软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2023,34(3):1010−1026 [doi: 10.13328/j.cnki.jos.006787]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

面向列语义识别的共现属性交互模型构建与优化*

高 珊[1,2], 袁宛竹[1,2], 卢 卫[1,2], 王 兰[1,2], 张 静[1,2], 杜小勇[1,2]

[1](数据工程与知识工程教育部重点实验室(中国人民大学), 北京 100872)

[2](中国人民大学 信息学院, 北京 100872)

通信作者: 卢卫, E-mail: lu-wei@ruc.edu.cn; 杜小勇, E-mail: duyong@ruc.edu.cn

摘 要: 政务数据治理正在经历从“物理数据汇聚”到“逻辑语义汇通”的新阶段. 逻辑语义汇通是指针对各孤岛政务系统因长期“自治”而形成的元数据缺失、元数据同名不同义以及同义不同名等问题, 在不重建或修改原系统代码以及不物理汇聚各政务数据的前提下, 通过技术手段, 统一各孤岛信息系统元数据的语义表达, 实现元数据的语义互联互通. 该工作是将各孤岛信息系统的元数据语义对齐到已有的标准元数据上, 具体地, 将标准元数据名称看作语义标签, 对孤岛关系数据的列投影进行语义识别, 从而建立列名和标准元数据的语义对齐, 实现孤岛元数据标准化治理. 已有基于列投影的语义识别技术无法捕捉到关系数据的列顺序无关性特征以及属性语义标签之间的相关性特征, 针对这一问题, 提出了基于预测阶段和纠错阶段的两阶段模型: 在预测阶段, 提出了共现属性交互的 CAI 模型(co-occurrence-attribute-interaction model), 利用并行化的自注意力机制保证列顺序无关的共现属性交互; 在纠错阶段, 结合语义标签之间的共现性, 通过引入纠错机制(correction mechanism), 优化 CAI 模型预测结果. 在政务基准数据和 Magellan 等多组公开英文数据集上进行了实验, 结果表明, 引入纠错机制的两阶段模型, 在宏平均和加权平均两个指标上, 比已有最优模型最多可分别提高 20.03%, 13.36%.

关键词: 孤岛政务; 逻辑语义汇通; 列语义识别; 共现交互; 注意力机制

中图法分类号: TP311

中文引用格式: 高珊, 袁宛竹, 卢卫, 王兰, 张静, 杜小勇. 面向列语义识别的共现属性交互模型构建与优化. 软件学报, 2023, 34(3): 1010–1026. http://www.jos.org.cn/1000-9825/6787.htm

英文引用格式: Gao S, Yuan WZ, Lu W, Wang L, Zhang J, Du XY. Construction and Optimization of Co-occurrence-attribute-interaction Model for Column Semantic Recognition. Ruan Jian Xue Bao/Journal of Software, 2023, 34(3): 1010−1026 (in Chinese). http://www.jos.org.cn/1000-9825/6787.htm

Construction and Optimization of Co-occurrence-attribute-interaction Model for Column Semantic Recognition

GAO Shan[1,2], YUAN Wan-Zhu[1,2], LU Wei[1,2], WANG Lan[1,2], ZHANG Jing[1,2], DU Xiao-Yong[1,2]

[1](Key Laboratory of Data Engineering and Knowledge Engineering (Renmin University of China), Ministry of Education, Beijing 100872, China)

[2](School of Information, Renmin University of China, Beijing 100872, China)

Abstract: Government data governance is undergoing a new phase of transition from “physical data aggregation” to “logical semantic unification”. Thus far, long-term “autonomy” of government information silos, lead to a wide spectrum of metadata curation issues, such as attributes with the same names but having different meanings, or attributes with different names but having the same meanings. Instead of either rebuilding/modifying legacy information systems or physically aggregating data from isolated information systems, logical semantic unification solves this problem by unifying the semantic expression of the metadata in government information silos and

* 基金项目: 国家重点研发计划(2020YFB2104101)
本文由“大数据治理的理论与技术”专题特约编辑杜小勇教授、杨晓春教授和童咏昕教授推荐.
收稿时间: 2022-05-15; 修改时间: 2022-07-29; 采用时间: 2022-09-23; jos 在线出版时间: 2022-10-27

achieves the standardized metadata governance. This work semantically aligns the metadata of each government information silo to the existing standard metadata. Specifically, the standard metadata names are viewed as semantic labels, and the semantic meanings of columns of relations in each government information silo are semantically identified, so as to establish the semantic alignment of column names and standard metadata and achieve standardized governance of silo metadata.

Key words: government information silo; logical semantic unification; column semantic recognition; co-occurrence interaction; attention mechanism

从 20 世纪 90 年代起, 以“政府部门的业务信息化建设”为中心, 我国启动了各类“金字工程”建设, 包括金税工程、金贸工程、金材工程、金卡工程等. 由于我国国土面积辽阔, 不同区域、不同行业、不同层级、不同业务之间存在较大差异, 难以实现信息化统筹建设. 政务数据被分割存储在不同部门的信息系统中, 无法实现互联互通、互相分享、整合利用, 难以为老百姓提供高效与便捷的公共服务, 造成“信息孤岛”[1]现象. 近年来, 国家多次发布文件, 要求“利用大数据打通政务信息孤岛”. 响应国家号召, 地方政府成立了“大数据局”等机构, 解决部门“信息孤岛”问题.

打破“信息孤岛”, 目前有两个典型的做法.

1) 物理数据汇聚. 围绕部门信息系统, 由大数据局等机构重构“原业务系统表单”, 用户办理业务填写的表单事项数据, 先汇聚到大数据局, 再流转到原职能部门进行后续的业务处理. 通过重构业务表单事项, 统一规范化事项名称, 从而完成信息孤岛数据治理. “物理汇聚”各孤岛信息系统数据, 在一些省份取得了较好的效果. 但也存在着一些弊端, 例如, 重构业务表单需要重新梳理原业务系统, 涉及大量人工成本和开发成本; 巨量数据物理汇聚到同一个地方, 造成额外管理和运维成本、以及可能存在的数据安全等问题.
2) 逻辑语义汇通. 与物理数据汇聚相反, 逻辑语义汇通不是把孤岛信息系统中的数据物理汇聚到同一地方, 而是通过构建标准标签体系, 梳理各信息系统中的元数据并关联到标准标签体系, 实现孤岛信息系统的元数据标准化治理. 由于无需物理汇聚数据, 也不需要重建或修改原业务系统代码, 逻辑语义汇通是当前政务信息系统元数据标准化治理比较理想的做法. 但该方法对技术要求比较高, 需要构建自动化的元数据到标签映射机制, 可支持政务系统元数据缺失、元数据同名不同义以及同义不同名等问题的解决.

本文聚焦逻辑语义汇通技术, 基于政务服务信息资源的国家标准、地方标准、行业标准构建的元数据标签体系[2], 对孤岛信息系统关系数据的列投影进行语义识别, 建立元数据(列名称)到标准元数据的语义对齐, 实现孤岛元数据标准化治理.

对关系数据列投影进行语义识别, 也叫列语义识别, 主要是对关系列进行语义表示, 映射到已知的语义类别. 其中, 关系数据的属性值和上下文信息对关系列的语义识别有着重要作用, 比如, 对于某一列属性值, 包括“腾讯、浙江省某票务公司、字节跳动”等, 那么这一列的语义很大概率和企业相关, 比如“企业名称”“参保单位”“纳税单位”等, 但仅仅依赖单独列的属性值无法准确区分其语义, 事实上, 关系表的上下文信息可以帮助进一步确定列的语义信息. 例如, 对待识别列, SATO 模型[3]仅仅考虑待识别列的相邻两列作为局部上下文, 并使用 CRF 模型编码列间关系进行约束; 而 SCA 模型[4]则是将同一关系主题下的其他所有列按行拼接构成一段线性化文本, 作为待识别列的上下文, 在预训练模型 BERT[5]上微调直接进行分类, 将待识别列映射到已知的语义标签.

然而, 当前基于关系表上下文的研究中存在的问题有:

- 首先, 忽略了关系数据具有列顺序无关的特性. 如图 1 列顺序交换的两张表, 任意交换关系表中列的位置, 都不影响关系列和整个关系主题的意思表达. 不同的信息系统对同一关系表往往会以不同的列顺序进行存储, 如果模型对列顺序敏感, 可能会导致对同一关系表由于各列顺序发生变化而输出错误的语义类别, 模型预测准确率下降, 缺少泛化性.
- 其次, 现有研究关于上下文的考虑仅限于利用关系表层面或者关系属性值层面的上下文对单独列进

行类别预测，却忽略了属性语义标签之间的上下文关系.

