基于多任务层投影 ERNIE 的短文本实体链接及实体分类方法*

何长鸿[1]，孙承杰[1]，林磊[1]，单莉莉[1]，张根宇[1]

哈尔滨工业大学

changhong.he@hit.edu.cn

{cjsun,linl,shanll}@insun.hit.edu.cn

摘要 针对中文短文本实体链接任务，将实体链接和 NIL 实体概念分类作为两个子任务。基于百度 ERNIE 和 BERT-PALs，使用飞桨框架实现多任务模型层，即层投影 ERNIE。该模型节省多任务场景下模型参数量的同时，提升模型在多任务下的泛化性能。任务一利用待链接实体上下文以及目标实体描述信息，计算待链接实体与目标实体的关联度，提升实体链接的准确率。任务二利用实体所在文本以及实体位置信息，进行 NIL 实体分类。使用预测缓存在验证集上搜索关联度最佳阈值判断待链接实体是否为 NIL 实体。在 CCKS2020 短文本实体链指任务的测试集上，单模型 f1 值为 0.877，通过模型融合达到 0.882。

Keywords: 层投影 ERNIE · 多任务 · 实体链接 · 实体分类

1 引言

从海量的文本数据中抽取结构化的知识，用规范的结构存储，从而构建出的知识库，有效的推动了自然语言处理领域的研究进展。知识库相关技术在智能问答、对话系统、搜索引擎等领域都展现出较大的潜力。因此，近年来关于如何构建高可靠性的知识库以及准确高效地利用已有知识库等问题被学术界广泛关注。而实体链接在知识库的构建和使用阶段都具有关键作用。

通常将文本中出现的需要链接的每一个实体称为一个实体提及。给定一个由实体集合 E 构成的知识库，以及一个包含实体提及集合 M 的文本集，且每一个实体提及已在文本中标记出其位置，实体链接需要将每一个实体提及 $m \in M$ 映射到知识库中的某一个实体 $e \in E$ 中，如果 E 中不存在 e 与 m 匹配，则将 m 标记为 NIL（Not In the List）实体 [1]。

* Supported by AIStudio

由于自然语言表达的复杂性，尤其是“一词多义”以及“多词同义”等特点，导致实体链接在候选实体生成、名词歧义以及 NIL 实体的判断等存在难点。考虑人类判断实体含义的方式主要是通过理解实体上下文的语义信息，因此构建一个能够理解文本语义的模型是解决以上难点的一种关键方法。百度发布的 ERNIE 中文预训练模型 [2]，在 BERT[3] 的基础上针对中文的特点提出了针对实体或短语掩盖等新的预训练方式，并通过在中文命名实体识别、情感分析等多项任务上进行实验证明了其强大的语义解析能力。

本文的主要工作包括：

1. 基于飞桨深度学习框架以及 ERNIE，结合 BERT-PALs 的思想，实现层投影 ERNIE，通过在原模型每一层额外引入少量任务相关的参数，用于学习各个子任务之间的差异。在实现多任务模型参数共享的同时，在各个子任务上取得较好的效果。本文将 CCKS2020 实体链指任务划分为候选实体排序、NIL 实体细粒度分类两个子任务。
2. 其中候选实体排序的方式为：训练阶段对每一个实体提及 m，确定对应的候选实体集合为 E_m 后，用每一个候选实体 $e \in E_m$ 与实体提及组成一个样本对 $< m, e >$，并对其进行二分类。预测时用 $tanh$ 激活函数输出的数值大小作为 $< m, e >$ 关联度。
3. NIL 实体细粒度分类任务，对每一个实体提及，在短文本中实体提及两端插入位置标记后作为一个分类输入样本，分类结果即为被标记的实体提及的类别。将模型预测的关联度和 NIL 实体细粒度类别缓存后，通过在激活函数值域内尝试不同的阈值，搜索使得验证集 f1 值达到最高的阈值作为最佳阈值，该阈值用于判断一个实体是否为 NIL 实体。

2 相关工作

2.1 实体链接基本方法

目前关于实体链接已经有较长时间的研究积累，这些方法基本由如下三个步骤构成：

1. 候选实体集生成。根据实体提及字符串与知识库中所有实体的字符串按照一定规则匹配，生成候选实体集。良好的候选实体集要在保证覆盖目标实体的同时，尽可能的小 [4]。
2. 候选实体排序。通过算法比较待链接实体与候选实体集中每个实体中的关联度，这个过程通常需要利用实体提及的上下文以及候选实体的描述

