Software Engineering and Applications 软件工程与应用, 2014, 3, 9-14
http://dx.doi.org/10.12677/sea.2014.31002 Published Online February 2014 (http://www.hanspub.org/journal/sea.html)

The Automatic Marking Method of SQL Script Based on Syntax Analysis and Levenshtein Distance

Xia Liu, Li Tao, Yuhong Zhou, Kexin Ma, Xiaoqiang Liu

Donghua University, Shanghai
Email: liuxia880818@163.com

Received: Jan. 11th, 2014; revised: Feb. 4th, 2014; accepted: Feb. 11th, 2014

Abstract: An automatic marking algorithm based on syntax analysis and string Levenshtein distance for SQL script in database tests was proposed. Firstly, through the lexical analysis and syntax analysis for student's answer, the syntax analysis tree is obtained. Secondly, the most matched standard answer is chosen from the standard answers set by comparing the string edit distance between the student's answer and the standard answers. Thirdly, by comparing the similarity of all the corresponding nodes of the two syntax analysis tree, the mark is achieved based on the score value of the standard answer. The marking system is implemented using open source parser—ANTLR3.4 and VS2010. This automatic marking method not only can give reference scores, but also point out the error position. Furthermore, it supports efficient and accurate marking for multi-answers with fault tolerance.

Keywords: SQL Sentence; Automatic Marking; Lexical Analysis; Syntax Analysis; Levenshtein Distance

采用语法分析和编辑距离的 SQL 脚本评阅方法

刘　霞，陶　莉，　周余洪，马可辛，刘晓强

东华大学，上海
Email: liuxia880818@163.com

收稿日期：2014 年 1 月 11 日；修回日期：2014 年 2 月 4 日；录用日期：2014 年 2 月 11 日

摘　要：针对数据库测试中 SQL 脚本类题目，提出基于语法分析和字符串编辑距离相结合的自动阅卷算法。该算法首先对答案进行词法分析和语法分析，获得 SQL 语句的语法分析树，并基于字符串编辑距离从标准答案集中选择最匹配的标准答案，以及对语法树的节点进行相似度比较，并根据评分标准，实现题目的自动评阅。阅卷系统采用开源语法分析器——ANTLR3.4 和 VS2010 实现自动评阅。该自动阅卷算法得到参考评分，同时可以指出错误点，并且更好地支持答案的多样性和容错性，实现高效率、较精确的阅卷结果。

关键词：SQL 语句；自动阅卷；词法分析；语法分析；字符串编辑距离

1. 引言

目前针对编程题的自动评阅方法很多[1-8]，多数借助编译系统，在正确编译的前提下运行脚本，通过检查运行结果是否正确给出得分，这种方法存在以下缺点：

(1) 进行自动评阅的脚本必须能运行，若有编译错误或其他错误无法给出得分。

(2) 被评阅的脚本必须有输出，对于一些没有输出结果的脚本不能进行评阅。

(3) 此评阅方法只注重结果不注重过程，若考生

粗心写错关键字导致脚本不能运行则无法进行评分，这与自动评阅的初衷相违背。

针对目前自动评阅存在的问题，本文提出基于语法分析和编辑距离的自动评阅方法，采用此方法可对源脚本进行评阅，突破脚本必须运行才能评阅的限制，利用文本相似度技术对考生答案和标准答案进行相似度比较，最终得到评分。

2. SQL 脚本类题目的评阅算法

针对 SQL 脚本类题目自动评阅问题，本文提出在对脚本运行结果评分的基础上基于语法分析和字符串编辑距离的评分算法(如图 1)。该算法的两个阶段工作过程如下：

(1) 标准答案预处理：首先基于 SQL 文法规则对 SQL 语句命题的标准答案进行分解，形成语法分析树，然后对各节点进行评分权值设定，将分解后的标准答案和评分标准保存为 XML 文件。

