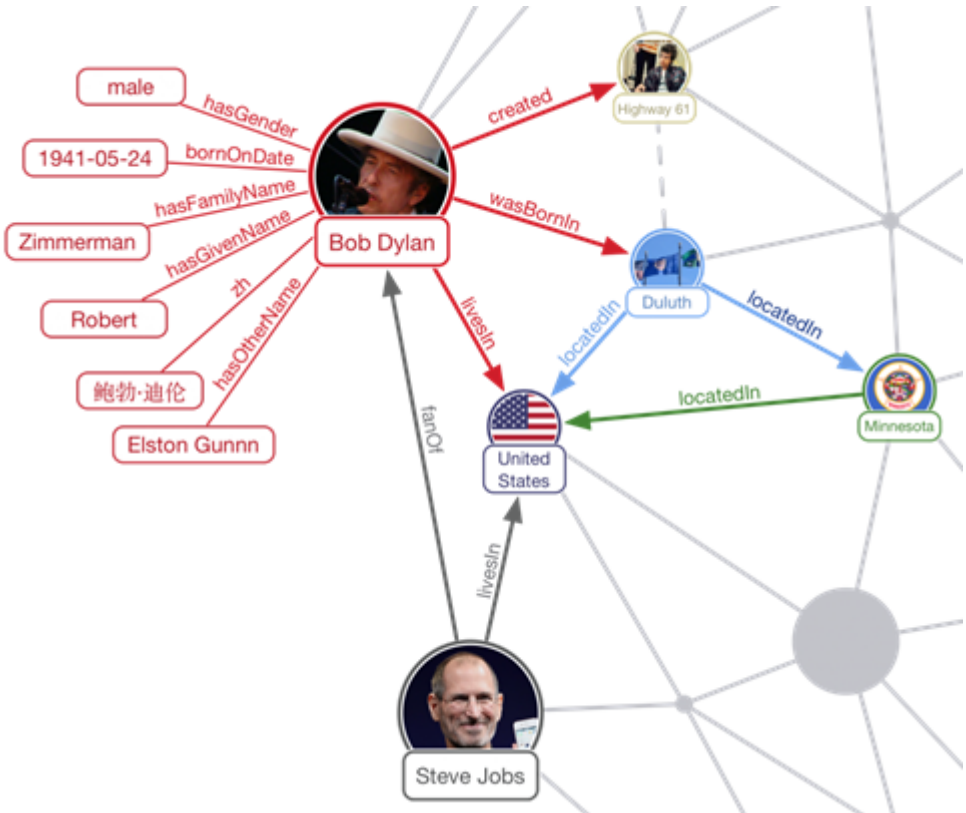
现在已经存在许许多多的知识图谱，如IDBM, DBpedia, YAGO, WordNet等等。这些知识图谱也被广泛应用于各式各样的AI领域的相关任务上。然而关于现有的知识图谱比较显著的一个问题是其不完全性，现有的知识图谱往往只记录了一部分现实世界的知识，对于剩下的部分则需要补充。针对这个问题有两种思路可以解决，一种是知识图谱补全，根据知识图谱内的信息来推理得到知识图谱内没有记录的知识，另外一种则是实体对齐，通过把两个不同的知识图谱的实体给对齐，从而达到把不同的知识图谱合并成一个更加完整的知识图谱的效果，从而更好得支持相关的下游任务。

本次实验则是构建了两个知识图谱实体对齐的数据集，构建了一个跨知识图谱实体对齐的模型，并且将其用pytorch实现，在数据集上验证了模型的有效性。数据集和代码在本报告相同目录下。

1 相关概念

1.1 知识图谱

是结构化的语义知识库，用于以符号形式描述物理世界中的概念及其相互关系。其基本组成单位是三元组，包括关系三元组(头实体-关系-尾实体)，如(Steve Jobs , livesin , United States)，以及属性三元组(实体-属性-属性值)，如(Bob Dylan, bornOnData , 1941-05-24)，实体间通过关系相互联结，构成图状的知识结构，而在图的每个节点则是知识图谱中的一个个实体，下图是一个知识图谱的实际例子。



