多模态知识图谱上的端到端实体分类

W.X. Wilcke,P.Bloem,V.de Boer,R.H.van ’ t Veer,and F.A.H.van Harmelen

摘要

实体分类在知识图（KG）中起着至关重要的作用，知识图谱上的端到端多模态学习在很大程度上没有得到解决。相反，大多数端到端模型 (如消息传递网络) 只从图形结构中编码的关系信息中学习: 原始值或文字要么被完全省略，要么从它们的值中剥离出来，作为常规节点处理。在这两种情况下，我们都失去了潜在的相关信息，而这些信息本可以被我们的学习方法所利用。我们提出了一个多模态消息传递网络，它不仅从图的结构中学习端到端，而且从它们可能的多模态节点特征的不同集合中学习。我们的模型使用专用的 (神经) 编码器来自然地学习属于五种不同类型模态的节点特征的嵌入，包括图像和几何图形，它们之间的关系信息一起被投影到联合表示空间中。我们在节点分类任务上演示了我们的模型，并评估了每个模态对整体性能的影响。我们的实验结果支持我们的假设，即包括来自多种模态的信息可以帮助我们的模型获得更好的整体性能。

关键词：知识图谱；端到端；多模态消息传递网络；

1 引言

知识图（KG）由关系事实和通过各种关系连接的实体组成，有助于许多与 AI 相关的系统，如推荐系统、问题解答和信息检索。然而，大多数 KGs 是为特定目的和单语环境而构建的，这导致了单独的 KGs，甚至对于相同的概念也有不同的描述。在本文中，我们旨在通过引入消息传递神经网络来展示该原理的第一个概念验证模型，该神经网络可以直接使用异构多模态数据，表示为知识图，并且其本身可以学习仅基于下游任务从每个模态提取相关信息。我们称包含多模态信息的知识图为多模态知识图。最基本的模态关系信息编码在图结构中。其他常见的模态具有数字、文本和时间的性质，例如各种测量、名称和日期，以及在较小程度上的视觉、听觉和空间构成。

通过使我们的模型能够自然地摄取文字值，并根据它们的形态来处理这些值，根据它们的特定特征来定制它们的编码，我们更接近于我们可用的原始和完整的知识。在这项工作中，我们通过专用 (神经) 编码器将来自许多不同模态的信息通过后期融合送入联合表示空间来测试这一假设。通过将我们的方法嵌入到消息传递框架中，并利用数据类型声明和通用词汇表 (如 XSD2 和 OGC3)。为了评估我们的假设，我们在六个不同的异质多模态知识图上调查了每个独立模态对分类准确度的影响。

因为对知识图的多模态学习的兴趣是最近才出现的，所以只有少数多模态基准数据集存在，其中大多数仅包括数字和文本信息[1]。图像通常也包含在内，但存储在图形外部，并使用超链接链接到，并在运行时导入。

2 相关工作

来自多模态源的机器学习是一个被广泛研究的问题。[2]很好地介绍了这个问题及其许多观点。这篇论文的方法是一种通过消息传递网络进行后期融合的方法，侧重于在联合表示空间中表示多个模态。本文有意识地忽略了对齐和翻译的难题：给定模态中的数据仅用于学习字面节点的向量表示。

多种其他方法已经探索了在知识图机器学习模型中使用来自一个或多个模态的文本节点的信息。[3]对链路预测的具体用例进行了概述。虽然调查了两个模型，但只有 MKBE[4]使用了表示各种模态的文字，包括图像。与本文的方法一样，MKBE 使用一组模态特定（神经）编码器将多模态信息映射到嵌入向量。所有这些模型都是基于应用于三元组的分数函数的简单嵌入模型。相比之下，本文的方法包括一个消息传递层，它允许多模态信息在用于分类之前通过图传播几次。本文的模型目前仅在实体分类上进行评估，与这些方法的直接比较超出了范围。