(2) 考生答案评分：首先基于 SQL 文法规则将考生 SQL 语句答案进行分解，形成语法分析树，得到分解后的学生答案；通过将考生语句答案与标准答案字符串进行基于编辑距离的相似度比较，从标准答案集中选择最匹配的标准答案；然后将分解后的学生答案与分解后的标准答案逐节点进行基于字符串编辑距离的比较，得到各节点的得分，并标记错误节点；最后汇总得到该语句的总评分。

下面对该算法中涉及的主要数据描述说明如下：

(1) SQL 文法规则

基于标准 SQL 语法定义得到 SQL 语句的产生式文法规则。由于 SQL 语句的语法表达的灵活性，一种 SQL 语句可能有多个不同形式的 SQL 语句文法规则，为了能评判各种可能的答案，需要给出一种 SQL 语句对应的所有形式的 SQL 文法规则。

(2) 标准答案集

命题者需要给出一种 SQL 语句命题的所有可能答案，形成一个标准答案集。在标准答案预处理阶段，将每一个答案进行分解；在评分阶段，首先将考生答案与标准答案集中的每一个答案进行基于字符串编辑距离的相似度比较，找到用于阅卷匹配的答案。

(3) 分解的标准答案、分解的学生答案

命题者给出的标准答案和学生给出的答案都以字符串描述，基于语法分析得到分解后的标准答案及评分标准被保存为 XML 文档的形式，XML 文档的节点对应语法树中的节点。考生答案分解后也被描述为 XML 文档。

3. 系统实现的关键技术

3.1. 词法分析和语法分析

词法分析是解释识别的基础，它逐个字符读取 SQL 脚本，从字符流中识别出关键字、标识符、常量等相对独立的记号 Token，形成记号流。词法分析过程中会过滤掉 SQL 脚本中的空格、换行符和注释等不属于 SQL 脚本的有效字符，还可以将记号归类，哪些记号属于标识符，哪些记号属于关键字、整数、浮点数等。

语法分析根据词法分析输出的记号流，基于 SQL 文法规则分析 SQL 脚本的语法结构，并添加代表语法结构的抽象单词，按照语法结构生成语法树的过程。语法分析后生成的抽象语法树含有结构化信息，而语法树的叶子节点就是 Token。

本课题采用 ANTLR 软件进行词法分析与语法分析[9]。ANTLR 是一种类 Yacc/Lex 的语言识别工具。它基于自顶向下的 LL(k)分析算法，能够把含有语义动作的语法描述文件转换成各种程序语言下的识别器、解释器和编译器。ANTLR 目前支持多种当前流行的开发语言，包括 Java、C#、C、C++、Python 和 Ruby 等，可以根据需要生成其中任何一种语言的语法分析器。

3.1.1. SQL 脚本处理规则定义

(1) Token 和终结符产生规则

词法分析定义 Token 的规则必须以字母开头，如“SELECT”，“Letter”。如表示一个字符的语法规则如下：

Letter ： 'a'..'z' |'A'..'Z' | '_' | '#' | '@' |'%';