知识图谱的应用十分广泛，例如智能搜索，深度问答，社交网络，个性化推荐等等，在金融、医疗、教育等行业都有着应用

1.2 跨知识图谱实体对齐

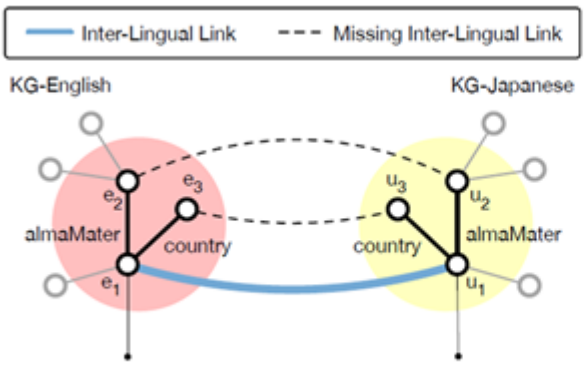
跨知识图谱实体对齐的目的是在不同知识图谱之间寻找现实世界中指代相同的实体对。因此，实体对齐任务能够将不同的知识图谱链接成一个联合知识图谱，从而更好地为各类以知识为驱动的应用提供数据支撑。在现实世界中，这种跨知识图谱已经被对齐的实体对数目往往比较少，以DBpedia为例，仅有15%的实体已被对齐。

2.问题定义

在这里我将知识图谱看成由关系三元组和属性三元组组成，其中关系三元组(头实体，关系，尾实体)构成知识图谱的图结构，而属性三元组(实体，属性，属性值)构成知识图谱内实体的属性信息，具体而言包括实体名信息，实体属性信息，实体描述信息。在这次实验中我们仅使用实体名以及实体的图结构信息。

跨知识图谱实体对齐这一任务定义为：在已知两个知识图谱KG_1, KG_2的关系三元组和属性三元组的情况下，给定少量已对齐实体对，希望构建模型,为KG_1中的每个未对齐实体找到KG_2中能够与之对齐的实体(或者说最相似的实体)

如下图所示，在给定两个知识图谱各自的图结构和属性信息的同时，已知e1与u1已对齐，希望知识图谱的信息和模型为英文知识图谱的实体e3,e2找到在日文知识图谱上与之对应的实体，即找到e3与u3对齐,e2与u2对齐



English: University of Toronto		Japanese: トロント大学	
Attribute	Value	Attribute	Value
Name	University of Toronto	大学名	トロント大学
Type	Public University	学校種別	州立
Found Date	1827-03-15	創立年	1827
Campus	Ontario	キャンパス	セントジョージ（トロント）
Former Name	King's College	旧名	キングスカレッジ
⋮		⋮	
Descriptions			
The University of Toronto is a public research university in Toronto, Ontario, Canada ···		トロント大学は、オンタリオ州、トロントに本部を置くカナダの州立大学である ···	

图结构 关系三元组

实体名 属性 实体描述 属性三元组

目的： KG_1 + KG_2 + 已知实体对 → 新实体对

一般来说，对于KG_1内的实体，在KG_2中最多有且仅有一个实体与之对应

3. 数据集构建

3.1 DBpedia

DBpedia是从维基百科(Wikipedia)的词条里撷取出结构化的资料，以强化维基百科的搜寻功能，并将其他资料集连结至维基百科。DBpedia 同时也是世界上最大的多领域知识图谱之一，拥有超过458万的实体，包括144万5000人、73万5000个地点、12万3000张唱片、8万7千部电影、1万9000种电脑游戏、24万1000个组织、25万1000种物种和6000个疾病。

本次实验使用的数据集也是从DBpedia上抽取数据构建得到，数据来源https://wiki.dbpedia.org/downloads-2016-04，从中按一定规则抽取得到一组小规模的知识图谱，并用于实体对齐。

3.2 数据集构建方法

由于DBpedia本身过于庞大(百万级别节点量的图)，不利于直接进行实验或者直接进行抽取，因此简单定义了一套规则用于进行数据集构建。笼统而言，可以分为两步骤：1. 将DBpedia的知识图谱缩小 2. 从缩小的知识图谱中提取用于实验用的数据集

