

申请上海交通大学硕士学位论文

SQL 查询语句的自动生成技术研究

论文作者 熊云翔

学 号 116037910048

导 师 沈备军

专 业 软件工程

答辩日期 2018 年 12 月 12 日

Submitted in total fulfillment of the requirements for the degree of Master
in Engineering

Research on Automatic Generation Technology of SQL Query Statement

XIONG YUNXIANG

Advisor

Prof. BEIJUN SHEN

SOFTWARE ENGINEERING

SHANGHAI JIAO TONG UNIVERSITY

SHANGHAI, P.R.CHINA

Dec. 12th, 2018

上海交通大学 学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：_____

日 期：_____年 _____月 _____日

上海交通大学 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于

保 密 ☐，在 _____ 年解密后适用本授权书。

不保密 ☐。

(请在以上方框内打 ☒)

学位论文作者签名： _____

指导教师签名： _____

日 期： _____ 年 ____ 月 ____ 日

日 期： _____ 年 ____ 月 ____ 日

SQL 查询语句的自动生成技术研究

摘 要

待写 关键词：上海交大, 饮水思源, 爱国荣校

RESEARCH ON AUTOMATIC GENERATION TECHNOLOGY OF SQL QUERY STATEMENT

ABSTRACT

write **KEY WORDS:** SJTU, master thesis, XeTeX/LaTeX template

目 录

插图索引	VII
表格索引	IX
算法索引	XI
主要符号对照表	XIII
第一章 绪论	1
1.1 研究背景	1
1.2 研究目标和研究内容	2
1.3 论文结构	3
第二章 基于映射的 NLI2SQL 生成	5
2.1 研究问题	5
2.2 相关技术	5
2.3 解决方案	5
2.4 实验与分析	8
2.5 本章小结	10
第三章 基于深度强化学习的 NL2SQL 生成	11
3.1 研究问题	11
3.2 相关技术	11
3.3 解决方案	11
3.4 实验与分析	12
3.5 本章小结	12
第四章 基于多任务学习的 NL2SQL 生成	13
4.1 研究问题	13
4.2 相关技术	13
4.3 解决方案	13

4.4 实验与分析	13
4.5 本章小结	13
第五章 总结与展望	17
5.1 本文工作小结	17
5.2 展望	17
参考文献	19
致 谢	21
攻读学位期间发表的学术论文	23
攻读学位期间参与的项目	25
简 历	27

插图索引

3-1 这里将出现在插图索引中	12
---------------------------	----

表格索引

算法索引

主要符号对照表

ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数

μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率

第一章 绪论

1.1 研究背景

随着社会的不断发展，以 IT 和互联网技术为标志的信息产业不断地改变着人类的工作和生活方式。在此背景之下，数据库技术应运而生。它是一种建立在计算机存储设备上的仓库，可以将大量数据按照数据结构来组织、存储和管理。关系数据库中存储了大量的数据和信息。医疗、教育、金融等各个行业都在使用关系型数据库作为数据存储以及应用程序的基础。在软件开发过程中，软件的开发和技术人员会频繁地进行 SQL 语句的创建与查询以及相关数据库的操作。在软件运行时，业务人员也会使用 SQL 语句进行报表与在线分析（OLAP）的定制。

目前，各种计算机系统，尤其是商业领域，应用关系型数据库系统和 SQL 语言来进行数据管理，仍然是最主流和最成熟的方案。需要注意的是，SQL 作为一种数据库操作语言，本质上仍然是一种编程语言，需要操作人员具有一定的专业知识，经过数据库和 SQL 相关知识的培训，才能比较熟练的进行数据库的管理。此外，除了要具备 SQL 和数据库技术的相关知识，具体到真实数据库的操作，数据管理人员还需要对于所使用的关系型数据库的关系模式有所了解，才能将各种操作需求转化为 SQL 语句，来对数据库系统进行管理。然而，随着数据库系统的应用场景越发广泛和复杂，以及数据库数据处理量的不断提升，数据管理人员对数据库的操作逻辑也越来越复杂，数据库查询需求所涉及的数据量也越来越大，相关的关系模式也越来越复杂和多样化。当数据库管理需求达到这样的复杂度，非专业的数据管理人员就越来越无法满足需求。

1.1.0.1 自然语言接口自动生成 SQL 查询语句

自然语言接口（NLI, NaturalLanguageInterfaces），是自然语言处理和人机交互的交叉领域，旨在为人类提供通过自然语言与计算机交互的手段。同时，自然语言接口也是人机交互领域研究的终极目标，从各种对话机器人，到今天各种智能穿戴设备装载的语音助手，人机交互领域和自然语言理解领域的专家们一直在朝建立真正智能的自然语言接口这个目标不断前进。