该词法规则表示将检查到大小写英文字母或'_'、'#'、'@' 、'%'时，产生一个 Token:Letter。

在词法规则定义中，规定了终结符的定义方法，主要有以下几种：

“..”符号：通过指定首尾字符定义一定范围内的字符。

“~”符号：表示除某些符号以外的符号。

“.“符号：表示单个任意字符，起通配符的作用。

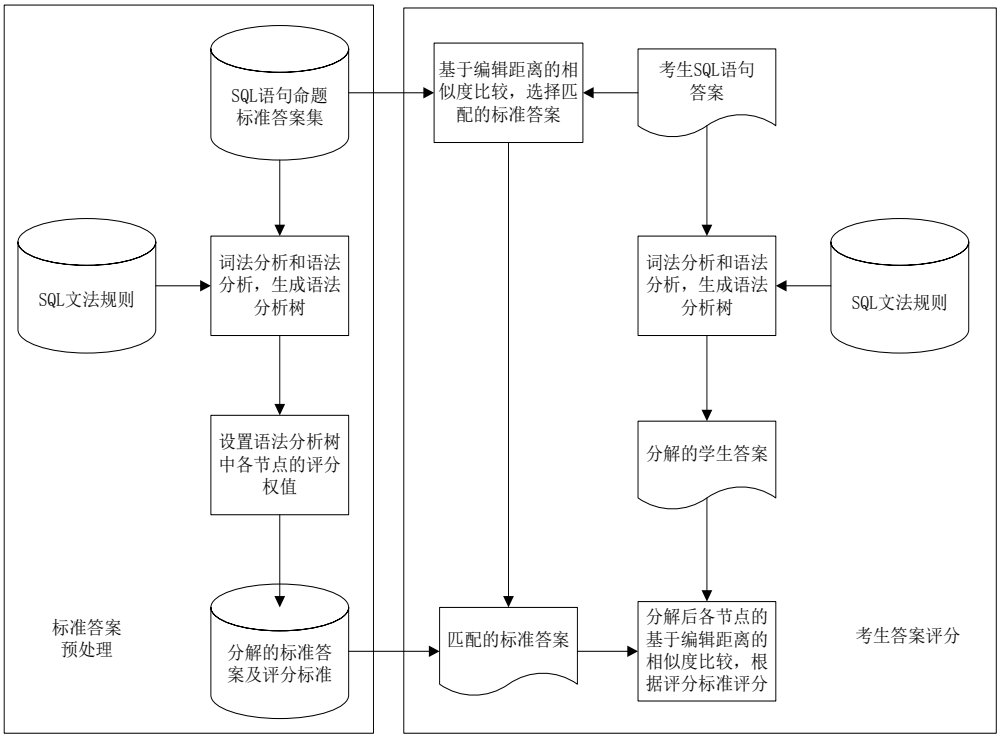

Figure 1. The marking algorithm based on syntax analysis and Levenshtein distance
图 1. 基于语法分析树和字符串编辑距离的评分算法

(2) 中文字符的处理

SQL 脚本中会出现中文字符，因此要考虑编码问题。定义汉字字符只能用十六进制编码定义法，使用”\u”开头加四位十六进制数定义来定义一个字符。

例如“Select * From School Where sex = ’男’”语句中，定义性别的语法规则为 Sex：’\u7537’|’\u5973’，相当于 Sex：’男’|’女’。

(3) Skip()方法

SQL 脚本中的有些字符，如空格、换行符和注释等在分析过程中应该忽略掉，可在词法定义中加入 Skip()方法来跳过。如下面定义了一个跳过空白的词法定义。

WS：(’ ’|’\t’|’\n’|’\r’) + skip()

空白符号 WS 中有空格符、制表符、回车和换行符，当遇到这些字符时词法分析程序会调用 Skip()方法跳过这些字符。

3.1.2. SQL 脚本语法树

为了生成抽象语法树，需要在提供给 ANTLR 的语法文件中说明构造规则。在对考生答案和标准答案集进行语法分析时，按照已规定的语法规则进行语法解析。同一条 SQL 脚本语句对于不同的语法规则生成不同的语法树，语法树构造规则以规则表达式说明。例如下面的文法片段是 SQL 数据表创建语句的规则表达式。

```
statement
    :createTable
;
createTable
    :createClause LPAREN columnClause (COMMA columnClause)* RPAREN
;
createClause
    :‘create’ ‘table’ tableName
;
columnClause
    :columnName columnType (columnConstaint)*
;
columnName
    :identifier
;
tableName
```

```
        :identifier
    ;
    constrainName
        :identifier
    ;
    identifier
        :NonQuotedIdentifier| QuotedIdentifier
    ;
    columnType
        :identifier (LPAREN (Integer|identifier) COMMA (Integer)? RPAREN)?
    ;
    columnConstaint
        :NotnullToken
        |NullToken
        |UniqueToken
        |PriToken
    ;
```