信息。候选实体排序方法包括有监督和无监督两大类，其中有监督方法有二分类 [5]、排序学习 [6]、基于图 [7] 的方法等。本文使用基于二分类的排序方法。

3. NIL 实体判断。根据候选实体排序结果，如果候选实体集为空或者排名较高的几个候选实体均不能匹配，那么就将实体预测为一个 NIL（Not In the List）实体。

对于 CCKS2020 中文短文本实体链接任务[1]，除了需要进行以上实体链接任务外，还需要对 NIL 实体进行细粒度分类，也就是本文3.3节描述的子任务二。

2.2 预训练模型

在候选实体排序、NIL 实体判断及其分类阶段，如何利用实体提及上下文以及知识库实体描述等信息是影响实体链接准确率的关键因素。对于不同知识库，实体描述信息可能有明显差异。CCKS2020 使用的百度百科知识库，每一个实体包含一系列的属性和属性值，其中属性值均为非结构化的描述文本。

谷歌提出的 BERT，通过在大规模无标注语料上进行预训练，并在下游任务上进行微调的方式，在多项任务上逼近甚至超越人类水平，证实预训练能够给模型带来显著的增益效果，使模型学习到非结构化文本中的语义信息。百度基于 BERT，提出以实体或短语为掩盖基本单元的预训练方式，能够更好的学习到中文的语义和语法信息，基于此开源的 ERNIE 在多项中文评测任务上超越了这之前的各项工作。

2.3 多任务学习

多任务学习的目标是通过在多个任务间共享模型参数，提升模型泛化性能，即学习一个能够在多个任务上都能有良好表现的模型，同时也能减小模型的总参数量。微软提出 MT-DNN 模型，以 BERT 为基础模型，通过在最后一层接任务相关的额外层来实现多任务学习 [8]，并展现出了多任务学习在一定程度上给部分子任务带来增益。文献 [9] 提出在模型内部每一层，对每个子任务添加具有少量额外参数的特殊层来实现多任务学习，并通过计算和实验表明这种方式具有更高的计算效率以及更好的泛化性能。

[1] http://sigkg.cn/ccks2020/?page_id=69

3 基于多任务学习和层投影 ERNIE 的实体链接

3.1 层投影 ERNIE

百度开源的中文预训练 ERNIE 模型的基本结构为 12 层堆叠的 Transformer 作为编码器，如图1所示。每一层的输入和输出均为 [序列长度 × 768] 的向量。本文根据文献 [9]，针对多任务场景对 ERNIE 的结构进行修改。多

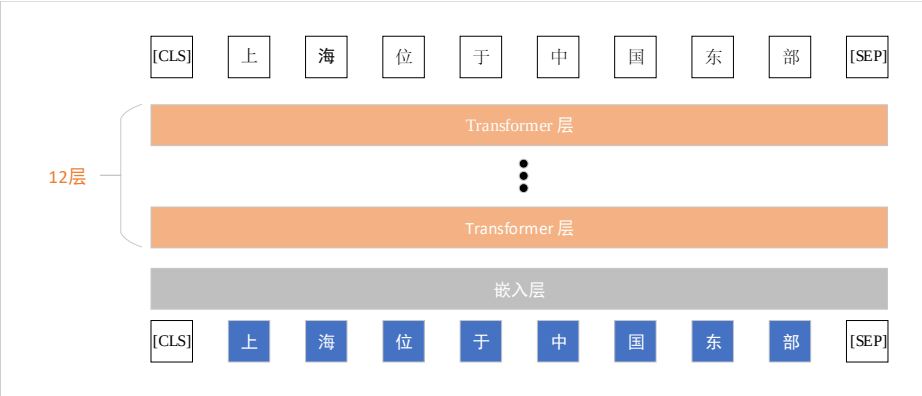

图 1. ERNIE 中的编码器由 12 层堆叠的 Transformer 构成

任务下的自然语言处理任务，各个子任务之间具有非常高的关联，因为它们遵循相同的人类语言的语法和语义规律。文献 [10] 表明 BERT 自底向上分别学到语言的词法、语法、语义等不同信息，因此只需要在每一层引入少量的参数来学习各个子任务之间的差异。