自然语言接口生成 SQL 也是人们关注的一个领域^[1]，自上个世纪提出以来，人们不断研究从自然语言生成 SQL 语句的可能性，并且的确在研究过程中取得了一些令人振奋的成果。通过自然语言接口生成 SQL 的数据库管理系统，原型已经出现在六十年代和七十年代初期，那时候最著名的自然语言接口数据库是 Lunar，正如其名字，包含月

球岩石和化学数据库的自然语言界面。这个原型的实现，是基于特定数据库的，因此无法很容易地修改为和不同的数据库一起使用。之后出现了其他的自然语言接口数据库，用户可以通过对话系统来定制查询，并且这些系统可以配置不同的接口，供不同的底层数据库调用。这时候的自然语言数据库系统使用语义语法，是一种句法和语义处理的综合技术。之后，还有关注于将自然语言输入转化为逻辑语言的技术，以此技术作为自然语言接口数据库的核心技术。

一直以来，自然语言接口是人机交互领域的终极追求，也是人机交互、机器学习领域专家孜孜不倦钻研学习的热门问题。对于本文所指出的业务层面与技术层面之间的矛盾，如果能对最终用户提供一个自然语言接口，使得我们的系统能直接从最终用户的自然语言中理解到用户的查询意图，并结合数据库，直接生成符合查询意图的 SQL 语句返回给用户，那么这一矛盾就可以较好的得到解决。

1.1.0.2 自然语言自动生成 SQL 查询语句

XXXXXXXXXX

XXXXXXXXXX

XXXXXXX

因此，本文设计和实现一个 SQL 查询语句自动生成工具，为数据库使用者提供简单、便捷的接口，将数据库信息映射到业务需求。用户无需了解 SQL 语句的使用方式，只需关注数据操作需求对应的业务需求，从而弥合业务人员与数据操作之间的矛盾。同时对从自然语言自动生成 SQL 查询语句技术（以下简称 NL2SQL）进行了研究，提出了针对英文自然语言生成 SQL 查询语句的解决方案以及针对中文自然语言生成 SQL 查询语句的解决方案，使得用户可以通过自然语言的表述方式生成 SQL 查询语句并从关系型数据库中找到所需信息，从而缩短业务与技术之间的鸿沟，提高报表与 OLAP 分析的开发效率。

1.2 研究目标和研究内容

本文的研究目标是研究 SQL 查询语句的自动化生成技术，采用解析树映射、语义解析、编码-解码器、注意力机制、深度强化学习和多任务学习等技术，提出基于自然语言接口、英文自然语言和中文自然语言自动生成 SQL 查询语句的技术方案。

本文的研究思路是，先对自然语言进行初步解析和理解，并在其中插入人机交互机制，让用户来引导生成的过程，指导自然语言理解，提出自然语言接口自动生成 SQL 查询语句的方法。然后结合编码-解码器和深度强化学习等技术，对更具难度的纯英文自然语言自动生成 SQL 查询语句技术进行研究。最后使用多任务学习技术将中文-英文翻

译任务和英文自然语言生成 SQL 查询语句技术有机结合，从而提出难度更高的中文自然语言生成 SQL 查询语句方法。

具体研究内容包括：

1. SQL 查询语句自动生成现状。xxxxxxx
2. 基于映射的 NLI2SQL 生成。xxxxxx
3. 基于深度强化学习的 NL2SQL 生成。xxxxx
4. 基于多任务学习的 NL2SQL 生成。xxxx
5. 实验。xxxx

1.3 论文结构

第一章 绪论

从自然语言接口和自然语言自动生成 SQL 查询语句两个方面介绍了课题的研究背景、研究目标和研究内容，对全文做出总览。最后说明了论文的组织结构

第二章 基于映射的 NLI2SQL 生成

第三章 基于深度强化学习的 NL2SQL 生成

第四章 基于多任务学习的 NL2SQL 生成

第五章 总结与展望

第二章 基于映射的 NLI2SQL 生成

2.1 研究问题

一直以来,自然语言接口是人机交互领域的终极追求,也是人机交互、机器学习领域专家孜孜不倦钻研学习的热门问题。对于本文所指出的业务层面与技术层面之间的矛盾,如果能对最终用户提供一个自然语言接口,使得我们的系统能直接从最终用户的自然语言中理解到用户的查询意图,并结合数据库,直接生成符合查询意图的 SQL 语句返回给用户,那么这一矛盾就可以非常自然和优美地得到解决。因此,本章将对从自然语言生成 SQL 语句的技术和模型进行探究,尝试寻找一种解决方案,能提供自然语言接口给非技术用户,让用户通过以自然语言表达的查询意图,得到目标 SQL 语句。