具体构建方法如下：

1. 根据实体的出现频率对跨知识图谱已对齐实体对进行筛选(去掉出现频率较低的部分)
2. 根据步骤1中保留的实体对，从两个知识图谱中提取相关实体间的关系三元组
3. 将步骤1，2再进行一次
4. 对保留的实体对，从中随机抽取15000对作为数据集的已对齐实体对
5. 根据数据集的已对齐实体对提取以其为头实体的关系三元组
6. 对于尾实体不在已对齐实体集合中的部分三元组，随机删去一些
7. 将数据集已对齐实体对的30%作为训练数据，70%作为测试数据

3.3 数据集信息

本实验从DBpedia中抽取得到两组数据集，其中一个暂时命名为“ZH-EN”，由一个小型中文知识图谱和小型英文知识图谱组成。另一个暂时命名为“JA-EN”，由一个小型日文知识图谱和小型英文知识图谱组成。两个知识图谱各有15000对可对齐的实体对，其中30%作为训练集数据，即已知部分，剩下的70%作为测试集。

数据集的详细信息如下表所示：

		实体数目	关系数目	三元组数目	训练集大小	测试集大小
ZH-EN	中文KG	19388	1701	70414	4500	10500
	英文KG	19572	1323	95142		
JA-EN	日文KG	19814	1299	77214	4500	10500
	英文KG	19780	1153	93484		

3.4 数据集格式

以ZH-EN为例：

- ent_ids_1记录着中文知识图谱中实体及其对应编号信息(可以从中得到实体名信息)
- ent_ids_2记录着英文知识图谱中实体及其对应编号信息(可以从中得到实体名信息)

- rel_ids_1记录着中文知识图谱中关系及其对应编号信息
- rel_ids_2记录着英文知识图谱中关系及其对应编号信息
- triples_1记录着中文知识图谱内的三元组信息，用实体和关系的编号表示
- triples_2记录着英文知识图谱内的三元组信息，用实体和关系的编号表示
- sup_pairs记录着用于训练的已对齐实体对
- ref_pairs记录着用于测试的已对齐实体对