例如，对 SQL 语句 create table School (SchoolCode char(4) primary key)进行语法分析后得到的抽象语法树如图 2 所示。

通过定义 SQL 主要语句的文法，可将 SQL 语句解析成语法分析树的形式，因为 XML 文档是以树的结构存储数据的，因此将此分析树保存为 XML 文档形式，之后可进行 XML 节点的比较。

3.1.3. 错误处理

错误处理是进行语法分析的重要组成部分，ANTLR 可以产生默认的错误处理代码，也可以指定自己的异常处理代码。出现文法错误时，ANTLR 会生成 try/catch 语句块。在进行语法分析和词法分析时，ANTLR 会进行语法错误检测，当脚本不符合语法规则时，不能正常生成 AST (抽象语法树)。ANTLR 中定义了多种异常，如 MismatchedTokenException (Token 匹配异常)、NoviableAltException(无可用产生式异常)等。当前 Token 无法分析时，产生错误节点，抛出相应的异常以方便后面进行错误恢复。

为提高系统的容错性，在检测到错误时能继续进行分析，ANTLR 的错误恢复机制有以下几种：尝试丢弃当前 Token、尝试插入一个 Token、检查后续 Token 等。

3.2. 基于编辑距离的答案相似度比较

3.2.1. 基于编辑距离的字符串相似度比较

字符串的编辑距离，又称 Levenshtein 距离，是指两个字符串中由一个字符串转换为另一个所需的最少编辑操作次数。编辑操作包括替换、删除与插入，计算两个字符串编辑距离的算法如下：

设有字符串 S 和 T，$S=S1,S2,S3,\cdots,Sn$，$T=T1,T2,\cdots,Tm$,建立S和T的$(n+1)\times(m+1)$阶匹配关系矩阵：

$$D_{(n+1)\times(m+1)}=\{d_{ij}\}(0\le i\le n,0\le j\le m) \quad (1)$$

$$d_{ij}=\begin{cases} i & i=0 \\ j & j=0 \\ \min\left(d_{(i-1)j},d_{i(j-1)},d_{(i-1)(j-1)}+a(i,j)\right) & i,j>0 \end{cases} \quad (2)$$

其中：

$$a(i,j)=\begin{cases} 0 & s_i=t_i \\ 1 & s_i\ne t_i \end{cases} \quad (3)$$

$(i=1,2,\cdots,n,j=1,2,\cdots,m)$，则矩阵 D 的右下角的元素 $D(n,m)$即为 S 和 T 的 Levenshtein 距离 LD。

在取得两个字符串的编辑距离后，可用算式 $1=\frac{LD}{L1+L2}$ 计算两个字符串的相似度度量，其中 LD 是字符串 S 与字符串 T 的编辑距离，L1、L2 分别是字符串 S 与字符串 T 的字符串长度。

根据算式得知，两字符串的编辑距离越大，两者的相似度越小。

3.2.2. 字符串相似度在答案评价中的应用

(1) 选择匹配的标准答案

由于标准答案集中对应一种 SQL 语句命题可能有多个标准答案，而考生答案只有一个，所以在对考生答案评价前，首先需要找出与考生答案语法描述最接近的一个标准答案。通过计算考生答案与该命题的标准答案集中的每个标准答案的字符串编辑距离，找到与考生答案相似度最高的标准答案，然后再基于语法分析树进行评分。这样可以支持多重答案，并且大大提高阅卷效率。