自然语言理解存在许多难题,如歧义、语序,或者存在复杂的依赖结构等等,要完全基于自然语言理解来进行 SQL 生成是很困难的,效果可能会不尽如人意。所以,受到人机交互思想的启发,本文将使用自然语言理解与人机交互相结合的方式,来进行从自然语言到目标 SQL 语句的转化。对于自然语言意图,先对自然语言进行初步解析和理解,并在其中插入人机交互机制,让用户来引导生成的过程,指导自然语言理解,纠正机器理解过程中出现的错误、歧义、含糊不清的问题,从而提升整体的准确性。

2.2 相关技术

2.3 解决方案

图 3-1 是本文从自然语言生成 SQL 语句模型的总体方案,它包含依赖解析树生成、解析树节点映射、解析树优化重构、查询树翻译、交互式对话器、用户接口六个模块: 1) 用户接口: 用户与系统进行交互的接口,包括输入自然语言、返回 SQL 语句、解析过程中的交互等等。2) 交互式对话器: 管理解析过程与用户的交互,在适当的时候与用户进行交互,让用户对解析过程进行指导。3) 依赖解析树生成: 将用户输入的,以自然语言表示的查询意图,应用自然语言理解技术,转化为依赖解析树,即将词语进行词性标注,以及识别出词语之间的关系,并将其组织成一个树状结构。详见 3.2 节。4) 解析树节点映射: 根据解析树节点对应的词语和数据库元数据、数据、SQL 语法等信息,将解析树节点映射为 SQL 语法组件。在这个过程中,如果节点的映射有多个候选答案,交互式对话器会发起与用户的交互,将多个候选映射展示给用户,让用户来进行选择。详见 3.3 节。5) 解析树优化重构: 节点映射完毕后,解析树节点经过系统匹配和

用户指导后,得到了比较准确的结果,但解析树的结构仍是最初由依赖解析树生成得到的结构,这一原始结构的准确度并不高,受限于自然语言的复杂性和省略性,可能会有错误关系、缺失关系、缺失节点等等。为了使解析树能得到比较准确的结构,将设计算法和规则,修正错误关系,补全缺失节点和缺失关系,得到较为准确的解析树,称为查询树。详见 3.4 节。6) 查询树翻译:解析树的结构已经符合 SQL 语法,节点映射结果也对应于真实的 SQL 语法、数据库 schema 或数据,可以很自然地将树状结构的翻译为 SQL 语句,最后将 SQL 语句通过用户接口返回给用户。详见 3.5 节。

接下来,将详细阐述依赖解析树模块、解析树节点映射模块、解析树优化重构模块、查询树翻译器的设计思路和实现细节。

依赖解析树生成

本模块将用户输入的自然语言查询意图解析为解析树,包含各词语的词性标注、关系提取等等信息。在具体实现过程中,这一模块基于 StanfordCoreNLP[8] 实现。StanfordCoreNLP 是斯坦福大学推出的自然语言处理工具集,支持多种语言,还提供了 C++、Python、Java 等多种程序语言的编程接口,提供依赖解析、命名实体识别、词性标注、情感分析、机器翻译等多种功能。本模块调用该工具,将自然语言解析为依赖解析树。

解析树节点映射

本模块将依赖解析树的节点对应的词语,映射到 SQL 组件上。解析树节点的类型定义如表 3-1 所示。

表 3-1 解析树节点的类型节点类型对应的 SQL 组件选择节点 (SN) SELECT 操作符节点 (ON) 一个操作符,如等于 (“=”)、小于 (“<”) 聚合函数节点 (FN) 一个聚合函数,如 AVG、MAX 名字节点 (NN) 业务数据库中的一个数据表的名字,或数据表的一个字段的名字值节点 (VN) 业务数据库中某字段的一个值度量节点 (QN) ALL, ANY, EACH 逻辑节点 (LN) AND, OR, NOT

其中名字节点和值节点是与当前应用的业务数据库有关,其余五种节点都与业务数据库无关,仅与 SQL 语法规则相关。所以,本系统建立了一个五种与业务数据库无关的节点类型与自然语言单词的词典映射。映射过程的实现如下:对每一个解析树节点对应的单词 n ,分别计算其与业务数据库元数据、存储数据、词典映射中词语 v 的相似度 $\text{Sim}(n,v)$, $\text{Sim}(n,v)$ 的定义如下:

其中为 n 和 v 的 WUP 相似度 [12], (n,v) 为 n 和 v 的 q -gram 的 Jaccard 相似度的平方根 [13]。经公式计算,可以得到节点单词 n 与所有 SQL 组件的相似度;对相似度进行排序,可以得到前五相似的 SQL 组件,如果前五相似的 SQL 组件的相似度 $\text{Sim}(n,v)$ 的值差别较大,则直接以相似度最高的 SQL 组件作为当前单词 n 的映射,并赋予该组件对应的节点类型;若前五相似的 SQL 组件的相似度差别较小,则视作歧义,将候选