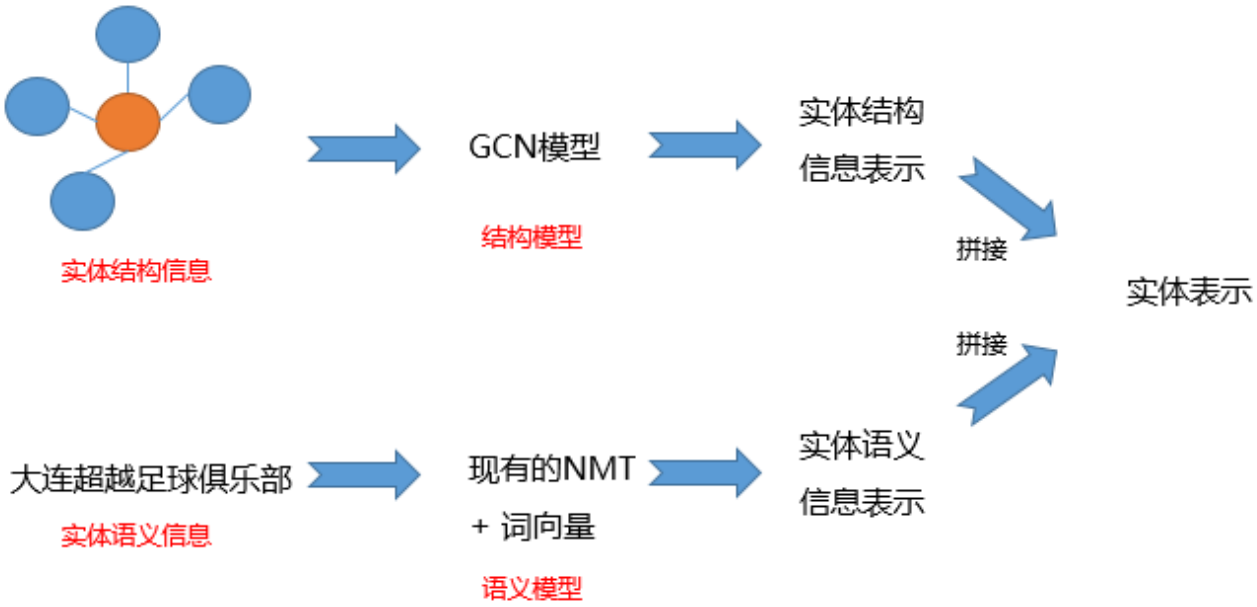
3.5 数据集特点

数据集的特点是 1. 监督数据比较少 2.结构上呈现一定的异构性

4. 模型介绍

4.1 模型总览

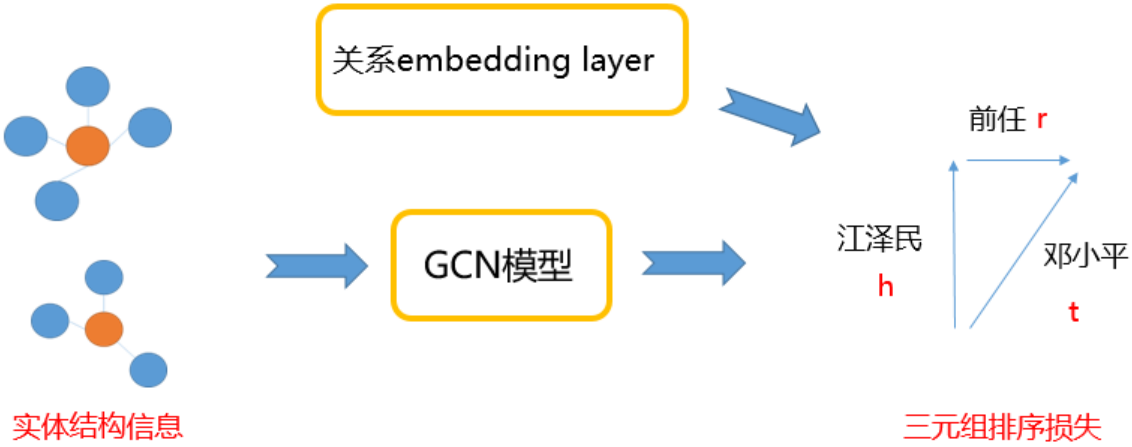
为了利用知识图谱多方面的信息来进行实体对齐，本实验的模型由两部分组成，如下图所示：



其中结构模型部分使用GCN模型来建模，用于获得将不同的知识图谱嵌入到同一空间中，得到实体的结构信息表示，而语义信息模型则是结合现有的神经网络翻译模型与词向量模型，用来将实体的语义信息文本转换成含语义信息的表示，最终，将二者拼接在一起可以用于或者结合了实体多角度信息的表示，这一表示可以通过计算不同实体间的cosine距离，从而用于实体对齐。

4.2 结构模型

4.2.1 结构模型的设计



如图，结构模型由GCN模型和一个embedding layer组成，GCN模型通过将实体的结构信息作为输入，得到实体的表示，embedding layer则是用于记录知识图谱中关系的表示。损失函数使用的是pair-wise ranking loss，这里希望正确的三元组比错误的三元组得分要高。损失函数的定义如下：

$$loss = \frac{1}{N} \sum_{\tau \in T^+, \tau' \in T^-} max(f(\tau) - f(\tau') + 1, 0)$$

其中，T代表正例三元组，即知识图谱中存在的三元组，T^-代表负例三元组，即根据正例三元组以一定规则构建得到的错误三元组，$\tau = (h, r, t)$，代表一个个三元组，$f((h, r, t)) = ||\mathbf{h} + \mathbf{r} - \mathbf{t}||_2^2$代表着三元组的分数，这里希望真实存在的三元组的分数尽可能低，不存在的三元组的分数尽可能高。

在已知已对齐实体以及两个知识图谱各自的三元组的情况下，先构建交换三元组，然后将其加入已知三元组中，并构建模型，获取负例，优化上述损失，则能够达到能将不同知识图谱嵌入到同一空间中，让比较相似的实体之间的距离尽可能近一点。

4.2.2 交换三元组构建

结构模型的一个关键操作便是预先进行三元组交换，在已知KG_1的实体a与KG_2的是实体b对齐的情况下，将KG_1中和a和有关的三元组如(a, r, t)，(h, r, a)内的a替换成b，得到交换三元组(b, r, t)，(h, r, b)并加入训练集，对KG_2也进行相似操作