例如，有考生答案：Select Grade From Grade Where Grade Between 80 And 90，有标准答案 1：Select

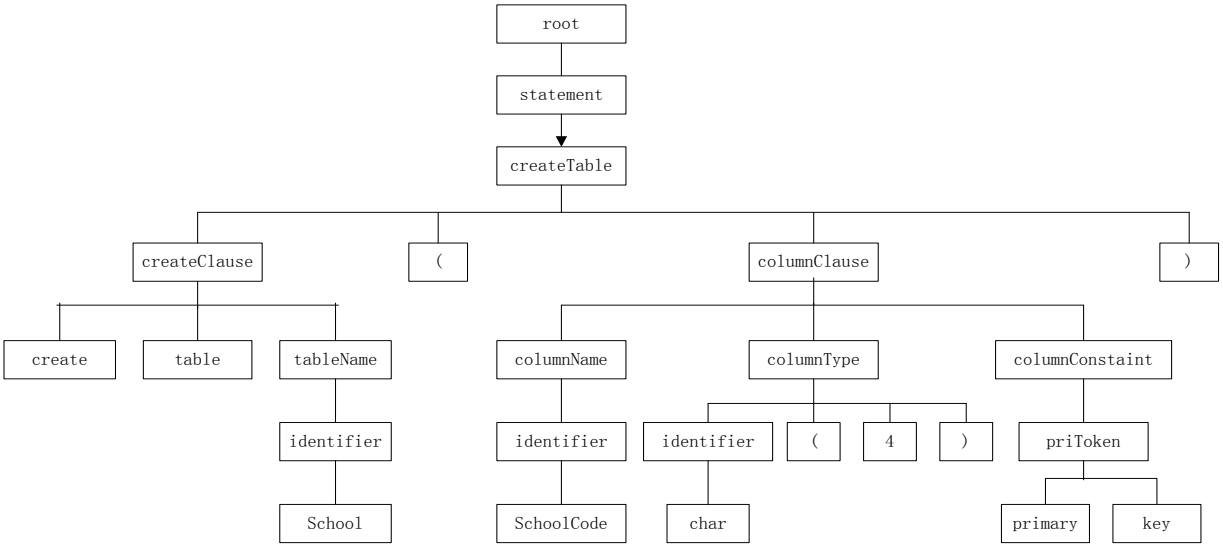

Figure 2. **The analysis tree of Create sentence**
图 2. **一条 Create 语句的语法分析树**

Grade From Grade Where Grade Between 80 And 90 和标准答案 2：Select Grade From Grade Where Grade ≥ 80 And Grade ≤ 90。根据可知考生答案与标准答案 1 的相似度为脚本的字符 1，与标准答案 2 的相似度为 0.853，因此将考生答案与标准答案 1 进行比较。

(2) 标识符的识别、匹配和语法树评分

SQL 语句语法分析树中的所有叶子节点对应了答案分解后的标识符，有 SQL 子句关键字，也有用户自定义标识符，最终评分依据评分标准中对一个叶子节点或几个相邻叶子节点设定的分值标准进行评分。若考生由于粗心将标识符写错，如对一个表名的定义，考生答案中误将表名写为“Studnt”，而标准答案集中对表名的定义为“Student”，人工阅卷时会酌情给分。所以系统将考生答案 XML 节点和标准答案对应的 XML 节点中的标识符进行字符串相似度比较，即使有错误，也可根据相似度情况决定相应分数，提高系统容错性。

4. 系统应用实例

下面以一个简单的 Select 语句为例，说明本文提出方法的有效性。

有一个学生表 Student (StuCode, StudentName)和一个成绩表 Grade (StuCode, Grade)，其中 StuCode 为学生学号，StudentName 为学生姓名，Grade 为学生成绩，现要求查询学号为“2001”的学生的姓名、成绩，本题 3 分。

教师定义的标准答案集如表 1，答案中各评分节点的评分权值分别为：A 节点 0.5 分，B 节点 0.5 分，C 节点 1 分，D 节点 1 分。

Table 1. **the set of standard answers**
表 1. **标准答案集**

答案编号	评分节点	标准答案
1	A	Select StudentName, Grade
	B	From Student, Grade
	C	Where Student.StuCode = Grade.StuCode
	D	And Grade.StuCode = ‘2001’
2	A	Select StudentName As 姓名, Grade As 成绩
	B	From Student, Grade
	C	Where Student.StuCode = Grade.StuCode
	D	And Grade.StuCode = ‘2001’
3	A	Select StudentName As 姓名, Grade As 成绩
	B	From Student Join Grade
	C	On Student.StuCode = Grade.StuCode Where
	D	Grade.StuCode = ‘2001’
4	A	Select StudentName,Grade
	B	From Student Join Grade
	C	On Student.StuCode = Grade.StuCode Where
	D	Grade.StuCode = ‘2001’