的 SQL 组件返回给用户，让用户来进行选择，最后用户选择的结果会作为当前节点的映射。

解析树优化重构

节点映射完成后，需要对解析树的结构进行重构，保证解析树的结构能有较高的准确性。由于解析树可能会存在关系解析错误和节点关系缺失，所以这一模块对于解析树的优化重构会分为两个步骤进行，分别为结构调整和隐藏节点插入。

结构调整

在进行结构调整之前，首先定义什么样的树结构是好的、合法的。我们从两个角度来考虑这个问题：第一点是树结构与经依赖解析器解析后的原始结构的差别有多大；第二点是树结构是否符合 SQL 的语法，这一点的评估可以根据表 3-2 的定义来确定 [9]。表 3-2 根据 SQL 语句的语法，结合了树状结构，定义了能合理的转化为 SQL 语句的语法树应该满足什么样的规则，这样的语法树我们称之为查询树。在表 3-2 中，“+”代表父子节点的关系，“*”代表兄弟节点的关系，上标“*”代表可重复出现的兄弟节点，“|”代表“或者关系”，表示当前节点可能存在的情况。

表 3-3 展示了结构调整的算法，算法的基本思想是，建立一个优先级队列，对于当前的解析树，调用 `adjust()` 函数（第 8 行），通过一次移动子树操作，生成在这一次操作后所有可能的结构；然后记录当前树的哈希值（第 12 行），防止之后出现重复的树结构；若当前树结构没有出现过（第 10 行），且 `edit` 值小于一个阈值，则对此树进行下一步操作；由于移动了一次子树，返回的树的 `edit` 属性加一（第 11 行），这一属性将用来评估生成的树结构与原始结构的差异；调用 `evaluate()` 函数，记录当前结构有多少节点不满足表 3-2 设定的规则，不满足规则的节点数将用来评估树结构在语法上的合法性；综合这两方面评估标准，为树打分，如果分数比之前的树结构要高，则加入优先级队列；若该树完全符合语法规则，则视为一颗查询树，加入 `result` 集合；之后对优先级队列内的树结构重复以上操作，直到优先级队列为空；最后根据评估分数对 `result` 集合排序，将结果返回。

结构调整之后的解析树结果集，将会通过交互式对话器，与用户进行交互，因为结果集中的解析树虽然都符合 SQL 语法规则，但仍有可能存在与用户意图不同的情况，如 `SELECT` 子句中的名字节点和 `WHERE` 子句中的名字节点，位置可能会互换，虽然仍然符合 SQL 语法规则，但与原始查询意图已经有比较大的差别了。所以，在这里需要再一次应用人机交互机制，让用户来选择结果集中与自己的查询意图比较相似的解析树。交互完成后，筛选出的结果集，会进入下一步——隐藏节点插入。

隐藏节点插入

结构调整完成后，对经过排序和用户交互筛选后的结果集合，进行隐藏节点插入。

在给出隐藏节点插入的方法之前，先给出需要用到的概念定义，即“核心节点”。核心节点指的是在节点类型为 `leftSubtree` 和 `rightSubtree` 的情况下（表 3-2），`leftSubtree`（`rightSubtree`）的所有子节点中，高度最高的名称节点被称为核心节点。

经过研究，需要进行隐藏节点插入的情况有以下几种：1）左子树（`leftSubtree`）与右子树（`rightSubtree`）的核心节点对应了不同的 SQL 组件，即认为右子树真正的核心节点在自然语言表达时被省略了 [14]；这是十分常见的现象，因为人在进行自然语言表达时，对两个值进行比较时，会很自然的省略掉后者的一部分，如“Ihavemorebooksthanyours”这句话，就将最后的“yourbooks”给省略成了“yours”；2）左右子树的约束条件应该一致，如果不一致，则认为右子树一部分约束条件被省略了，如“returnauthorswhosepaperspublishedin2018morethanJack’s”这句话，过滤条件的左子树有“in2018”这一约束，而右子树在解析之后没有这一约束，事实上右子树的这一约束被隐藏了，需要作为隐藏节点插入进去；3）某些函数会被省略，如聚合函数“COUNT”，在自然语言表达中经常会被省略，如“Ihavemorebooksthanyours”这句话，“thenumberof”就被省略了。在树结构中，如果过滤条件的操作符为“more”、“less”等词语，而左右子树的核心节点对应的是非数字类型的 SQL 组件，那么就认为“COUNT”被省略了，需要作为隐藏节点插入解析树。