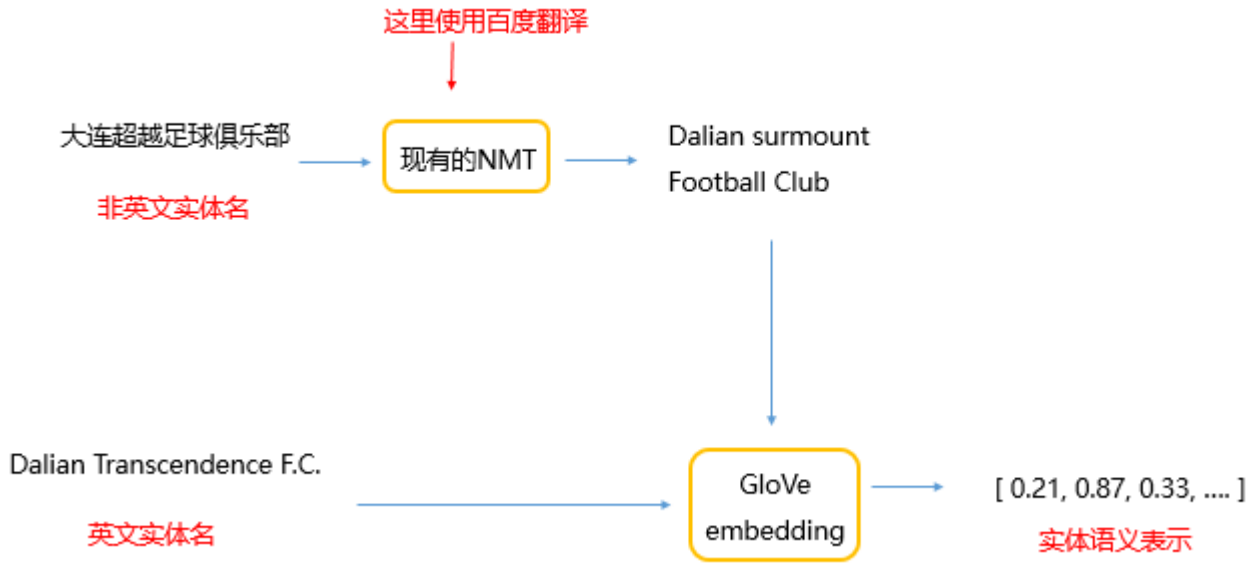
将知识图谱内的三元组和交换三元组作为训练数据，优化模型损失时可以将不同知识图谱嵌入到同一空间中

4.2.3 负例的获取

负例随机将三元组的头尾实体替换成错误的实体得到，例如已知三元组(江泽民，前任，邓小平)，将尾实体用知识图谱内的其它实体进行替换，得到负例如：(江泽民，前任，李克强 ），(江泽民，前任，马云)

在选择替换实体时，根据实体之间的cos相似度选择那些比较像的或者比较有难度的负例，从而避免出现像是(江泽民，前任，英语)，(江泽民，前任，北京市)这类过于简单的负例的产生

4.3 语义模型



如图，语义模型由两部分组成，现有的神经网络翻译模型(NMT)和GloVe embedding模型组成。并且以实体的名称作为带有其语义信息的字符串序列。

对于一个英文实体名，直接使用GloVe embedding词向量模型将其分词后的每个单词的表示平均，得到代表实体语义的表示。

对于一个其它语言的的实体名，则是先使用NMT模型将其翻译成英文序列(这里使用的是百度翻译)，然后使用GloVe embedding词向量模型将其分词后的每个单词的表示平均，得到代表实体语义的表示。