纳税单位	纳税金额	纳税项目	纳税日期
腾讯	42403	行为税类	17/5/2016
中国人民大学	58005	财产税类	19/1/2017
字节跳动	345	财产税	19/8/2016

纳税日期	纳税单位	纳税项目	纳税金额
17/5/2016	腾讯	行为税类	42403
19/1/2017	中国人民大学	财产税类	58005
19/8/2016	字节跳动	财产税	345

图 1　列顺序交换的两张表

如图 2 所示，即使考虑了关系表级别的上下文，仍然是对各个列进行单独预测，没有考虑到语义标签之间的相关性. 比如在这张表中，“纳税单位、纳税金额、纳税项目”和“参保日期”显然不应该出现在同一个关系表中. 在此假设下，当预测出周围的语义标签是“纳税单位、纳税金额、纳税项目”时，进一步结合语义标签之间的相关性，将最后一个标签纠正为更符合整个关系主题意思的“纳税日期”，而不是“参保日期”.

纳税单位	纳税金额	纳税项目	参保日期
?	?	?	?
腾讯	42403	行为税类	17/5/2016
中国人民大学	58005	财产税类	19/1/2017
字节跳动	345	财产税	19/8/2016

图 2　纳税信息表中的“参保日期”预测错误

针对以上问题，本文提出了基于预测阶段和纠错阶段的两阶段模型构建与优化方法：首先，针对关系数据的上下文信息和列顺序无关性特征，提出了共现属性交互的 CAI 模型，利用顺序无关的自注意力(self-attention)机制实现关系数据中待识别属性与共现属性的交互，以更高效的方式对待识别属性融入关系表级别的上下文信息；其次，针对属性语义标签之间的相关性特征，引入纠错机制，利用 Transformer 模型学习到同一关系主题下属性语义标签之间的相关性，对 CAI 模型关于单列预测的初步结果进一步优化，提高对关系表中各个列的最终语义识别效果.

本文的主要贡献有：

(1) 提出了面向列语义识别的两阶段模型(CAI-correction model)构建与优化方法：在预测阶段，提出共现属性交互的 CAI 模型(co-occurrence-attribute-interaction model, CAI)，利用自注意力机制，让待预测列学习到关系表级别的上下文信息，在保证关系数据列顺序无关的前提下，解决对关系列语义预测过程中存在的歧义问题.

(2) 在纠错阶段，引入基于语义标签共现的纠错机制(correction mechanism)，利用并行化且顺序无关的 Transformer 模型[6]学习到语义标签之间的共现依赖性，进一步优化模型预测结果.

(3) 在政务基准数据集和两个公开英文数据集上进行对比实验，实验结果表明，CAI-correction 模型在宏平均和加权平均上，比已有最优模型最多可分别提高 20.03%, 13.36%.

本文第 1 节是对研究的问题进行定义. 第 2 节介绍列语义识别的相关工作以及本文用到的注意力机制. 第 3 节介绍 CAI-Correction 模型的整体框架和具体细节. 第 4 节为实验部分，将本文提出的模型与基准模型进行实验对比与分析，验证所提模型的有效性. 最后对全文进行总结.

1　问题定义

对于一个表 T 有 n 行 m 列，表 T 可以由列的集合 $Col=\{col_1,col_2,\ldots,col_m\}$表示，列也叫表的属性；表 T 也可以由行的集合 $Row=\{row_1,row_2,\ldots,row_n\}$表示，一行也称作元组. 列与行的交汇构成单元格，每一个单元格的

值也叫属性值. 因此, 每一列 Col_i 也可以表示为该列的属性值集合, 即 $Col_i=\{cell_{1i},cell_{2i},\ldots,cell_{ni}\}$. 列语义识别是指将关系表中各个列映射到语义类别的过程, 在考虑到关系表级别的上下文时, 我们将这个问题看作关于每列的多分类问题. 训练数据由不同的表构成, 语义类别是事先定义好的标签. 特别地, 在政务基准数据中, 政府发布的地方标准性文件为各政务领域的元数据提供了规范标准. 标准文件中的元数据命名为 $S=\{S_1,S_2,\ldots,S_k\}$, k 表示不同元数据个数. 这些标准元数据名称是政务基准数据待映射的语义类别,如图 3 所示. 注意: 假设所有关系列都可以映射到语义类别集合中的某一个具体类别, 即真实的列类别可以被语义类别集合完全覆盖, 不考虑真实列类别不在事先定义的语义类别集合中的情况.

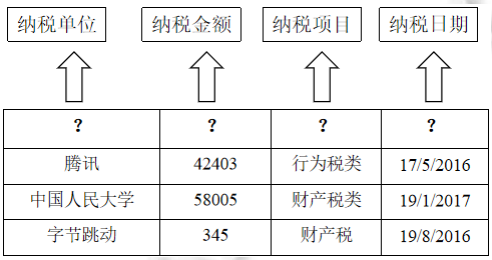


图 3 政务关系表到语义类别的映射

列语义识别是对每一列 $Col_i=\{cell_{1i},cell_{2i},\ldots,cell_{ni}\}$ 找到映射 f, 将其对齐到对应某个正确的语义类别 S_j, 即 $f(Col_i)=f(cell_{1i},cell_{2i},\ldots,cell_{ni})=S_j$. 这样, 我们的问题形式化为: 对于一张关系表 T, 对其中每一列 Col_i, 都能映射到语义标签集合 S 中的对应类别 S_j.

2 相关工作

本节主要介绍列语义识别的现有工作, 包括基于知识库、基于统计特征和基于深度学习语义表示. 除此之外, 我们还介绍了本文模型构建用到的注意力机制.

2.1 基于知识库的列语义识别

Venetis 等人[7]建立了属性值到预定义标签集的映射词典, 然后, 关于各属性值类型的最大投票数决定属性标签. 现有的外部知识库为关系数据的语义识别提供了好的支持, 比如 DBPedia[8]、Freebase[9]等本体或知识库, 现有工作可以利用这些外部知识库将语义标签分配给表中的各个元素(SemTab 2019[10]), 比如直接基于这些知识库自带的 lookup 服务进行查询, 来获取和关系列相匹配的语义类型[11−13]. AMALGAM[14]也是利用外部知识库的实体链接对关系列中的所有项进行注释, 然后利用随机实体链接来估计属性标签. MTab[15]则是结合投票算法和概率模型来解决关系数据到知识图匹配的关键问题, 通过对带有语言参数的多个服务执行实体查找和采用文字列匹配来找到相关属性. T2KMatch[11]结合了模式匹配和实体匹配的迭代匹配方法, 利用实体匹配和模式匹配结果的相互促进, 可以实现将网页中表的列与 DBPedia[8]的已有属性之间的对齐.

2.2 基于统计特征的列语义识别

早期的方法中, 也有利用统计特征表示数据相似度来匹配列的语义类型的做法. Ramnandan 等人[16]利用启发式方法将数值列和文本列进行类型区分, 然后分别使用 Kolmogorov-Smirnov(K-S)检验和 TF-IDF 方法进行列值比较, 从而检测样本的语义标签. Pham 等人[17]在此基础上加入了额外的特征和检验, 包括关于数值数据的 Mann-Whitney 检验和关于文本数据的 Jaccard 相似性, 来训练逻辑回归和随机森林模型.

除此之外, 现有许多方法将语义类型检测定义为关于列语义识别的多分类问题, 将语义类型作为标签. 有用的特征对于理解表的语义很重要, Chen 等人[18]假设一个模式标签和从列内容中观察到的特征是高度相关的, 将每个列值的每个字符都看作一个特征, 并从关系列的整体内容中获取更高级别的特征, 包括在每一列上提取 13 种单列特征, 例如字符数单元格的长度、列的最大值、最小值等, 用来描述不同的模式标签, 使标

签相似性取决于内容的相似性, 而不是不规范列名的表面形式.

Hulsebos 等人[19]提出的 Sherlock 模型则从关系数据的列中抽取出更多的统计特征, 定义了 27 维的全局统计特征, 比如列熵(column entropy), 用以区分属性值中含有重复值的特性; 又比如, 对于数值类型属性列的均值、最大值、最小值等, 还有字符级别的特征分布(character-level distribution), 通过将属性列中每一个字符转化为对应的 ASCII 字符, 并进行计数、求平均、求和等操作, 关于每一列提取出 960 维的特征; 再利用训练好的 Glove 字典得到属性列中各个词的词向量, 同时也根据分布式的词袋模型得到主题向量, 最后, 利用多层感知机(MLP)网络进行分类. 该方法在列的语义识别上达到了不错的效果.