设 n 个子任务构成集合为 $\{task_1, task_2, \cdots, task_n\}$，如图2，在 ERNIE 模型每一层增加 n 个降维层 $\{FC_{D1}, FC_{D2}, \cdots, FC_{Dn}\}$ 以及 n 个升维层 $\{FC_{U1}, FC_{U2}, \cdots, FC_{Un}\}$。理论上降维层和升维层实现方式并不受限制，本文实验使用的为普通的全连接层。模型进行任务 $task_i (i \in n)$ 的训练或推理时，只有 FC_{Di} 和 FC_{Ui} 参与运算。其中全连接层 FC_{Di} 输入神经元个数为 768，这与 ERNIE 的隐藏层大小相等，输出神经元的个数为 $h (h < 768)$，本文实验取 $h = 200$。记模型第 j 层的输出为 L_j，当前输入任务为 $task_i$ 时，每一层的计算方式如式 (1)。其中，Transformer 为多头自注意力，具体细节在文献 [11] 中。

$$L_j = Transformer(L_{j-1}) + Project(L_{j-1}) \tag{1}$$

其中 $Project(L_{j-1})$ 等价于式 (2)

$$Project(L_{j-1}) = FC_{Ui}(FC_{Di}(L_{j-1})) = W_2 \cdot f(W_1 L_{j-1} + b_1) + b_2 \quad (2)$$

W_1、W_2 为全连接层权重参数，大小分别为 $768 \times h$ 以及 $h \times 768$。b_1、b_2 为偏置项，$f(\cdot)$ 为非线性激活函数，此处与 ERNIE 默认激活函数一致，使用 $relu$ 激活函数。

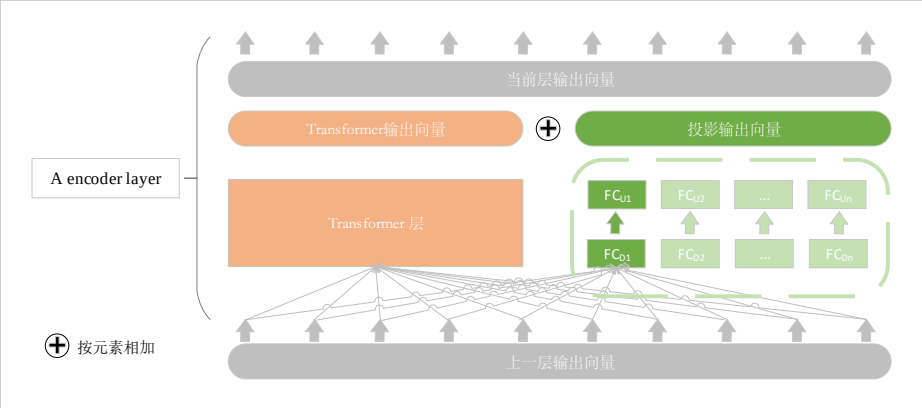

图 2. 虚线框中为在 ERNIE 基础上增加的任务相关的投影结构，图为 $task_1$ 运行时的状态示意。

3.2 子任务一：通过二分类进行候选实体排序

本文基于 CCKS2020 数据集构建，该数据集提供的知识库中的实体均包含丰富的别名，因此在候选实体集生成阶段，对知识库中的每一个实体 e，如果实体名称与别名列表中包含实体提及 m，则将 e 加入到 m 的候选实体集。在训练集上，通过这种方式生成的候选实体集，能完全覆盖标注的目标实体。

对数据集中全部实体提及构成的集合 M 中的每一个元素 m，在知识库中搜索得到它的候选实体集 E_m 后，将实体提及与候选实体集中的每一个实体两两组合，每一个组合 $< m, e_m >$ 为一个输入样本。训练阶段，如果组合中的候选实体与训练数据中的标签一致，则将当前组合作为正例，正例标签值为 1，该实体提及构成的其他组合作为负例，标签值为-1，模型对这些组合进行二分类。使用 $tanh(\cdot)$ 激活函数以及均方误差损失函数。

图 3. 在 ERNIE 嵌入层的基础上增加了目标实体类型嵌入，该嵌入仅在候选实体排序任务中生效。对于同一个样本中的每个位置，嵌入向量相同。这里的 Loc 表示候选目标实体“中国”的类别为 Location

如图3所示，对于每一个组合，除了 ERNIE 原有的三种嵌入信息外，增加一个嵌入层，该嵌入层使用候选目标实体的类别进行特征嵌入。输入为实体提及源文本和候选目标实体描述信息拼接成的文本序列。对于源文本，在当前处理的实体提及两端插入 # 作为位置标记，而候选实体描述信息包括知识库中的别名、属性名称以及属性值。