3 本文方法

3.1 知识图谱

在这篇文章中，我们定义了一个多模式知识图 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 在模式 $1, ..., \mathcal{M}$ 是由一组结点 $\mathcal{V} = \mathcal{I} \cup \mathcal{L}^{\mathrm{m}}$ 和一组有向边 $\mathcal{E}$ 定义的标号多向图，$n = |V|$。节点属于两个类别之一：Entity $\mathcal{I}$ 表示对象 (纪念碑、人物、概念等)，以及文字 $\mathcal{L}^{\mathrm{m}}$ 表示通道 $m \in \mathcal{M}$ 中的原始值 (数字、字符串、坐标等)。我们还定义了一组关系 $\mathcal{R}$，它包含组成 $\mathcal{E}$ 的边类型。关系也称为谓词。

$\mathcal{G}$ 中的信息被编码为形式为 (h, r, t) 的三元组 $\mathcal{T}$，具有头部 $h \in \mathcal{I}$、关系 $r \in \mathcal{R}$ 和尾部 $t \in \mathcal{I} \cup \mathcal{L}^1 \cup ... \cup \mathcal{L}^{\mathrm{m}}$ 关系和文字的组合也称为属性或节点特征。如图 1 所示，知识图有七个节点，其中两个是实体，其余是文字。

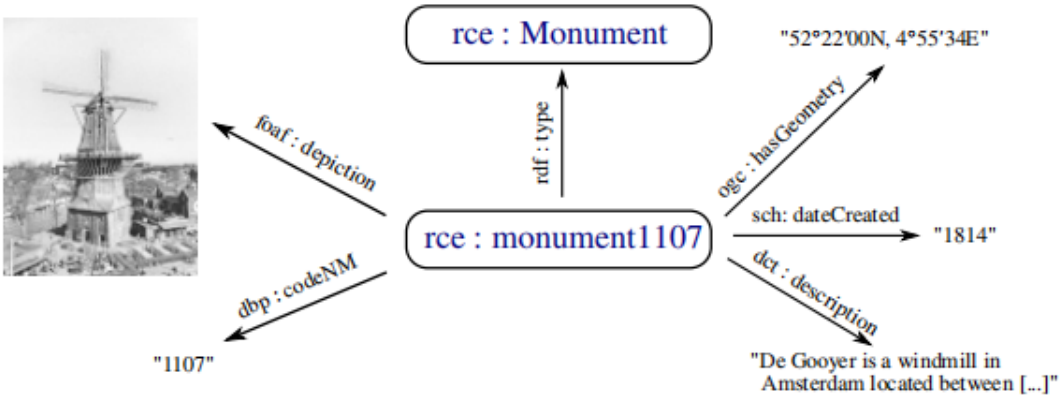

图 1: 荷兰古迹图示例

3.2 消息传递网络

消息传递神经网络[5]是一种图神经网络模型，它使用可训练函数在神经网络的边缘上传播节点嵌入。消息传递的一种简单方法是图卷积神经网络 (GCN)[6]。我们在其上构建的 R-GCN[7]是知识图设置的直接扩展。设 H^0 是图中所有 n 个节点的 q 维节点嵌入的 $n \times q$ 矩阵。也就是说，H^0 的第 i 行是图中第 i 个节点的嵌入，R-GCN 通过以下计算 (图卷积) 来计算 l 维节点嵌入的更新的 $n \times l$ 矩阵 H^1：

$$H^1 = \sigma(\sum_{r \in R} A^r H^0 W^r) \tag{1}$$

这里，σ 是一个激活函数，类似于 ReLU，以元素方式应用。A^r 是关系 r 的行归一化邻接矩阵，而 W^r 是可学习权重的 q $\times$ l 矩阵。该操作通过对节点的所有相邻节点的嵌入进行平均，并通过 W^r 线性投影到 l 维来得到节点的新节点嵌入。然后对所有关系求和嵌入并应用非线性 σ。