进行隐藏节点插入之后，查询树的结构就比较完整了，将会输入下一模块进行翻译。

查询树翻译

这一阶段的查询树已经在节点映射、树结构、完整性方面都比较可信、合法了，翻译步骤如下：1）根据树结构，在 `SClause` 子树下的结构为 `SELECT` 子句，读取 `SClause` 下的名称节点，根据对应的 SQL 组件（如果 SQL 组件对应某数据表，该数据表会预定义一个核心字段，如用户的名字、城市的名称，该数据表的核心字段将作为结果返回），填充入 `SELECT` 子句，并记录 SQL 组件对应的数据表，以备 `FROM` 子句的生成 2）在 `ComplexCondition` 子树下的结构为 `WHERE` 子句，分别读取左子树和右子树核心节点对应的 SQL 组件，记录下 SQL 组件对应的数据表，并查看左右子树中所有的节点，根据其节点类型，将其翻译为对应的 SQL 组件，并应用于核心节点；左右子树解析完后，以操作符连接左右子树，将其填充入 `WHERE` 子句 3）根据之前记录的相关数据表，生成 SQL 语句的 `FROM` 子句 4）将三部分按照语法连接起来，作为合法的 SQL 语句返回给用户。

2.4 实验与分析

数据集

本次实验所使用的业务数据库为 MySQL 的样例数据库 `Classicmodels`，图 3-2 为

Classicmodels 数据库的数据库模式图

实验所使用的自然语言查询数据集由作者根据 Classicmodels 数据库的模式建立, 根据查询意图的复杂度, 分为简单、中等、困难三类。每种类别, 提出了 20 条自然语言查询, 共计 60 条, 用来测试模型的准确性。表 3-4 给出了三种类别的自然语言查询示例。

结果分析 1) 经过两次实验, 分别检验完整模型和无交互机制模型生成 SQL 语句的准确性。可以看出, 交互机制对于模型的准确性有很大的提升; 而加入了交互机制后, 模型可以比较准确的处理简单和中等复杂度的查询意图, 即便是复杂度较高的查询意图, 本文提出的模型仍然可以正确生成一部分困难复杂度的 SQL 语句。在实验过程中, 经常出现节点歧义需要映射, 如 “price” 一词可以映射到 “products” 表中的 “buyPrice”, 也有可能映射到 “orderdetails” 表中的 “priceEach”。那么对于测试语句 “return order details whose price is higher than 50”, 如果没有交互机制, 对于 “price” 这个节点的映射就会优先映射到 “products” 中的 “buyPrice”, 而实际上应该映射到 “orderdetails” 表中的 “priceEach”。可以看出, 交互机制在节点映射这一部分准确率的提升, 大大影响了整体模型的准确率。2) 在作者构造的自然语言查询意图数据集中, 简单复杂度的查询意图大体上是返回某数据表中某一字段, 中等复杂度的查询意图会添加一些聚合函数和简单过滤条件, 困难复杂度的意图会增加更多的聚合函数和跨表查询, 总体上来说结构都比较简单。在表 3-5 所示的结果中, 困难复杂度意图的错误生成情况, 一部分来源于查询意图对应的 SQL 语句需要子查询, 如 “return the customer who has the most orders”。目前的模型还不能很好的处理这种情况, 在解析树语法和翻译过程中都还没考虑子查询的情况, 这可能是接下来需要进一步进行的工作, 即提升模型的处理能力, 使其能处理更复杂的查询意图。3) 目前模型基于相似度的节点映射机制仍然有不足之处, 如 “return the mobile number of customer whose name is Australian Gift Network” 这一查询意图, 对于顾客名称 “Australian Gift Network”, 系统会将其映射为三个不同的节点, 导致生成出错。尽管模型对于数据库模式中的一些连接词或短语, 如 “thenumberof”、“customername”, 进行了特殊处理, 但对于上文所示的这些特殊短语或词语, 目前没有较好的方法来进行处理。4) 本文自行建立了一个解析树节点类型与自然语言单词的映射词典 (3.3 节), 能处理一些常见的单词与 SQL 语法的对应关系, 如 “return”、“in”、“have”、“thenumberof”, 但映射关系仍然不足, 所以本实验所使用的测试查询意图都需要使用这些比较固定的词语来构建。事实上, 自然语言的表达非常多样, 如果要增加模型的处理能力, 扩充这个映射词典也是非常必要的。5) 尽管人机交互机制对于节点映射的准确率有较大提升, 但根据观察, 立足于目前数据量较小、数据库模式简单的前提下, 这一机制能保证映射相似度排名前五的 SQL 组件包含正确的映射关系; 但随着数据库数据量的增加、数据库模式的复杂化, 目前节点映射的相似度计算机制, 可能无法确保正确的 SQL 组件能