但 Sherlock 模型[19]只使用了列值来预测其类型, 而没有考虑到列在表格中的上下文信息等, 这无法解决歧义问题. 例如, 对于一个包含“佛罗伦萨”“伦敦”“纽约”的列, 地点、城市或出生地都可以是该列的语义类型. 因此, Zhang 等人[3]在 Sherlock 模型的基础上提出了一种包含表上下文的混合机器学习模型 SATO, 将结构化学习与基于 Sherlock 模型的单列语义类型预测相结合, 同时考虑关系表的主题信息和局部上下文信息, 利用条件随机场 CRF 模型[20]限制同一表格中的标签序列.

2.3 基于深度学习语义表示的列语义识别

随着深度学习技术的发展, 越来越多的工作开始用深度学习模型对关系列进行语义表示, 然后映射到对应的语义类别. 元数据信息, 例如表名、列名和表结构对列内容的理解至关重要, 我们可以基于表的元数据信息和对应真实类别进行相似度匹配. 这个工作类似于知识图谱对齐或实体对齐, 可以同时考虑到实体的名称和结构相似度进行匹配[21−23]. 但在实践中, 对于不同信息系统的关系表, 往往存在元数据信息缺失、不完整或不明确等问题, 为了解决这种依赖, 关系表中丰富的实例信息可以有助于对列进行特征提取和语义识别.

Chen 等人[24]利用 word2vec 模型将列的内容映射到词向量空间, 然后利用 CNN 模型和单词的语义表示学习单元格级别和列级别的语义表示, 实现关系列到知识库中候选类的映射. Ding 等人[4]提出了 CCA 模型, 只考虑了关系表的单列信息, 将待识别列的单元格值按行拼接看作一段句子在预训练模型 BERT[3]上进行微调, 直接分类到对应的语义标签; 并在此基础上提出了上下文感知的 SCA 模型, 该模型是在 CCA 模型基础上进一步考虑到关系表的上下文信息, 将同一表格中的其他列整体按行拼接看作待识别列的上下文, 和线性化的待识别列一起在预训练模型 BERT[3]上进行微调, 让关系列学习到上下文的信息, 增强关系列的语义表示, 从而分类到语义类型. 除此之外, Deng 等人[25]提出了 TURL 预训练模型框架, 通过将表格线性展开, 同时融入表的标题、元数据和按行展开的表内容等信息, 定义可见性矩阵保证关系表的列与列之间若相关则可见, 以此让关系表学到更丰富的上下文表示. 不过, 由于预训练过程考虑了表元数据, 因此对于列类型注释的下游任务来讲, 当表元数据缺失也就是仅仅依赖列的单元格值时, 其效果会变差. 另外, TURL 模型框架本质是做表格的表示学习预训练, 而不是具体的列语义识别, 两者存在差距. 未来, 我们可以考虑将列语义识别作为表格学习预训练的下游任务, 进行进一步探索.

尽管关系表上下文信息的加入可以丰富列的语义表示, 但是关系表不同于一般的文本数据, 是结构化数据, 具有列顺序无关性的特征. 而现有的模型忽略了这个特性, 导致出现列顺序依赖, 交换列的顺序会对模型的预测结果造成较大的干扰.

2.4 注意力机制

注意力(attention)机制[6]最早应用于图像领域, 在图像分类中有很好的效果. 类似于人的视觉功能, 当我们在看某一件东西的时候, 注意点并不会放在这样东西的每一个细节. 事实上, 某一些重要的部分就足够我们区分或辨别不同的东西. Attention 机制可以让模型把注意点放在更重要的位置, 给它们赋以更高的权重. 所以简单来讲, Attention 机制本质就是“权重”参数, 通过对重要的有用的信息给以更大的权重值, 以增强对目标词的语义表达[26]. 在注意力机制中, 随着 BERT 模型[3]的广泛使用, 自注意力机制得到更多的关注, 它可以不依赖外部信息, 直接将句子中每一个词和句子内部的其他词进行相似度匹配, 从而计算出权重,自注意力机制可以捕捉到句子内部各个词之间的依赖关系. 于是, 自注意力机制对于表征上下文信息对目标词的影响有很

好的效果; 同时, Transformer 模型[6]Encoder 部分的自注意力机制可以并行计算, 提升模型训练速度, 并且具有顺序无关性, 本文将充分运用自注意力机制进行模型构建与优化.

3 两阶段的列语义识别模型

3.1 模型基本思想

Harris 的分布式[27]假设提出, 目标词总是与其邻居具有较高的相关性. 因此, 可以基于邻居也就是上下文对目标词进行向量表示. 本文将充分考虑这种相关性, 在关系数据层面, 将待识别属性列的在同一关系主题下的共现属性列看作是待识别属性的上下文. 进一步地, 在语义标签层面, 同一关系主题下的语义类别同样具有共现性. 为了更好地引入这两个层面的共现性, 我们的模型可以分为两个阶段: 预测阶段和纠错阶段, 如图 4 所示.

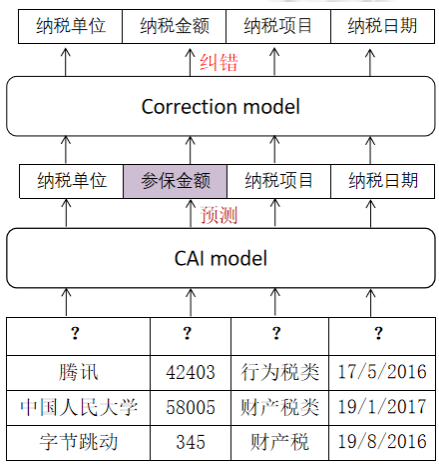


图 4 两阶段的列语义识别过程

其中, 预测阶段的 CAI 模型又分为 3 个模块.

(1) 线性化属性列编码: 针对结构化的关系数据, 将关系列线性化, 利用预训练模型 BERT[3]对线性化的属性列进行编码, 并将 BERT 中的特殊标识符[CLS]对应的输出向量作为初步的列向量.

(2) 共现属性列交互模块: 利用关系表内部属性列之间的共现依赖进行交互, 让每一个列向量都可以学习到其共现属性列的信息, 丰富各个列向量的语义信息.

(3) 分类模块: 对融入共现属性列信息的列向量, 再经过多层感知机网络, 利用 Softmax 函数归一化, 得到输出向量对应每一个语义类别的概率, 从而进行分类.

纠错阶段的 Correction 模型主要是基于预测阶段中 CAI 模型对每一个关系列预测得到的类别标签, 进一步考虑类别标签之间的共现性, 在原预测结果上进行优化, 提升模型最终识别效果.

模型整体框架如图 5 所示.

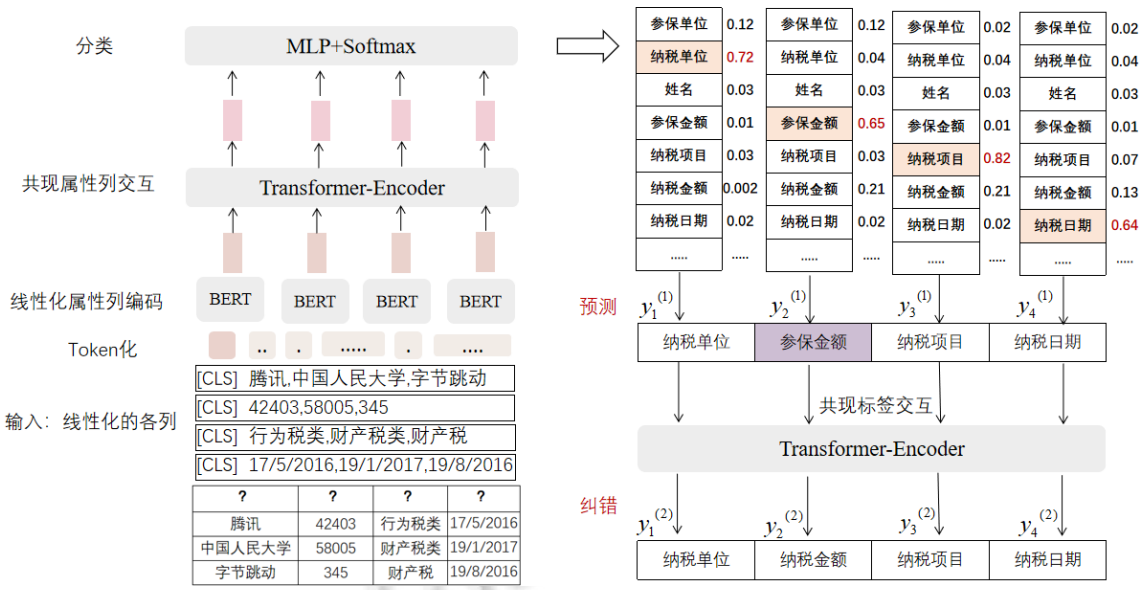


图 5 模型整体框架

3.2 引入关系属性共现的模型预测

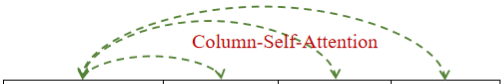
3.2.1 关系表的上下文信息

在自然语言理解问题中，上下文信息可以看作是有助于理解目标词/句子的其他信息. 对于文本数据，目标词的上下文信息可以看作是同一段落中共同出现的其他词，而关系数据不同于一般的文本数据，具有结构性的，由多个行和列构成，而每个行或列又是由许多单元格构成，单元格内部是一些具体的数据. 它们的数据类型有多种，包括文本、数值、日期等. 事实上，对于关系表包含的上下文信息，可以看作是两类.