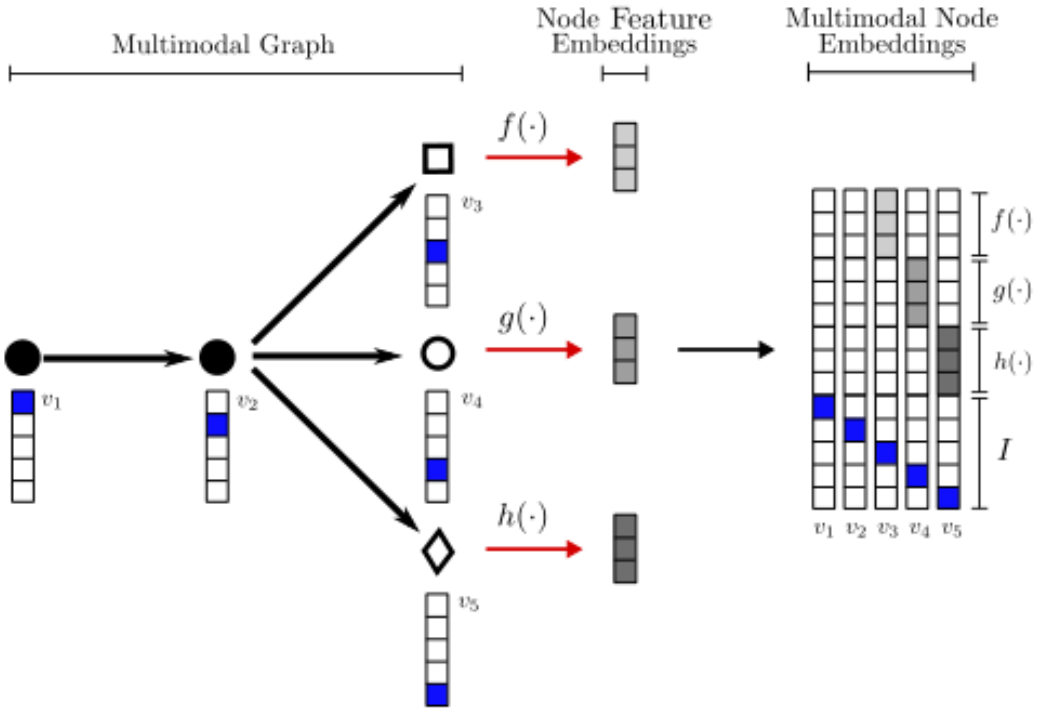

图 2: 概述了我们的模型如何为节点 v1 到 v5 创建多模式节点嵌入

3.3 一种多通道消息传递网络

本文的模型作为消息传递网络的扩展来介绍，它可以从任意图的结构中进行端到端的学习，并且对于它来说，$H^0 = I$。为此，我们设 $f(\cdot), g(\cdot)$ 和 $h(\cdot)$ 作为特征编码器，对于节点 $v_i \in \mathcal{V}$，输出长度为 ℓ_f,ℓ_g 和 ℓ_h 的特征嵌入。定义 F 为具有 $f = \ell_f + \ell_g + \ell_h$ 的多模态特征嵌入的 $n \times f$ 矩阵，并将 F 与单位矩阵 I 连接以形成多模态节点嵌入：

$$H^0 = \begin{bmatrix} I & F \end{bmatrix} \tag{2}$$

嵌入矩阵 H^0 与 A^r 一起馈送到消息传递网络，例如 R-GCN。将来自网络的误差信号通过编码器一路反向传播到输入，编码器和网络都被端到端一致地训练。

3.4 模态编码器

我们为知识图谱中常见的五种不同的模态增加了编码器。我们放弃讨论关系信息 (第六种模态)，因为这已经在信息传递网络的相关工作中得到了广泛的讨论。对于数字和时间信息，由于问题的简单性，我们使用直接的确定性编码。对于文本、视觉和空间信息，我们使用神经编码器，由于其效率和速度，我们选择了卷积神经网络（CNN）。在神经编码器的情况下，我们还引入了一个中间步骤，将原始值转换为其矢量表示。