有较高的相似度。

2.5 本章小结

第三章 基于深度强化学习的 NL2SQL 生成

3.1 研究问题

3.2 相关技术

3.3 解决方案

给定一个输入自然语言问题，我们的目标是生成相应的 SQL 查询。在下文和整篇论文中，我们使用 WikiSQL 数据集 (Zhong et al., 2017) 作为我们的激励示例。但是，应该注意的是，我们的方法通常适用于其他 NL2SQL 数据，可以正确选择操作清单并重新设计解析器状态。

WikiSQL 数据集包含 80,654 对问题和 SQL 查询，分布在维基百科的 24,241 个表中。与自然语言问题一起，输入还包含单个表模式（即表列名称）。每个表只存在于一个拆分（火车，开发或测试）中，这要求模型推广到看不见的表。图 1 显示了一个示例。WikiSQL 数据集查询的 SQL 结构受到限制，并始终遵循模板 `SELECT agg selcol WHERE col op val (AND col op val) *`。在这里，selcol 是一个

单表列和 agg 是聚合器（例如，COUNT，SUM，空）。WHERE 段是一系列连接过滤条件。每个 op 是过滤运算符（例如，=），并且在问题中提到过滤值 val。虽然数据集带有条件的“标准”线性排序，但鉴于 AND 的语义，顺序实际上是无关紧要的。

在整篇论文中，我们将解析器的输入表示为 x 。它由带有标记 w_i 的自然语言问题 w 和具有列名称 c_j 的单个表模式 c 组成。列名称 c_j 可以包含一个或多个标记。解析器需要生成可执行的 SQL 查询 y 作为其输出。

给定输入 x ，结构化输出 y 的生成被分解为一系列解析决策。解析器从初始状态开始，并根据学习的策略逐步采取操作。每个动作都将解析器从一个状态推进到另一个状态，直到它到达终端状态之一，在那里我们可以提取完整的逻辑形式 y 。我们采用概率方法塑造政策。它在给定输入 x 和运行的解码历史的情况下预测有效的后续动作集合上的概率分布。然后，训练这种增量语义解析器的目标是优化该参数化策略。

965/5000 形式上，我们让 $P(y|x) = P(a|x)$ ，其中 θ 是模型参数。执行动作序列 $a = a_1, a_2, \dots, a_k$ 将解析器从初始状态引导到包含解析结果 y 的终端状态。这里我们假设每个 y 只有一个相应的动作序列 a ，我们将在第 4 节重新讨论这个假设。动作序列的概率被进一步考虑为增量决策概率的乘积： $P(a|x) = \prod_{i=1}^k P(a_i|x, a_{<i})$ ，其中 $|a| = k$ 。在推理期间，我们的解码器不是试图枚举整个输出空间并找到最高得分 $a^* = \arg$

Action	Resulting state after taking the action at state p	Parameter representation
AGG(agg)	$p[\text{AGG} \mapsto \text{agg}]$	—
SELCOL(c_i)	$p[\text{SELCOL} \mapsto c_i]$	r_i^C
CONDCOL(c_i)	$p[\text{COND} \mapsto p.\text{COND} \parallel \begin{bmatrix} \text{COL} & c_i \\ \text{OP} & \epsilon \\ \text{VAL} & \epsilon \end{bmatrix}]$	r_i^C
CONDOP(op)	$p[\text{COND}_{-1} \mapsto p.\text{COND}_{-1}[\text{OP} \mapsto \text{op}]]$	—
CONDVAL($w_{i:j}$)	$p[\text{COND}_{-1} \mapsto p.\text{COND}_{-1}[\text{VAL} \mapsto w_{i:j}]]$	r_i^W and r_j^W
END	p (as a terminal state)	—

图 3-1 中文题图

Figure 3-1 English caption

$\max_a P(a|x)$ ，而是采用贪婪的方法：在每个中间步骤，它根据以下方式选择最高的得分行动：政策： $a^* = \arg \max_a P(a|x, a^* < I)$ 。在以下小节中，我们定义解析器状态和动作清单，然后描述编码器 - 解码器神经网络模型体系结构。

我们的解码器的主要组成部分是对概率分布 $P(a|x, a < i)$ 进行模拟解析器动作以输入 x 和过去动作 $a < i$ 为条件。它有两个主要挑战：（1）没有固定的有效解析器动作集：它取决于输入和当前解析器状态；（2）解析器决策依赖于上下文：它依赖于解码历史和信息嵌入在输入问题和列标题中。我们采用基于 LSTM 的解码器框架，通过个人解决第一个挑战行动得分。该模型将每个候选动作 a 评分为 s_a 并使用 softmax 函数将分数标准化为概率分布。在时间步骤 i ，我们表示隐藏的当前解码器以 h 为状态