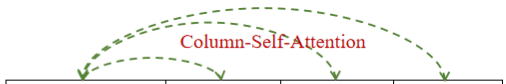
(1) 列级别的上下文：属性列的实例值可以看作是属性名称的上下文信息. 表的字段名称可以看作是对属性值的抽象描述，而表的属性值可以看作是对字段名称的具体描述，有助于对字段名称的理解；同时，这些属性值总是和字段名称共同出现，于是，在同一列中，属性值所表达的语义和字段名称所表达的语义是相近的. 比如，某字段名称是“GSMC”，只看属性名，可能很难理解到其含义；但实例信息为“腾讯、浙江省某票务公司、字节跳动...”时，可以猜到这一属性列表达的含义可能是“公司名称”，也就是“GSMC”表达的含义是公司名称. 由于不同信息系统的关系数据可能存在列名缺失、命名不标准或不一致等问题，基于这种考虑，可以利用关系属性值的语义信息得到整列的语义信息，将其映射到正确的语义类别.

(2) 关系表级别的上下文：同一关系主题下的共现属性可以看作是待识别列的上下文. 这是由于每一张关系表不是任意列随意组合的，一张关系表总是具有一定主题信息，这种主题信息往往通过表名体现出来. 比如图 6 中的两张表，左表是“纳税信息”、右表是“参保信息”. 但是由于设计人员对于同一关系主题的表命名总是不一致或者难获得，于是，关系表的主题信息可以体现在该关系表的共现属性列上，这种属性列之间的共现依赖性有助于对列的语义识别. 这是因为不同关系主题下总存在实例特征一致但语义不同的属性，它们对应的实例特征完全一样，如果仅仅考虑当前属性下的实例信息，无法对它们做出准确区分. 例如，图 6 中都有共同列值“腾讯、中国人民大学、字节跳动”，但是左表中其语义是“纳税单位”，而在右表其语义是“参保单位”. 决定这两个相同列具有不同语义的原因就在于其所在关系主题信息不同，或者它们的共现属性不同. 因此，也可以把这些共现属性看作是待识别列的上下文，以丰富基于属性值表示的待识别列的语义信息. 同样地，图 6 中的两表除了第 1 列“纳税单位”与“参保单位”存在歧义以外，还都涉及金额或者日期类型的属性列，其特征非常

相似, 仅靠单列信息无法识别, 此时, 在不同关系主题下的共现属性可以帮助确定具体语义.



纳税单位	纳税金额	纳税项目	纳税日期
腾讯	42403	行为税类	17/5/2016
中国人民大学	58005	财产税类	19/1/2017
字节跳动	345	财产税	19/8/2016



参保单位	参保金额	保险类别	参保日期
腾讯	12403	养老保险	17/3/2015
中国人民大学	18005	医疗保险	20/9/2013
字节跳动	3002	工伤保险	19/5/2018

图 6 两个关系表下特征相似但语义不同的“纳税单位”和“参保单位”

基于上述关系表不同级别的上下文信息考虑, 我们选择利用关系属性值和共现属性的上下文信息对待识别列进行语义表示, 从而构建模型.

3.2.2 基于关系表列顺序无关性的共现属性交互模型

关系数据不同于一般的文本数据, 其是具有结构性的, 由多个行和列构成, 而各个列之间是具有顺序无关性的, 也就是任意交换关系数据的列的位置都不影响整个关系主题的意思表达. 我们希望在不违背关系数据列顺序无关性的前提下, 让待识别属性融入其共现属性的信息.

受到 Transformer 模型[6]具有顺序无关性的启发, 本文提出了共现属性交互的 CAI 模型, 利用自注意力机制实现待识别属性和共现属性的语义交互, 让待识别属性学到其共现属性, 即表上下文信息的语义特征.

接下来, 将具体讨论预测阶段 CAI 模型的组成. CAI 模型主要分为 3 个部分.

(1) 第 1 部分, 属性列线性化编码.

在这一部分, 参照 CCA 模型[15]对关系列 Col_i 线性化的方式, 将待预测列按行拼接看作一段文本, 例如图 6 中“纳税单位”这一列, 线性化之后的输入文本是“[CLS]腾讯、中国人民大学、字节跳动[SEP]”. 然后,选择 Google 开源的叠加了 12 层 Transformer 编码器的预训练模型 BERT 作为这一模块的基础模型框架, 将输入文本进行分字 Token 化, 然后转为 BERT 的输入表示, 最后将特殊标识符[CLS]对应的输出向量 $c_i^{(1)}$ 作为每一列的初步列向量. 事实上, 关系表同一列的各个单元格之间也存在顺序无关性, 可以利用每列各单元格嵌入的平均池化作为列的最终表示[28]来保证这种顺序无关性. 但由于在训练过程中会对关系表的行进行随机打乱, 通过第 3.2.3 节的损失函数可以保证对同一列, 无论各单元格的顺序如何, 其语义都是一样的, 从而间接使得这种基于 BERT 模型得到的列语义表示是顺序无关的.

特别地, 由于不同表中的行数不同, 在训练时需对所有表打乱行的顺序, 并设置固定最大行数将每一张关系表拆分成多个同样关系主题的小表, 再对各列按行拼接进行线性化. 对线性化后的文本列, 也设置 BERT 的最大序列长度, 当线性化的输入列长度超过该值, 模型会对输入序列自动进行截取; 反之, 则会用 0 填补至最大序列长度, 以保证输入序列的固定长度.

(2) 第 2 部分, 共现属性交互模块.

本文又称其为 Column-Encoder, 用到的自注意力机制来自 Transformer 模型中 Encoder 部分的多头自注意力模块, 我们称其为列注意力 Column-Self-Attention. 其中, 列注意力层一共 3 层, 可以更好地实现同一关系主题下的共现属性交互. 在这一部分, 输入是一张 m 列的表经过第 1 步得到的初步列向量, 也就是 m 维的向量矩阵 $cols\_embedding=[c_1^{(1)},c_2^{(1)},...,c_m^{(1)}]$, 这样, 经过 Column-Encoder 之后, 每一个输入的列向量都有对应的输出向量, 即对一张表仍然输出 m 维的列向量 $cols\_out=[c_1^{(2)},c_2^{(2)},...,c_m^{(2)}]$, 将其作为每个属性列学习到上下文之后的列语义表示. 为了同时考虑列本身的特征和学习到关系表上下文之后的特征, 我们将最终的列语义表示定义为公式(1)所示.

$$cols\_out=Self\_Atten(cols\_embedding)+cols\_embedding \tag{1}$$

(3) 第 3 部分, 分类模块.

在这一部分, 我们将最终融入共现属性信息的列向量 $cols\_out$ 再经过 MLP 网络, 即一层的全连接层和利

用 tanh 的激活函数让模型学习到非线性的特征, 将最终的向量经过 Softmax 归一化, 得到各个列向量对应到映射为每个语义类别的概率, 如公式(2)所示.

$$logits=softmax(W_2\cdot\tanh(W_1\cdot cols\_out+b_1)+b_2) \tag{2}$$

我们对第 1 阶段共现属性交互模型的各个模块进行联合训练, 以实现对模型各参数的联合调整.

3.2.3 损失函数

属性线性化编码模块和共现属性列交互模块可以分开训练, 也可以联合训练. 为了方便, 在我们的模型中采用联合训练, 两个过程可以对模型参数进行联合微调. 于是, 我们的训练目标如公式(3)所示.

$$\max\ P(y_1=S_l,...,y_i=S_j,...|Col_1,...,Col_m)=\max\prod_i P(y_i=S_j|Col_1,...,Col_m)=\max\prod_i\left(\frac{e^{Col_i}}{\sum_j e^{Col_j}}\right) \tag{3}$$

其中, y_i 为 Col_i 的类别变量, S_j 是观察到的类别真值.