在下文中，我们让 e_m^i 是节点 v_i 对模态 m 的特征嵌入向量。一个节点的身份向量和它对每个 $m \in M$ 的所有特征嵌入向量 e_m^i 的连接，等于 H^0 的第 i 行。

本文的三个神经编码器都是用 CNN 实现的。对于文本信息，我们使用一个具有 4 个卷积层的时间 CNN，每个卷积层后面都有 ReLU，以及 3 个密集层（表 1），它的最小输入序列长度为 12 个字符。空间编码器也使用了类似的设置，但有 3 个卷积层和不同数量的过滤器（表 2），最小输入长度为 4 个坐标。在这两种情况下，我们在修剪异常值并在需要时使用零填充。对于视觉编码器，我们使用来自[8]的高效 MobileNets 架构，输出维度为 128。所有三个 CNN 都是使用 $\mathcal{N}(0, 1)$ 启动的，并使用迷你

批处理（每个历时 4 次）进行训练。

表 1: 带有 4 个卷积层（顶部）和 3 个密集层（底部）的文本编码器的配置

Layer	Filters	Kernel	Padding	Pool	Dimensions
1	64	7	3	Max(2/2)	-
2	64	7	3	Max(2/2)	-
3	64	7	3	-	-
4	64	7	2	Max(2/2)	-
5	-	-	-	-	256
6	-	-	-	-	64
7	-	-	-	-	16

表 2: 带有 3 个卷积层（顶部）和 3 个密集层（底部）的空间编码器的配置

layer	filters	kernel	padding	pool	dimensions
1	16	5	2	Max(3/3)	-
2	32	5	2	-	-
3	64	5	2	avg(•)	-
4	-	-	-	-	128
5	-	-	-	-	23
6	-	-	-	-	16

4 复现细节

4.1 与已有开源代码对比

本次复现的论文有提供源代码，本人在复现的过程中，参考了作者源代码进行复现。在复现过程中，原作者的代码存在一些细微的 BUG，如数据集的处理以及程序运行相关方面。在对 BUG 进行修改之后，成功跑通了实验。

4.2 实验环境搭建

- **python:** version 3.9
- **pytorch:** version 1.21.1
- **numpy:** version 1.23.3
- **rdflib:** version 6.2.0
- **scipy:** version 1.9.3
- **deep_geometry:** version 2.0.0

4.3 界面分析与使用说明

安装环境 cd mrgcn/ pip install. 安装后，我们必须首先使用配置文件作为参数调用来准备数据集。对于我们论文中使用的数据集，配置文件在目录中可用。**准备数据集，请运行:python mrgcn/mk-dataset.py --config ./if/<dataset>.toml --output ./data/ -vv** 这将创建一个 tar 文件 ()，其中包含运行后续实验所需的所有数据。**通过运行以下命令在准备好的数据集上运行实验:python mrgcn/run.py --input ./data/<DATASET[unix_date]>.tar --config ./if/<dataset>.toml -vv**

4.4 创新点

在本次论文中，主要体现在更换损失函数和卷积层数进行实验，本文的原损失函数为 CrossEntropy 即交叉熵损失函数，在实验中我将其修改为 NLLL 即负对数似然损失函数进行了部分实验。

5 实验结果分析

5.1 实验实现过程

在一个实体分类任务中评估了本文的模型，同时改变了学习过程中所包含的模式。为此，计算了每个结构和模式组合以及所有模式组合的分类准确率，并将其与只使用关系信息和多数类分类器的性能进行评估。