3.4 实验与分析

3.5 本章小结

第四章 基于多任务学习的 NL2SQL 生成

4.1 研究问题

4.2 相关技术

4.3 解决方案

4.4 实验与分析

4.5 本章小结

Machine Comprehension Question answering (QA) models receive a question and a context that contains information necessary to output the desired answer. We use the Stanford Question Answering Dataset (SQuAD) [1] for this task. Contexts are paragraphs taken from the English Wikipedia, and answers are sequences of words copied from the context. SQuAD uses a normalized F1 (nF1) metric that strip out articles and punctuation. Machine Translation. Machine translation models receive an input document in a source language that must be translated into a target language. We use the 2016 English to

German training data prepared for the International Workshop on Spoken Language Translation (IWSLT) [2]. Examples are from transcribed TED presentations that cover a wide variety of topics with conversational language. We evaluate with a corpus-level BLEU score [3] on the 2013 and 2014 test sets as validation and test sets, respectively. Natural Language Inference. Natural Language Inference (NLI) models receive two input sentences: a premise and a hypothesis. Models must then classify the inference relationship between the two as one of entailment, neutrality, or contradiction. We use the Multi-Genre Natural Language Inference Corpus (MNLI) [4] which provides training examples from multiple domains (transcribed speech, popular fiction, government reports) and test pairs from seen and unseen

domains. MNLI uses an exact match (EM) score. Sentiment Analysis. Sentiment analysis models are trained to classify the sentiment expressed by input text. The Stanford Sentiment Treebank (SST) [5] consists of movie reviews with the corresponding sentiment (positive, neutral, negative). We use the unparsed, binary version [6]. SST also uses an EM score. Semantic Parsing. SQL query generation is related to semantic parsing. Models based on the WikiSQL dataset [7] translate natural language questions into structured SQL queries so that users can

interact with a database in natural language. WikiSQL is evaluated by a logical form exact match (lFEM) to ensure that models do not obtain correct answers from incorrectly generated queries.

Summarization. Summarization models take in a document and output a summary of that document. Most important recent progress in summarization was the transformation of the CNN/DailyMail (CNN/DM) corpus [Hermann et al., 2015] into a summarization dataset [Nallapati et al., 2016]. We include the non-anonymized version of this dataset in decaNLP. On average, these examples contain the longest documents in decaNLP and force models to balance extracting from the context with generation of novel, abstractive sequences of words. CNN/DM uses ROUGE-1, ROUGE-2, and ROUGE-L scores [Lin, 2004]. We average these three measures to compute an overall ROUGE score.

Sentiment Analysis. Sentiment analysis models are trained to classify the sentiment expressed by input text. The Stanford Sentiment Treebank (SST) [Socher et al., 2013] consists of movie reviews with the corresponding sentiment (positive, neutral, negative). We use the unparsed, binary version [Radford et al., 2017]. SST also uses an EM score.

Semantic Role Labeling. Semantic role labeling (SRL) models are given a sentence and predicate (typically a verb) and must determine who did what to whom, when, and where [Johansson and Nugues, 2008]. We use an SRL dataset that treats the task as question answering, QA-SRL [He et al., 2015]. This dataset covers both news and Wikipedia domains, but we only use the latter in order to ensure that all data for decaNLP can be freely downloaded. We evaluate QA-SRL with the nF1 metric used for SQuAD.

Relation Extraction. Relation extraction systems take in a piece of unstructured text and the kind of relation that is to be extracted from that text. In this setting, it is important that models can report that the relation is not present and cannot be extracted. As with SRL, we use a dataset that maps relations to a set of questions so that relation extraction can be treated as question answering: QA-ZRE [Levy et al., 2017]. Evaluation of the dataset is designed to measure zero shot performance on new kinds of relations: the dataset is split so that relations seen at test time are unseen at train time. This kind of zero-shot relation extraction, framed as question answering, makes it possible to generalize to new relations. QA-ZRE uses a corpus-level F1 metric (cF1) in order to accurately account for unanswerable questions. This F1 metric defines precision as the true positive count divided by the number of times the system returned a non-null answer and recall as the true

positive count divided by the number of instances that have an answer.