3.3 子任务二：NIL 实体分类

对训练集中每一个实体提及，如果为 NIL 实体，则以对应的标注 NIL 类别作为标签，否则将其目标实体的类别作为该实体提及类别标签。和候选实体排序任务中的预处方法类似，对每一个实体提及在其前后分别插入 # 作为位置标记，作为一个分类样本。如果一句话中出现多个实体提及，则分别处理为多个样本，使模型能够学习到句子中不同位置上的实体的差异。模型最后一层接 $softmax(\cdot)$ 激活函数，对每一个样本进行分类，分类结果作为当前被标记的实体提及的细粒度类别。

3.4 训练及推理过程

在 ERNIE 编码器中的每一层添加多个全连接层，作为任务相关的投影层。每个任务输入需要同时提供当前任务的编号，图2表示当前输入为 $task_1$ 时的运行示意图，只有当前任务编号对应的投影层，即 FC_{D1} 和 FC_{U1} 会进行前向计算。投影层的输出与 Transformer 的输出维度相同，通过简单的相

加后作为当前编码器层的输出。在反向传播过程中，根据飞桨框架的自动求导机制，每一次迭代只有参与计算的投影层的参数会被更新，因此一个任务的推理或训练不会影响其他任务的投影层，这使得模型的任务数目可以方便的扩展。

多任务训练时，每个 batch 使用随机采样，同一个 batch 中的样本来自一个任务。对于训练样本数目分别为 $|X_1|, |X_2|, \cdots, |X_n|$ 的任务集合，每个 batch 选择第 i 个任务的概率 p_i 是个超参数。本文默认使用的概率计算方式如式3:

$$p_i = \frac{\sqrt{|X_i|}}{\sum_{i=1}^{n} \sqrt{|X_i|}} \tag{3}$$

4 实验与分析

4.1 实验背景

本文为 CCKS2020 中文短文本实体链指竞赛评测论文，所有数据集均为比赛提供的官方数据，该数据集训练集大小为 70000 条样本，每一条样本中可能有一个或多个待链接的实体，共计 266645 个。验证集包含 10000 条样本，共计 33053 个待链接实体。实验环境为 aistudio 平台。本文的模型使用飞桨框架 1.8 版本构建。

4.2 最佳阈值搜索

预测阶段，先对每一个实体提及 m 的候选实体进行排序，候选实体的输出得分最高的记为 $score_m$，如果 $score_m$ 大于阈值，则将该实体提及链接到得分最高的候选实体，否则作为 NIL 实体，对其进行 NIL 概念分类。

为了寻找最佳阈值，使用模型对验证集进行预测，同时保留各个实体提及 m 以及对应排名第一的候选实体编号 e_{best} 及其分值 s_{best}，再对该候选实体进行 NIL 概念分类，并记录概率最高的类别 t。由此每个实体提及生成一个四元组 $T_m = (m, e_{best}, s_{best}, t)$。在验证集上使用公式4进行阈值搜索，其中 S 为阈值。

$$result(T_m, S) = \begin{cases} e_{best} & , (s_{best} \geq S) \\ t & , (s_{best} < S) \end{cases} \tag{4}$$

由于 $tanh(\cdot)$ 激活函数输出的预测分值范围为 $(-1, 1)$，以此为搜索边界，0.05 或者更小的值为搜索步长，对每一个阈值通过公式4得到一个验证集预测结果后，计算其 f1 分值。

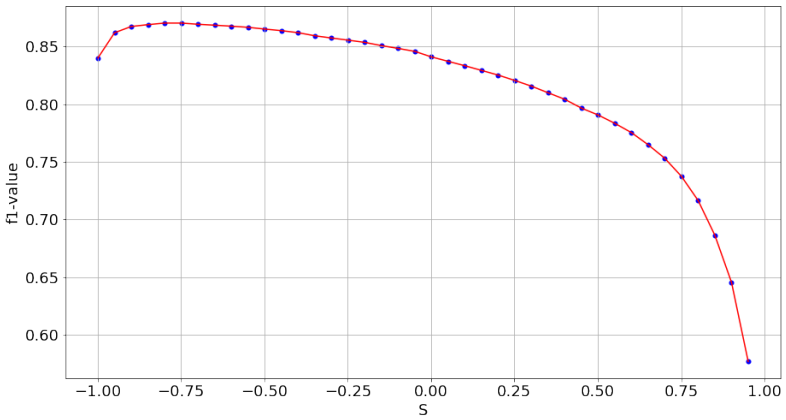

图 4. 使用 0.05 作为搜索步长，图中阈值为-0.76 左右时，验证集达到最高的 f1 值

对于不同的超参数，训练出的模型通常有不同的最佳阈值。通过本文的方法，对于 10000 个样本的验证集，每次计算时间小于 1 秒，因此可以以较小代价对各个版本的模型搜索对应的最佳阈值。图4为某个模型在验证集上的 f1 值关于不同阈值的折线图，由于候选实体排序过程中大多数为负样本，因此最佳阈值通常在 -1 到 0 之间。