我们测试的另一个维度是图的结构如何被表示并输入我们的模型，以及这对有无节点特征的性能有何影响。本次测试的两种图形表示方法只在如何处理具有相同价值的文字节点方面有所不同。最常见的方法是将这些字面符号折叠成一个节点，称为合并字面价值，而另一种方法是将这些重复的值分开，用同样多的节点来表示它们的值。我们将这后一种称为分割字母价值。

使用 R-GCN 作为主要的构建模块，将各种编码器堆叠在上面。R-GCN 可以对关系图的结构进行端到端的学习，并考虑到关系类型。如果只对从图的结构中学习感兴趣，就让 H^0 成为节点的 $n\times n$ 身份矩阵 I（即 $H^0 = I$）。为了在学习过程中也包括字面价值，或节点特征，我们让 F 是 $n\times f$ 特征嵌入矩阵，并将其与方程 2 中的 H^0 相连接，形成 $H^0 = [IF]$。

为了应对包括节点特征在内的复杂性的增加，我们通过将方程 1 的计算分成结构和特征部分的总和，来优化我们对稀疏矩阵操作的实现。为此，我们再次将 H^0 拆分为身份矩阵 $H_I = I$ 和特征矩阵 $H_F^0 = F$，并将计算结果重写为:

$$H^1 = \sigma(\sum_{r\in R} A^r H_I W_I^r + A^r H_F^0 W_F^r) \tag{3}$$

这里，W_I^r 和 W_F^r 分别是结构和特征成分的可学习权重。对于层 $i > 0$ 来说，持有 $H_F^i = H^i$，$A^r H_I W_I^r = 0$。由于 $A^r H_I = A^r$，我们在计算方程 3 时可以省略这一计算，因此也不再需要 H_i 作为输入。图 3 以矩阵运算的方式说明了这种计算。