最终得到的实体语义表示可以用于进行实体对齐。

5. 实验

5.1 模型的输入输出

如图所示，模型将两个知识图谱的结构信息和实体的名称作为输入，将实体之间的相似度分数作为输出。

5.2 模型的训练开销

模型训练期间GPU内存占用大概在947Mib左右，训练时间为40分钟，比有着相似性能的实体对齐模型稍快

5.3 评测指标

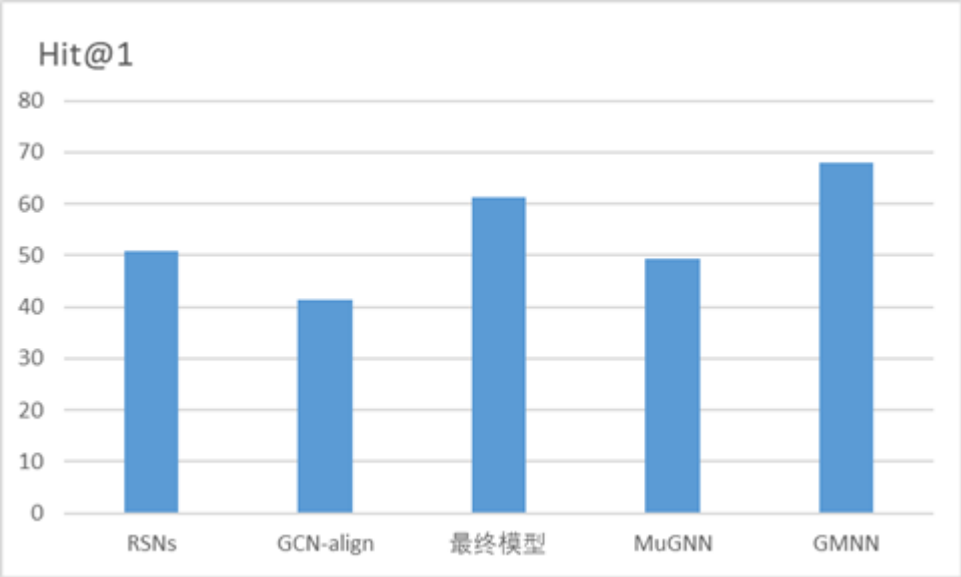
Hit@K,排序结果前TOP-k个实体中存在正确答案的平均个数，这里使用Hit@1，它和准确率是等价的，这一指标越高越好

5.4 模型的效果

本实验在两个数据集上各种进行测试，并且进行了简单的消融实验和现有的一些方法进行对比(直接运行他们的tensorflow代码得到结果)

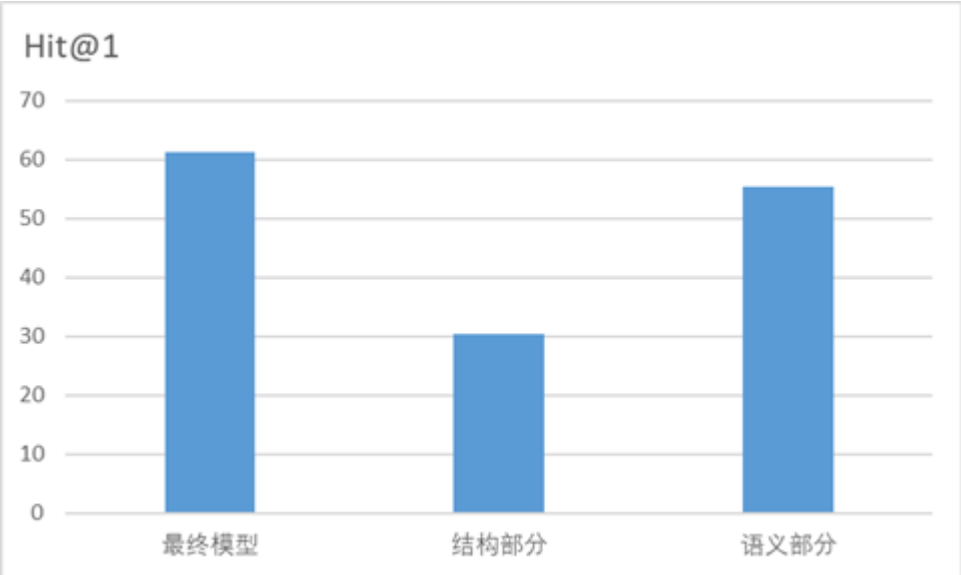
在ZH-EN数据集上

与其他方法对比：



可以看到，本实验所用模型超过了大部分现有方法，离目前最佳的模型GMNN仍有一定差距(但是实验中观察到运行速度比他快20%)