因此在训练阶段, 根据 CAI 模型第 3 部分概率计算模块得到的概率值, 我们采用的损失函数是常见的逻辑交叉熵损失函数, 如公式(4)所示.

$$Loss=-\sum_{i=1}^{m}y_i\log(P(\hat{y}_i|Col_1,...,Col_m))=-\sum_{i=1}^{m}y_i\log\left(\frac{e^{Col_i}}{\sum_j e^{Col_j}}\right) \tag{4}$$

3.3 引入语义标签共现的纠错机制

3.3.1 基于语义标签共现的纠错模型

现有的研究中, 关于关系数据的上下文大都是从关系表层面进行考虑, 利用关系属性值的语义信息对各个列进行语义预测. 尽管我们前面从关系属性值角度融入了关系表的上下文信息, 但这种做法仍然是基于关系表上下文信息对列进行单独预测, 忽略了同一关系主题下的语义标签之间的共现依赖性. 例如, 图 6 中左表的标签序列“纳税单位、纳税金额、纳税项目、纳税日期”往往有更大概率会在同一张表里出现, 表达“纳税信息”的语义; 相对地, 右表的“参保单位、参保金额、保险类别、参保日期”有更大概率在同一张表里出现, 表达是“参保信息”的主题信息. 这种标签共现, 是比关系列共现更直观的考虑到上下文信息的方式.

前面提到, 各孤岛信息系统对列的命名总是存在不标准、列名缺失或同名不同义的情况, 但是我们在预测阶段通过 CAI 模型已经初步实现了对各个关系列的语义类别检测, 将关系表的大部分列都可以准确地映射到标准的字段名称. 基于第 1 阶段的预测结果, 本文进一步结合语义标签之间的共现性, 输出更符合在同一关系表出现的语义标签, 实现对第 1 阶段模型预测结果的优化, 达到更好的识别效果.

本文将额外搭建一个共现标签交互的纠错模型, 将第 1 阶段 CAI 模型预测的不完全正确的标签序列 $y^{(1)}$ 映射到更正确的标签序列 $y^{(2)}$.

Transformer 模型的 Self-Attention 机制在表征上下文的共现性上一直表现优秀, 因此在纠错阶段, 我们依然选择可以并行化的且具有顺序无关性的 Transformer 模型的 Encoder 模块, 不仅可以学习到语义标签之间的共现性, 还可以提高模型训练效率.

在这一阶段, 纠错模型的输入是第 1 阶段里 CAI 模型对每一张关系表的预测标签序列 $\{y_1^{(1)},y_2^{(1)},...,y_k^{(1)}\}$, 其中, $y_1^{(1)},y_2^{(1)},...,y_k^{(1)}$ 不一定是完全识别正确的标签, 通过考虑 $y_1^{(1)},y_2^{(1)},...,y_k^{(1)}$ 之间的相关性, 对输入序列的每一个预测标签, 我们都将其映射到该位置上真实的语义标签. 我们将这个过程看成一个序列标注模型, 这样, 输入序列中的每一个标签经过纠错模型都可以得到一个对应的输出向量, 每一个输出位置都是关于输入的标签序列的概率最大化. 于是, 在这一阶段,我们的训练目标是:

$$\max P(S_1,S_2,...,S_k|y_1^{(1)},y_2^{(1)},...,y_k^{(1)})=\max\prod_{i=1}^{k}P(S_i|y_1^{(1)},y_2^{(1)},...,y_k^{(1)}) \tag{5}$$

其中, $\{S_1,S_2,\ldots,S_k\}$ 为对应位置观察到的类别真值. 在这一阶段的损失函数仍然是逻辑交叉熵损失函数, 如公式(4)所示.

这样, 基于标签共现的纠错模型框架如图 7 所示.

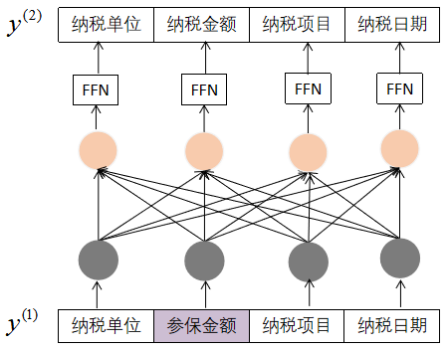


图 7 纠错阶段模型框架

3.3.2 纠错模型的训练数据构建

本节将介绍纠错阶段的训练数据构建方法.

我们将预测阶段训练得到的 CAI 模型分别在训练集、验证集、测试集上进行测试, 得到关于每一列的预测标签. 由于在预测阶段训练样本数是所有的列总数, 进入模型的是表中的各个列值, 输出的也是每一列对应的概率中 top1 对应的类别标签, 而在纠错阶段, 模型的输入是对每一个关系表预测的标签看作一条无序的序列输入, 所以纠错阶段的训练样本数等于预测阶段训练集中的关系表数量. 也就是, 如果在预测阶段的训练样本有 Q 个关系表、一共 P 列, 若预测阶段只保留对每一列的 top1 预测结果, 那么纠错阶段的训练样本数等于预测阶段关系表的数量. 于是, 纠错阶段的训练样本数仅仅只有 Q 条. 因此, 为了进一步扩充这一步的训练样本, 在预测阶段, 通过 CAI 模型对每一列都返回 top5 的预测结果. 这样, 对于一个 M 列的关系表, 每次都固定其他 M–1 列, 对剩下的一列, 从这 top5 的预测结果中随机选择一个值作为该位置的标签. 通过这种方式, 实现对训练样本数的增加. 于是, 在这纠错阶段, 仍然可以得到 P 条的训练样本.

对每一条样本, 我们将每一个标签映射到对应的 id, 向量化之后, 利用 Transformer 模型 Encoder 部分的 Self-Attention 机制实现标签之间的共现, 每一个输入标签都可以得到对应的输出向量, 进一步分类映射到真实的类别标签.

特别地, 我们的两阶段模型是分开进行训练的; 同时, 我们的工程支持用户对模型最终预测结果进行进一步反馈以更新模型. 于是, 在基于用户的反馈对模型进行更新时, 可以针对更轻量级的第 2 阶段进行单独的微调. 在未来的工作中, 我们还将进一步研究基于用户反馈的结果对模型进行更新的具体方式.

4 实验结果及分析

4.1 数据集

本文在 3 个数据集上进行了对比实验.

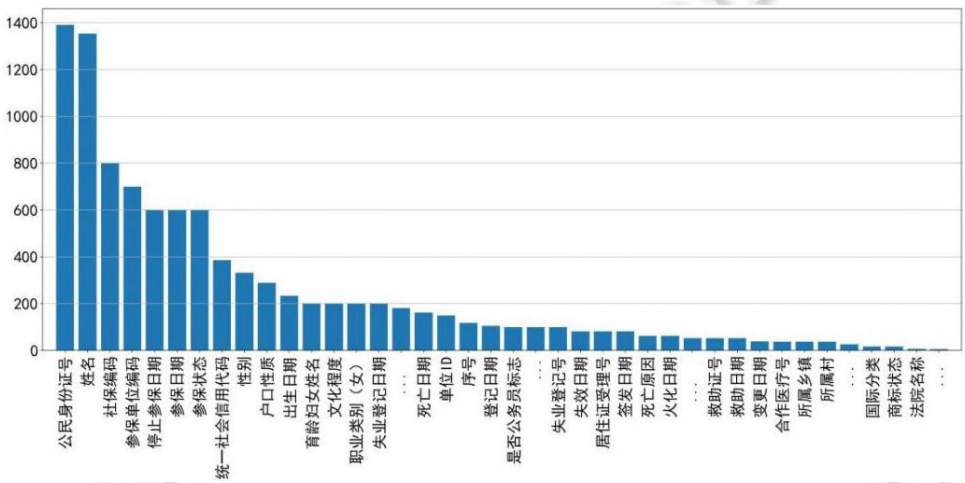
- 第 1 个数据集是政务基准数据集, 我们参考了国家发布的自然人、法人的数据元标准规范, 抽取出对应的元数据名称, 通过观察元数据之间的依赖关系, 利用数据生成工具, 自定义地构建了以自然人、法人为中心的政务基准数据集. 对应地, 这些数据集的语义类别就是标准数据元. 在政务基准数据集中不同的关系主题数有 31 个, 属性类别一共 209 个.
- 为了验证模型的有效性, 我们额外收集了 SIGMOD 2018 年发表的用于实体匹配(entity matching, EM)实验研究的数据集[29], 包括书目、音乐、电商等领域的表格数据, 以及同样可以用于实体匹配实验的 Magellan[30]、Corleone[31]、Falcon[32]数据集. 由于这些数据集之间存在领域交集, 因此, 我们对这些数据集合并在一起又重新做了划分, 得到两个新的英文数据集, 其中, EM1 包括酒店、音乐、电商、教材等领域的数据, 我们定义了 66 个不同的语义类别; EM2 包括书籍、音乐、电影评论等相关数据

集, 定义了 74 个不同语义类别.