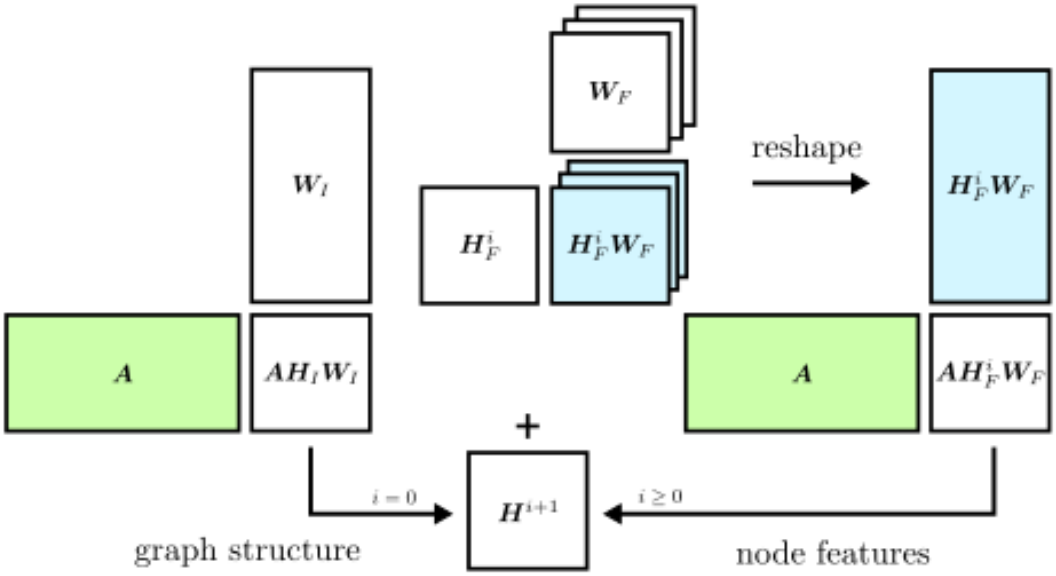

图 3: 对方程 3 的实现的图形描述

5.2 数据集

本文在六个具有不同程度的多模态的知识图谱上评估我们的模型。AIFB、MUTAG、BGS 和 AM 是现有的知识图谱机器学习的基准数据集[9]。然而，AIFB、BGS 和 AM 缺乏准确确定字词模态所需的数据类型声明，因此我们增加了 AIFB+、BGS+ 和 AM2D+ 数据集。SYNTH 是一个合成数据集，这个数据集是为了消除数据集的特定特征对结果的影响，以及图的结构中编码的任何信息的影响。使用 Watts-Strogatz 算法生成了 8 个随机图结构，从中随机抽出 256 个节点作为信号实体，而其余的节点则作为噪声发挥作用。对于每个实体的其中五种模式中的每一种添加了（p=0.9）文字属性，信号实体的值是从两个狭窄的分布中随机抽取的（创造了一个二元分类问题），而所有其他实体的值是通过在手头的模式的价值空间中随机选择一个点来取样的。

5.3 实验结果

本次复现采用的损失函数为 CrossEntropy 即交叉熵损失函数，为常用的损失函数。本次复现选取了其中三个数据集，对于每个数据集，显示了有节点特征和无节点特征的学习结果。所有的结果都是使用具有 16 个隐藏节点的两层 R-CNN 获得的，并在全批模式下用 Adam 训练了 100 个 epochs，初始学习率为 0.01。总体结果显示，当包括某些节点特征时，几乎所有的数据集都有轻微到相当大的改善，除了 AM2D+，其中的差异也可能是由于初始化的随机性。我的实验结果如表 3 和表 4 所示:

表 3: 具有相同价值的字词被算作同一个节点

Dataset	**AIFB+**	**SYNTH**	**MUTG**
Structure+Features	0.7248	0.7123	0.6667
Structure+Numerical	0.7122	0.7429	0.6410
Structure+Temporal	0.7666	0.6981	
Structure+Textual	0.6739	0.6111	

表 4: 具有相同价值的字词被分割成不同节点

Dataset	AIFB+	SYNTH	MUTG
Structure+Features	0.7011	0.6481	0.6011
Structure+Numerical	0.7483	0.6241	0.6071
Structure+Temporal	0.7481	0.6617	
Structure+Textual	0.6667	0.6081	

6 总结与展望

这篇论文主要引入了一个在多模态知识图上进行端到端多模态学习的模型，本文结果表明，包括其他模态的信息可能略微提高模型的性能，也可能大幅度提高，这主要取决于数据的特点以及是否合并具有相同价值的字词（从而在图的结构中编码字词信息）。证明了即通过包括尽可能多的信息，更接近图中的原始和完整信息，使模型能够学习到更好的节点内部表征，结果是整体性能的提高，在未来还需要更多的研究来了解我们如何在学习过程中最好地包括多模态的节点特征

参考文献

[1] LIU Y, LI H, GARCIA-DURAN A, et al. MMKG: multi-modal knowledge graphs[C]//European Semantic Web Conference. 2019: 459-474.

[2] BALTRUŠAITIS T, AHUJA C, MORENCY L P. Multimodal machine learning: A survey and taxonomy [J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 41(2): 423-443.

[3] GESESE G A, BISWAS R, ALAM M, et al. A survey on knowledge graph embeddings with literals: Which model links better literal-ly?[J]. Semantic Web, 2021, 12(4): 617-647.

[4] RILOFF E, CHIANG D, HOCKENMAIER J, et al. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing[C]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018.

[5] PRECUP D, TEH Y W. Proceedings of the 34th International Conference on Machine Learning-Volume 70[Z]. 2017.

[6] KIPF T N, WELLING M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

[7] SCHLICHTKRULL M, KIPF T N, BLOEM P, et al. Modeling relational data with graph convolutional networks[C]//European semantic web conference. 2018: 593-607.

[8] HOWARD A G, ZHU M, CHEN B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017.

[9] RISTOSKI P, VRIES G K D D, PAULHEIM H. A collection of benchmark datasets for systematic evaluations of machine learning on the semantic web[C]//International semantic web conference. 2016: 186-194.