Goal-Oriented Dialogue. Dialogue state tracking is a key component of goal-oriented dialogue systems. Based on user utterances, actions taken already, and conversation history, dialogue state trackers keep track of which predefined goals the user has for the dia-

loguesystem and which kinds of requests the user makes as thesystem and user interact turn-by-turn. We use the EnglishWizard of Oz (WOZ) restaurant reservation task [Wen et al.,2016], which comes with a predefined ontology of foods,dates, times, addresses, and other information that would help an agent make a reservation for a customer. WOZ is evaluated by turn-based dialogue state EM (dsEM) over the goals of thecustomers.Semantic Parsing.SQL query generation is related to semantic parsing. Models based on the WikiSQL dataset[Zhong et al., 2017] translate natural language questions into structured SQL queries so that users can interact with a database in natural language. WikiSQL is evaluated by a logical form exact match (lfEM) to ensure that models do not obtain correct answers from incorrectly generated queries.Pronoun Resolution.Our final task is based on Winograd schemas [Winograd, 1972], which require pronoun resolution: "Joan made sure to thank Susan for the help she had[given/received]. Who had [given/received] help? Susan orJoan?" . We started with examples taken from the WinogradSchema Challenge [Levesque et al., 2011] and modified them to ensure that answers were a single word from the context.This modified Winograd Schema Challenge (MWSC) ensures that scores are neither inflated nor deflated by oddities in phrasing or inconsistencies between context, question, and answer. We evaluate with an EM score.The Decathlon Score (decaScore).Models competing on decaNLP are evaluated using an additive combination of each task-specific metric. All metrics fall between 0 and 100, so that the decaScore naturally falls between 0 and 1000 for ten tasks.Using an additive combination avoids issues that arise from weighing different metrics. All metrics are case insensitive.As shown in Table II. All metrics are case insensitive. nF1 is the normalized F1 metric used by SQuAD that strips out articles and punctuation. EM is an exact match comparison:for text classification, this amounts to accuracy; for WOZ it is equivalent to turn-based dialogue state exact match (dsEM)and for WikiSQL it is equivalent to exact match of logical forms (lfEM). F1 for QA-ZRE is a corpus level metric (cF1)that takes into account that some questions are unanswerable.Precision is the true positive count divided by the number of times the system returned a non-null answer. Recall is the true positive count divided by the number of instances that have an answer.

Because every task is framed as question answering and trained jointly, we call our model a multitask question answer-

ing network (MQAN). Each example consists of a context, question, and answer as shown in Fig. 1. Many recent QA models for question answering typically assume the answer can be copied from the context [Wang and Jiang, 2017, Seo et al., 2017, Xiong et al., 2018], but this assumption does not hold for general question answering. The question often contains key

information that constrains the answer space. Noting this, we extend the coattention of [Xiong et al., 2017] to enrich the representation of not only the input but also the question. Also, the pointer-mechanism of [See et al., 2017] is generalized into a hierarchical, multi-pointer-generator that enables the capacity to copy directly from the question and the context. During training, the MQAN takes as input three sequences: a context c with l tokens, a question q with m tokens, and an answer a with n tokens. Each of these is represented by a matrix where the i th row of the matrix corresponds to a d -dimensional embedding (such as word or character vectors) for the i th token in the sequence:

Encoder: xxxxx Decoder: xxxxx

第五章 总结与展望

5.1 本文工作小结

5.2 展望

参考文献

- [1] ANDROUTSOPOULOS I, RITCHIE G D, THANISCH P. Natural Language Interfaces to Databases - An Introduction[J]. Natural Language Engineering, 1995, 1(1): 29–81.

致 谢

感谢所有测试和使用交大学位论文 \LaTeX 模板的同学！

感谢那位最先制作出博士学位论文 \LaTeX 模板的交大物理系同学！

感谢 William Wang 同学对模板移植做出的巨大贡献！

感谢 @weijianwen 学长一直以来的开发和维护工作！

感谢 @sjtug 以及 @dyweb 对 0.9.5 之后版本的开发和维护工作！

感谢所有为模板贡献过代码的同学们, 以及所有测试和使用模板的各位同学！

攻读学位期间发表的学术论文

- [1] CHEN H, CHAN C T. Acoustic cloaking in three dimensions using acoustic metamaterials[J]. Applied Physics Letters, 2007, 91:183518.
- [2] CHEN H, WU B I, ZHANG B, et al. Electromagnetic Wave Interactions with a Metamaterial Cloak[J]. Physical Review Letters, 2007, 99(6):63903.

攻读学位期间参与的项目

- [1] 973 项目 “XXX”
- [2] 自然科学基金项目 “XXX”
- [3] 国防项目 “XXX”

简 历

基本情况

某某，yyyy 年 mm 月生于 xxxx。

教育背景

yyyy 年 mm 月至今，上海交通大学，博士研究生，xx 专业

yyyy 年 mm 月至 yyyy 年 mm 月，上海交通大学，硕士研究生，xx 专业

yyyy 年 mm 月至 yyyy 年 mm 月，上海交通大学，本科，xx 专业

研究兴趣

L^AT_EX 排版

联系方式

地址：上海市闵行区东川路 800 号，200240

E-mail: xxx@sjtu.edu.cn