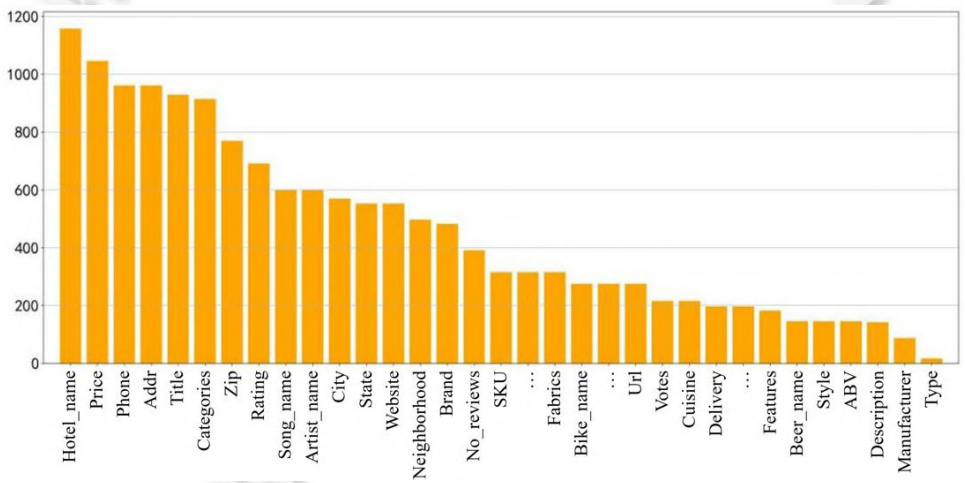
我们对每个数据集划分为训练集、验证集、测试集, 划分后的各个数据集统计情况见表 1. 图 8 展示了部分类别在训练集中的样本统计.

表 1 实验数据集

数据集	关系数	平均每表列数	语义标签数	训练集样本数	验证集样本数	测试集样本数
政务基准数据	4 733	11	209	22 096	17 169	12 842
EM1	7 513	7	66	24 211	12 669	12 669
EM2	4 770	10	74	27 212	9 141	9 141

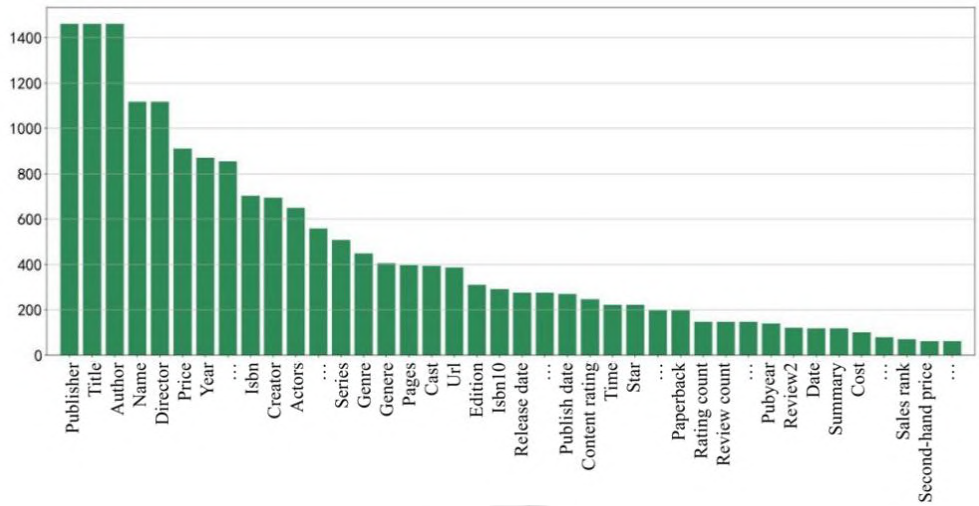


(a) 政务基准数据部分类别样本分布



(b) EM1 部分类别样本分布

图 8 3 个数据集的类别样本统计



(c) EM2 部分类别样本分布

图 8 3 个数据集的类别样本统计(续)

4.2 评价指标

我们将列的语义识别看作是对每一列的多分类问题, 因此, 选择在多分类问题的宏平均(macro-averaging)、加权平均(weighted-averaging)以及准确率(accuracy)作为评价指标. 宏平均是对所有语义类别分别求出各自的预测准确率, 然后进行平均, 不考虑类别之间的样本分布差异. 因此, 宏平均更能体现模型在小样本上的预测效果. 加权平均则是对宏平均的进一步改进, 考虑了每一个类别的样本数量在总样本数量中的占比.

(1) 宏平均

$$P_{macro}=\frac{1}{k}\sum_{i=1}^{k}\frac{tp_i}{tp_i+fp_i} \tag{6}$$

(2) 加权平均

$$P_{weighted}=\sum_{i=1}^{k}\frac{tp_i+fn_i}{total}p_i=\sum_{i=1}^{k}\frac{tp_i+fn_i}{total}\cdot\frac{tp_i}{tp_i+fn_i} \tag{7}$$

其中, k 表示类别数.

(3) 准确率, 在数值上也等于微平均

$$accuracy=\frac{sum(pred==gold)}{len(gold)}=\frac{tp+tn}{tp+tn+fp+fn} \tag{8}$$

4.3 实验环境和参数设置

在实验环境设置上, 我们使用的操作系统为 Ubuntu 20.04.3 LTS, CPU 型号为 Intel(R) Xeon(R) Gold 6138 CPU@2.00 GHz, 以及 1 个 TITAN RTX(24 GB)型号的 GPU 和 11.4 版本的 CUDA. 同时, 利用 PyTorch 框架进行了模型搭建, 其中, PyTorch 版本为 PyTorch 1.9.0+cu102.

在预测阶段的共现属性交互模型训练过程中, 小表的固定最大行数设置为 100, *batch_size* 设置为 16, BERT 的最大序列长度设置为 128, 隐藏层维度为 768, 训练轮次 *epoch* 为 10, 优化器选择 AdamW, 初始学习率设置为 2e–5, 并选择设置 *warmup* 为 0.1 来进行预热以及动态调整学习率. 对于 Sherlock 模型的学习率, 选择公开代码设置的 1e–4. 在 CAI 模型的 Column-Encoder 部分, 为了更好地学习到共现属性列之间的交互,在这一模块设置 Self-Attention 层数为 3 层.

此外，对于纠错阶段的模型训练，我们设置 Transformer 的 Encoder 模块层数为 6，多头注意力的“头”数为 8，隐藏层维度为 512，设置固定学习率 1e−4，*batch_size* 为 32，这一阶段的最大序列长度设置为 32，训练轮次 *epoch* 依然为 10，优化器选择 Adam.

4.4 实验结果与分析

为了验证模型的结果，我们与几个基准方法进行实验对比.

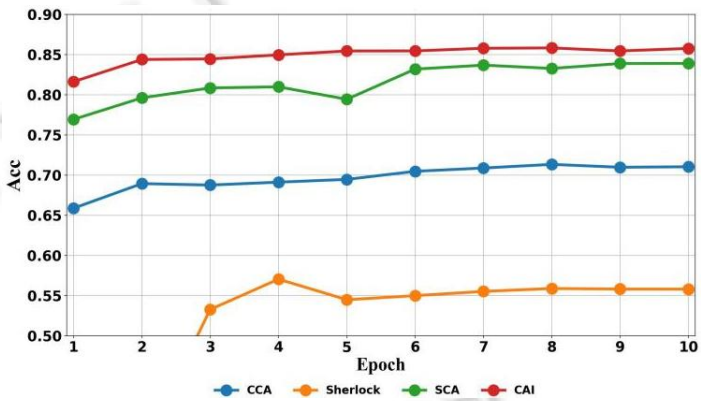
(1) CCA model[4]: CCA 模型仅仅考虑了待识别列的单列内容信息，简单将待识别列线性化，通过在预训练模型 BERT 上微调直接进行分类.

(2) Sherlock model[19]: Sherlock 模型同样只考虑了待识别列的单列信息，但 Sherlock 对待识别列提取了大量统计特征，并和列的词向量以及“段落表示”拼接联合训练，利用 MLP 进行分类. 由于 Sherlock 模型除了统计特征，还使用了 Glove 预训练好的英文词向量，于是，在中文的政务基准数据上，我们利用 Li 等人[33]预训练好的中文词向量，统计特征的提取仍使用 Sherlock 模型的提取方式.