系统将标准答案集中的所有标准答案进行词法分析和语法分析，将生成的语法分析树保存为 XML 文档。

如果考生给出答案为：

Select StudentName,Grade

From Student,Grade

Where Student. StuCode = Grade.StuCode And Grade. StuCode = ‘2011’

这对考生答案，系统的评分过程如下：

(1) 执行考生答案，与标准答案的执行结果进行比较，并不相同。

(2) 将考生答案进行词法分析和语法分析，将生成的语法分析树保存为 XML 文档，本例词法分析和语法分析没有错误。

(3) 计算考生答案与标准答案集中的各个答案的相似度，考生答案与标准答案 1、2、3、4 的相似度分别为 0.995、0.925。0.861、0.921，可知与最相近的标准答案是 1。

(4) 将标准答案 1 的 XML 文档和考生答案的 XML 文档进行节点比较，如果不完全匹配，则计算字符串相似度，并指出错误点。根据节点得分值相加得到考生此题的分数为 2.5 分，并指出其错误在 D 节点。如果在评分标准中对 D 节点的分值进一步细分，由于“2011”书写错误的扣分可以进一步减小。

传统自动阅卷算法进行脚本的运行，运行考生答案和标准答案生成结果文件，之后进行结果文件的比较，以上试题为例，由于考生粗心写错标识符，考生运行结果文件与标准答案运行结果文件不同，用传统自动阅卷算法的得分为 0，违背了自动阅卷的初衷。本文提出的自动评阅方法，在针对触发器等无输出结果的脚本情况下也可进行评阅，是真正意义上的针对过程的自动评阅方法。

5. 结语

本文实现针对 SQL 脚本类题目的自动评阅，采用基于语法分析树和字符串编辑距离的评分算法，有效实现了题目的准确评价，并提供一定的容错能力，指出错误点。针对一个题目多重答案的问题也可实现高效评价。本方法要求在评阅时给出尽可能全面的标准答案集和语法形式，否则可能会存在一定的误判率。

参考文献 (References)

[1] 柏雪 (2013) 主观题自动阅卷系统的研究与设计. 硕士论文, 西南交通大学教育技术学, 成都.

[2] Burstein, J., Chodomw, M. and Leaeoek, C. (2004) Automatic essay evaluation: The criterion online writing service. *AI Magazine*, **25**, 27-36.

[3] Flesca, S., Masciari, E., et al. (2005) Fast detection of XML structural similarity. *IEEE Transactions on Knowledge and Data Engineering*, **17**, 160-175.

[4] 由东友 (2012) SQL Server 考核自动阅卷系统设计与实现. 硕士论文, 吉林大学, 长春.

[5] 陈尧妃, 陈焕通 (2009) SQL Server 数据库技能测评方案的设计与实现. *计算机应用与软件*, **11**, 147-149.

[6] 孙鸿伟 (2011) 基于相似度计算的编程题自动评判方法研究. 硕士论文, 哈尔滨工程大学, 哈尔滨.

[7] 周汉平 (2011) Levenshtein 距离在编程题自动评阅中的应用研究. *计算机应用与软件*, **5**, 328-331.

[8] 杨巍巍 (2013) 单向贴近度匹配法在主观题自动阅卷中的应用. *数字技术与应用*, **4**, 77-78.

[9] 潘旭, 康慕宁 (2011) 基于 ANTLR 的试卷识别和导入系统的研究. *电子设计工程*, **7**, 45-49.