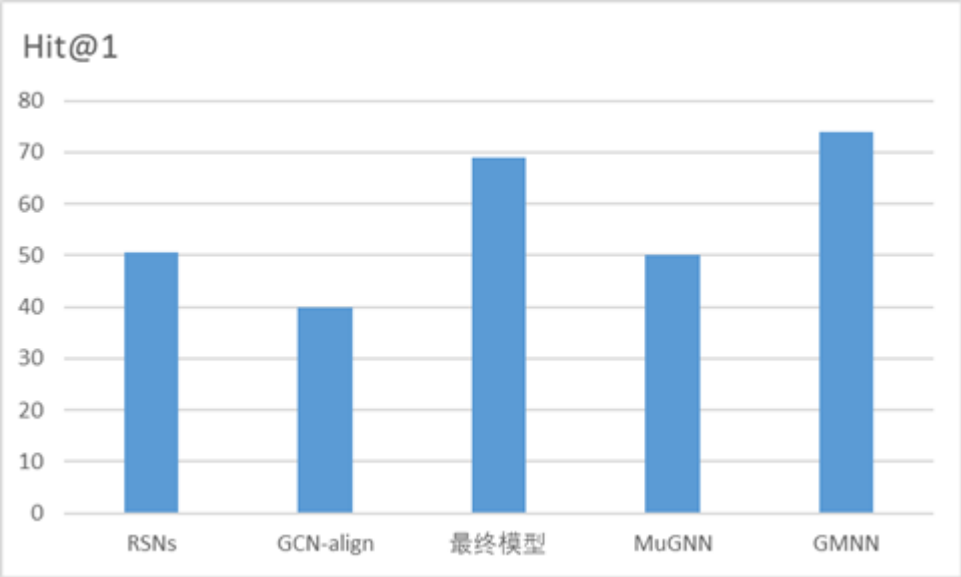
消融实验结果：



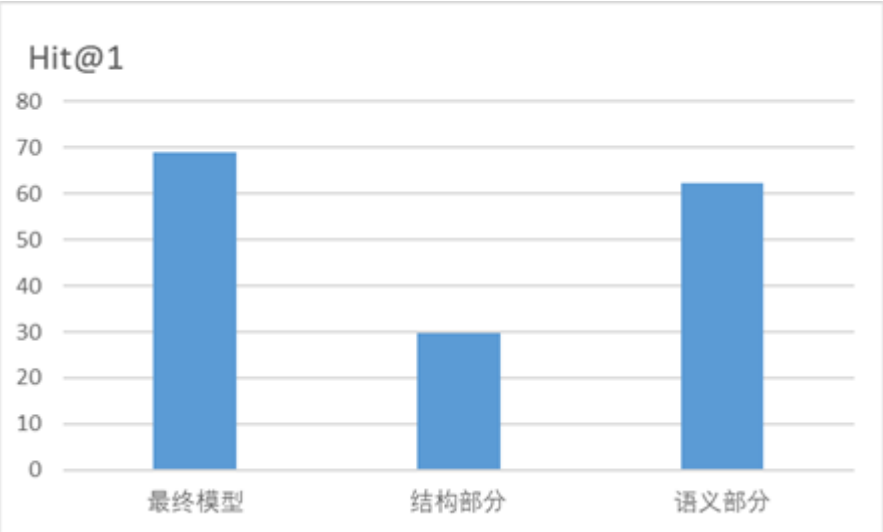
这里比较的是最终模型的效果和只用结构模型部分的表示，只用语义部分效果的表示各自的效果差距；可见实体名承载的语义信息对于实体对齐任务十分有效，同时结构信息和语义信息结合能够达到一个更好的效果。说明这种多视图结合的方式是比较有效的。

在JA-EN上也观察到了类似的现象：

与其他方法对比：



消融实验结果：



6 总结

本实验根据一定的规则构建了两个跨知识图谱实体对齐数据集，从知识图谱的结构信息和实体的语义信息两个角度入手，构建能够利用两方面信息的模型，为每个实体生成表示，并可用于实体对齐。并基于pytorch构建模型并在数据集上进行实验，与不同的模型对比，验证了模型及各子部分的性能。

从实验结构上也能看到此模型还有提升的空间，结构模型部分可以考虑和GAT等方法结合，语义信息部分可以考虑和BERT等相结合。相信仍有提升的空间。