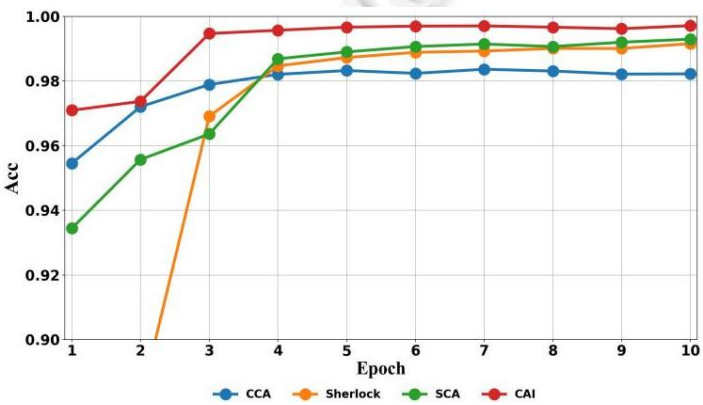
(3) SCA model[4]: SCA 模型是在 CCA 模型基础上进一步考虑关系表的上下文信息，同样对待识别列线性化，并将同一关系主题下的其他属性列按行拼接看作一个句子，与线性化的待识别列用特殊标识符[SEP]隔开，作为一整段文本，在预训练模型 BERT 上微调进行分类.

4.4.1 模型训练准确率对比分析

图 9 是各个模型在每个数据集上分别迭代训练 10 个 *epoch* 以内的准确率变化情况.

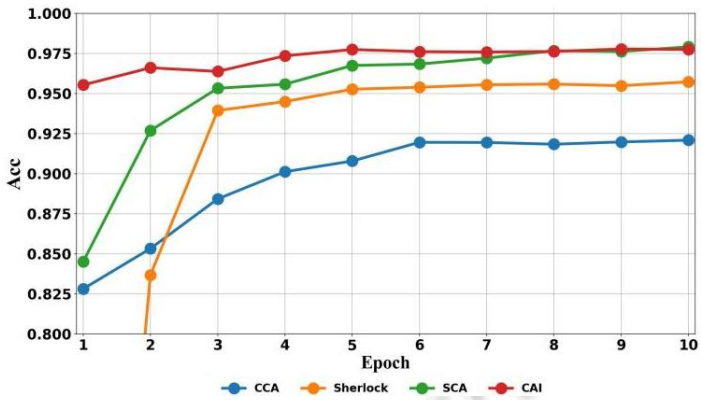


(a) 政务基准数据上的准确率变化



(b) EM1 数据集上的准确率变化

图 9 各模型准确率随迭代次数变化示意图



(c) EM2 数据集上的准确率变化

图 9 各模型准确率随迭代次数变化示意图(续)

可以发现, 在 3 个数据集上, 除了 Sherlock 模型在 EM2 上一直未收敛以外, 其他各模型都在 10 个 *epoch* 以内达到收敛, 但 Sherlock 模型的准确率已经达到 95%+的水平. 其中, 在政务基准数据集上, Sherlock 表现糟糕, 准确率仅仅达到 55%+; 同样地, CCA 模型虽然也已经收敛, 但其准确率也仅仅稳定在 70%+; 而同样考虑到关系表上下文信息的 SCA 模型和 CAI 模型在准确率上显著高于 Sherlock 和 CCA 模型, 不过, 比较而言, CAI 模型的准确率仍然高于 SCA 模型, 其数值也随着迭代次数逐渐稳定高于 85%. 同样地, 在英文数据集 EM1 和 EM2 上, CAI 模型的准确率也都表现最好, SCA 模型与之接近. 可以看出关系表上下文信息的加入, 对于列语义识别的重要性和稳健性, 而其中, 以自注意力机制实现关系表共现属性交互的方式效果更突出.

4.4.2 共现属性交互和纠错机制的有效性验证

在 3 个数据集上, 关于各个模型, 我们在宏平均和加权平均上进行了更详细的比较. 除了几个基准模型和 CAI 模型以外, 这里也比较了在共现属性交互的基础上引入纠错机制的结果, 验证纠错阶段对列语义识别工作的有效性, 其中, Correction-top1 表示第 3.3.2 节中训练样本只来自预测阶段的 top1 结果, Correction-top5 表示训练样本来自第 1 阶段的预测结果固定一列从 top5 中随机选择的结果(见表 2).

表 2 实验结果(%)

	政务基准数据		EM1		EM2	
	macro	weighted	macro	weighted	macro	weighted
CCA[2]	57.71	71.56	98.67	98.33	89.22	91.16
Sherlock[17]	45.09	59.47	99.63	99.48	94.87	95.81
SCA[23]	78.25	86.60	99.64	99.44	95.80	97.63
CAI	82.02	88.57	99.71	99.65	96.43	97.95
CAI+Correction-top1	96.95	99.67	99.99	99.99	99.59	98.09
CAI+Correction-top5	**98.28**	**99.96**	**100.0**	**100.0**	**99.97**	**99.96**

从实验结果可以看出, 考虑共现属性的 SCA 和 CAI 模型都表现很好. 比较而言, 在同样考虑关系数据上下文的基础上, CAI 模型的列向量之间通过自注意力机制交互的方式比 SCA 模型效果更好, 在政务基准数据上尤其显著.

对于政务基准数据集, CAI 的宏平均和加权平均分别达到了 82.02%, 88.57%, 比 SCA 提高了 3.77%, 1.97%, 比 CCA 模型分别提高了 24.31, 17.01%. 在政务基础数据集中, 各模型的宏平均普遍偏低, 具体分析发现: 该数据集存在部分类别的样本数格外不均衡的现象, 比如“公告类型”“变更项目”等属性的样本就非常稀疏, 模型对这些属性的特征学习不够, 导致在政务基准数据集上各个模型的宏平均较差. 另外, 在政务基准数据集中存在大量实例特征相似但是语义不同的属性, 比如“配偶身份证号”“公民身份证号”“育龄妇女身份证号”等, 还有大量日期类型的属性列, 比如“参保日期”“停止参保日期”“出生日期”等等, 对于这些属性,仅仅依靠单列的属性值信息无法进行区分, 此时, 关系表的上下文信息就显得格外重要. 所以在政务基准数据集中,

CCA 模型结果不如在两个英文数据集上表现良好. 除此之外, 在政务基准数据集上, 可能由于选择的预训练中文词向量质量不够好, Sherlock 模型在政务基准数据集上表现糟糕, 虽然已经收敛, 但宏平均和加权平均仅仅分别达到 45.09%, 59.47%.

对比政务基准数据, 各个模型在两个英文数据集中表现都不错, 但 CAI 模型仍然表现最优. 其中,对于英文数据集 EM1, CAI 的宏平均和加权平均分别达到了 99.71%, 99.65%, 比 SCA 模型提高了 0.07%, 0.21%, 相差不大; CCA 模型在数据集 2 上也表现不错, 仅仅比考虑关系上下文的 SCA 模型和 CAI 模型低了 1 个百分点左右. 这是因为在数据集 2 中, 各个列的上下文依赖关系不大, 且关系列之间特征差异大, 仅仅依靠单独列的信息就可以很好地区分. 在英文数据集 EM2 上, CAI 模型表现仍然最好, 宏平均和加权平均分别达到 96.43%, 97.95%, 比 SCA 模型高 0.63%, 0.32%, 比 Sherlock 模型提高 1.56%, 2.14%.

尽管 CAI 模型在对比其他几个基准实验时表现最好, 但是通过结果也可以发现, 不管是训练样本选择 top1 的结果还是 top5 的结果, 在 CAI 模型基础上加入标签共现的纠错阶段, 都可以更加有效地提高对关系列的语义识别效果. 在政务基准数据集上, CAI+Correction-top1 可以在宏平均和加权平均分别达到 96.95%, 99.67%, 而 CAI+Correction-top5 在宏平均和加权平均可以分别达到 98.28%, 99.96%, 在 CAI 模型基础上, 进一步分别提升 16.26%, 11.39%, 比 SCA 模型可以分别提高 20.03%, 13.36%. 对于两个英文数据集, 在 CAI 模型基础上, 也还可以进一步提升效果, 尤其在英文数据集 EM1 上, 宏平均和加权平均甚至可以达到 100%; 对于英文数据集 EM2, CAI+Correction-top5 在宏平均和加权平均达到了 99.97%, 99.96%, 比 CAI 模型分别提高 3.54%, 2.01%,比 SCA 模型分别提高 4.17%, 2.33%.

总结以上对比, 本文提出的两阶段 CAI-Correction 模型对于关系列语义识别工作具有显著有效性.