4.3 候选实体特征选取

知识库中的每一个实体，包含实体名称、别名列表、多个属性名、属性值、实体类别等信息，并且每个实体的属性至少包括“摘要”或“义项描述”。通分析数据发现，摘要和义项描述能够很好的描述实体，而其他属性包含大量噪声，因此将摘要和义项描述作为基本特征，其他特征信息拼接在这后面。实验选取不同的实体信息进行拼接。表1中的实验结果微调过程中使用的学习率为 $2.5e^{-5}$，批大小为 32-24 之间，迭代步数为 12 万步。可以看到，使用越多的特征，模型准确率会更高，但相应的训练时间也会大大增加。仅仅使用“摘要 + 义项描述”的训练时间大约只有“摘要 + 义项描述 + 属性名 + 属性值 + 实体名 + 别名”的三分之一。

通过将不同特征训练得到的模型进行加权投票，可以进一步提高最终的准确率。以模型在验证集上的 f1 值为加权权重，在比赛测试集上取得的最佳结果 f1 值为 0.88187。

表 1. 不同候选实体特征对实验结果的影响

特征类别	最佳阈值	验证集 f1 值
摘要 + 义项描述	-0.759	0.87096
摘要 + 义项描述 + 属性值 + 别名	-0.739	**0.87420**
摘要 + 义项描述 + 属性值 + 别名 + 属性名 + 实体名	-0.779	0.8721
加权投票融合	-	**0.88187**

5 结束语

本文将 CCKS2020 中文短文本实体链指任务划分为候选实体排序和 NIL 实体细粒度分类两个子任务，并构建了多任务层投影 ERNIE，使用同一个编码器分别训练和预测这两个子任务，在比赛中最终取得第 8 名。本文实现的模型可以将一个基于 ERNIE 的编码器扩展到更多的子任务上，并且几乎不受限制。

在具体任务方案上，本文仍有改进空间，对于 NIL 实体细粒度分类任务，一方面可以使用序列标注的方法，使得一句话中存在多个实体的情况下的计算量进一步缩减，但经过我们的尝试，准确率有略微下降，因此该方案并未被我们采纳。此外，虽然在文献 [8][9] 中表明了预训练模型的共享参数在多任务场景下会有明显优势，但本文的模型并未经过充分实验证明，其扩展到更多子任务上的效果有待进一步实验。

参考文献

1. Shen W, Wang J, Han J. Entity linking with a knowledge base: Issues, techniques, and solutions[J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 27(2): 443-460.
2. Sun Y, Wang S, Li Y, et al. Ernie: Enhanced representation through knowledge integration[J]. arXiv preprint arXiv:1904.09223, 2019.
3. Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.
4. 杨子翰. 基于多源信息的实体链接算法研究及应用 [D]. 电子科技大学,2020.

5. Zuheros C , Tabik S , Valdivia A , et al. Deep recurrent neural network for geographical entities disambiguation on social media data[J]. Knowledge Based Systems, 2019, 173(JUN.1):117-127.
6. 詹飞, 朱艳辉, 梁文桐, 冀相冰. 基于 BERT 和 TextRank 关键词提取的实体链接方法[J]. 湖南工业大学学报,2020,34(04):63-70.
7. Moro A, Raganato A, Navigli R. Entity linking meets word sense disambiguation: a unified approach[J]. Transactions of the Association for Computational Linguistics, 2014, 2: 231-244.
8. Liu X, He P, Chen W, et al. Multi-task deep neural networks for natural language understanding[J]. arXiv preprint arXiv:1901.11504, 2019.
9. Stickland A C, Murray I. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning[J]. arXiv preprint arXiv:1902.02671, 2019.
10. Jawahar G, Sagot B, Seddah D. What Does BERT Learn about the Structure of Language?[C]. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 3651-3657.
11. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]. Advances in neural information processing systems. 2017: 5998-6008.