5 总 结

逻辑语义汇通是当前政务数据治理的重要环节, 本文通过对孤岛系统关系列的语义识别工作, 实现孤岛元数据标准化治理. 针对现有列语义识别工作的不足, 我们同时结合关系数据层面和类别标签层面的上下文关系, 提出了两阶段的 CAI-Correction 模型. 该模型基于预训练模型 BERT 对线性化的关系列进行编码, 并利用顺序无关的自注意力机制实现关系表中共现列之间的交互, 在保证关系表列顺序无关性的基础上, 增强列的语义表示进行预测; 进一步地, 结合标签共现的纠错机制, 实现对模型初步预测结果的优化. 本文算法的优点在于:

(1) 引入了具有关系数据列顺序无关性的共现属性交互, 模型可以学习到关系表的上下文信息, 并且具有更好的鲁棒性.

(2) 引入了基于标签共现的纠错机制, 对CAI模型预测结果进一步优化, 保证列语义识别工作的闭环性, 更加显著地提升模型识别效果.

(3) 充分利用并行化的自注意力机制, 可以提高模型训练效率.

本文在政务基准数据集和两个公开的英文关系数据集上进行了对比与分析, 充分验证了两阶段模型的有效性.

References:

[1] Du XY, Chen YG, Fan J, *et al*. Data wrangling: A key technique of data governance. Big Data Research, 2019, 5(3): 13−22 (in Chinese with English abstract).

[2] Wu XD, Dong BB, Du XZ, Yang W. Data governance technology. Ruan Jian Xue Bao/Journal of Software, 2019, 30(9): 2830−2856 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5854.htm [doi: 10.13328/j.cnki.jos.005854]

[3] Zhang D, Suhara Y, Li JF, Hulsebos M, Demiralp Ç, Tan WC. Sato: Contextual semantic type detection in tables. CoRR abs/1911.06311, 2019.

[4] Ding Y, Guo YH, Lu W, Li HX, Zhang MH, Li H, Pan AQ, Du XY. Context-aware semantic type identification for relational attributes. Journal of Computer Science and Technology, 2021. https://jcst.ict.ac.cn/EN/10.1007/s00000-021-1048-2

[5] Devlin J, Chang MW, Lee K, *et al*. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805, 2018.

[6] Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. In: Proc. of the Advances in Neural Information Processing Systems. 2017. 5998−6008.

[7] Venetis P, Halevy A, Madhavan J, Pasca M, Shen W, Wu F, Miao GX, Wu C. Recovering semantics of tables on the Web. Proc. of the VLDB Endowment, 2011, 4(9): 528−538.

[8] Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. DBpedia: A nucleus for a Web of open data. In: Proc. of the ISWC. 2007. 722−735.

[9] Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: A collaboratively created graph database for structuring human knowledge. In: Proc. of the SIGMOD. 2008. 1247−1250.

[10] Jiménez-Ruiz E, Hassanzadeh O, Efthymiou V, *et al*. Semtab 2019: Resources to benchmark tabular data to knowledge graph matching systems. In: Proc. of the European Semantic Web Conf. Cham: Springer, 2020. 514−530.

[11] Ritze D, Lehmberg O, Bizer C. Matching html tables to DBpedia. In: Proc. of the 5th Int'l Conf. on Web Intelligence, Mining and Semantics. 2015. 1−6.

[12] Efthymiou V, Hassanzadeh O, Rodriguez-Muro M, *et al*. Matching Web tables with knowledge base entities: From entity lookups to entity embeddings. In: Proc. of the Int'l Semantic Web Conf. Springer, 2017. 260−277.

[13] Zhang Z. Towards efficient and effective semantic table interpretation. In: Proc. of the Int'l Semantic Web Conf. Springer, 2014. 487−502.

[14] Azzi R, Diallo G, Jiménez-Ruiz E, *et al*. AMALGAM: Making tabular dataset explicit with knowledge graph. In: Proc. of the SemTab@ISWC. 2020. 9−16.

[15] Nguyen P, Kertkeidkachorn N, Ichise R, *et al*. MTab: Matching tabular data to knowledge graph using probability models. arXiv:1910.00246, 2019.

[16] Ramnandan SK, Mittal A, Knoblock CA, Szekely P. Assigning semantic labels to data sources. In: Proc. of the ESWC. Springer, 2015. 403–417.

[17] Pham M, Alse S, Knoblock CA, Szekely P. Semantic labeling: A domain-independent approach. In: Proc. of the ISWC. Springer, 2016. 446–462.

[18] Chen Z, Jia H, Heflin J, *et al*. Generating schema labels through dataset content analysis. In: Companion Proc. of the Web Conf. 2018. 1515−1522.

[19] Hulsebos M, Hu KZ, Bakker MA, Zgraggen E, Satyanarayan A, Kraska T, Demiralp A, Hidalgo C. Sherlock: A deep learning approach to semantic data type detection. In: Proc. of the KDD. 2019. 1500−1508.

[20] Lafferty JD, McCallum A, Pereira FCN. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. of the ICML. 2001. 282−289.

[21] Ge C, Gao Y, Miao X, *et al*. A hybrid data cleaning framework using Markov logic networks. IEEE Trans. on Knowledge and Data Engineering, 2022, 34(5): 2048−2062.

[22] Ge C, Liu X, Chen L, *et al*. Largeea: Aligning entities for large-scale knowledge graphs. arXiv:2108.05211, 2021.

[23] Tang X, Zhang J, Chen B, *et al*. BERT-INT: A BERT-based interaction model for knowledge graph alignment. In: Proc. of the Int'l Joint Conf. on Artificial Intelligence. 2020. 3174−3180.

[24] Chen J, Jiménez-Ruiz E, Horrocks I, *et al*. Colnet: Embedding the semantics of Web tables for column type prediction. Proc. of the AAAI Conf. on Artificial Intelligence, 2019, 33(1): 29−36.

[25] Deng X, Sun H, Lees A, *et al*. TURL: Table understanding through representation learning. ACM SIGMOD Record, 2022, 51(1): 33−40.

[26] Hu D. An introductory survey on attention mechanisms in NLP problems. In: Proc. of the SAI Intelligent Systems Conf. Cham: Springer, 2019. 432−448.

[27] Harris ZS. Distributional structure. Word, 1954, 10(2−3): 146−162.

[28] Du L, Gao F, Chen X, *et al*. TabularNet: A neural network architecture for understanding semantic structures of tabular data. In: Proc. of the 27th ACM SIGKDD Conf. on Knowledge Discovery & Data Mining. 2021. 322−331.

[29] Mudgal S, Li H, Rekatsinas T, *et al*. Deep learning for entity matching: A design space exploration. In: Proc. of the Int'l Conf. on Management of Data. 2018. 19–34.

[30] Konda PV. Magellan: Toward Building Entity Matching Management Systems. Proc. of the VLDB Endowment, 2016, 9(12): 1197–1208.

[31] Gokhale C, Das S, Doan AH, *et al*. Corleone: Hands-off crowdsourcing for entity matching. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. 2014. 601–612.

[32] Das S, Paul SGC, Doan AH, *et al*. Falcon: Scaling up hands-off crowdsourced entity matching to build cloud services. In: Proc. of the ACM Int'l Conf. on Management of Data. 2017. 1431–1446.

[33] Li S, Zhao Z, Hu RF, Li WS, Liu T, Du XY. Analogical reasoning on Chinese morphological and semantic relations. In: Proc. of the ACL 2018. 2018. 138–143.

附中文参考文献:

[1] 杜小勇, 陈跃国, 范举, 等. 数据整理——大数据治理的关键技术. 大数据, 2019, 5(3): 13–22.

[2] 吴信东, 董丙冰, 堵新政, 杨威. 数据治理技术. 软件学报, 2019, 30(9): 2830–2856. http://www.jos.org.cn/1000-9825/5854.htm [doi: 10.13328/j.cnki.jos.005854]

高珊(1997－), 女, 硕士, 主要研究领域为自然语言处理, 数据标准化.

王兰(1993－), 女, 硕士, 主要研究领域为自然语言处理.

袁宛竹(1998－), 女, 硕士生, 主要研究领域为自然语言处理.

张静(1973－), 女, 博士, 副教授, 博士生导师, CCF 专业会员, 主要研究领域为数据挖掘, 自然语言处理.

卢卫(1981－), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为数据库基础理论, 大数据系统研制, 时空背景下的查询处理, 云数据库系统和应用.

杜小勇(1963－), 男, 博士, 教授, 博士生导师, CCF 会士, 主要研究领域为智能信息检索, 高性能数据库, 非结构化数据管理.