基于组合范畴语法的中文语义分析

说明文档

章晔

2016/5/2

**目录**

一、组合范畴语法 3

1.1 词典 3

1.2 语法规则 4

二、逻辑表达式 4

2.1 Lambda Calculus 4

三、模型设计 5

3.1对数线性模型 6

3.2特征 6

3.3参数估计 6

四、词典生成 7

4.1词典初始化 7

4.2 新词条生成 7

4.3 词条拆分 8

五、数据标注 10

5.1标注方案 10

5.2数据说明 11

六、实验 11

# 

# **一、组合范畴语法**

组合范畴语法（Combinatory Categorial Grammar）是一种将句法（syntax）和语义（semantics）紧密结合的语言学表示，可以用来模拟广泛的语言现象。用于语义分析的组合范畴语法主要包含词典和组合规则。

## 1.1 词典

词典中每个词条以如下形式表示：

words :- X : h

其中words表示自然语言中的短语或句子，X表示该词组的句法成分，h是一个逻辑表达式，表示该短语对应的语义。

句法成分中，S表示句子，NP表示名词成分。在本项目中，所有的句法成分都用S和NP的组合表示。比如S\NP表示该短语可以与前面的名词成分NP构成一个句子S，S/NP表示该短语可以与后面的名字成分NP构成一个句子S。S|NP表示S\NP或者S/NP均可，例如S/(S|NP)后面接S/NP、S\NP或S|NP都可以构成一个句子S。

逻辑表达式用Lambda Calculus表示。

词条示例：

西单 :- NP : 西单:s

附近有 :- S\NP/NP : λxλy.zone(x, y)

## 1.2 语法规则

组合范畴语法根据一些组合规则，将相邻的短语连接起来，形成组合的句法成分和语义。本项目中用到的是四条基本规则：

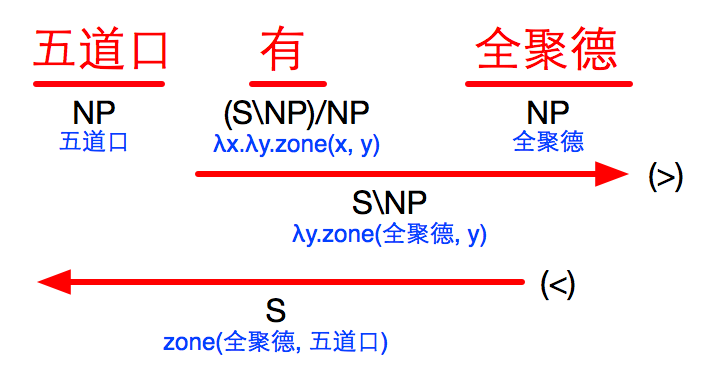
X/Y:f Y/Z:g => X/Z:λx.f(g(x)) forward composition (>B)

X/Y:f Y:g => X:f(g) forward apply (>)

Y\Z:g X\Y:f => X\Z:λx.f(g(x)) backward composition (<B)

Y:g X\Y:f => X:f(g) backward apply (<)

比如对于“五道口有全聚德”这句话，由三个短语“五道口”“有”“全聚德”构成，对相邻的短语根据规则进行组合，最后得到整个句子的句法成分是S，表示这是一句完整的语句，并且解析出这句话的语义。



# **二、逻辑表达式**

语义分析（Semantic Parsing）是将自然语句转化成逻辑表达式来表示语义。本项目逻辑表达式使用Lambda Calculus。

## 2.1 Lambda Calculus

Lambda Calculus有三种类型：t表示真值类型，e表示实体类型，i表示数字。Lambda Calculus中也有函数，即表示实体之间的关系。函数类型由输入类型和返回类型组成，比如price(x)的类型是<e, i>，输入x是实体，返回值是数字；restaurant(x)的类型是<e, t>，输入是实体，返回值是真值。

其中实体e，还可以根据特定领域设定子类型。

Lambda Calculus表达式有以下几种部分构成：

常量，又分为名词常量和关系常量。名词常量，比如“中关村”“火锅”这些地名或者标签，名词常量都是实体，需要标明实体类型。关系常量，即根据需要自定义的关系函数，比如商圈，人均价格。关系常量也需要标明函数类型。

逻辑联结词，即与、或、非。

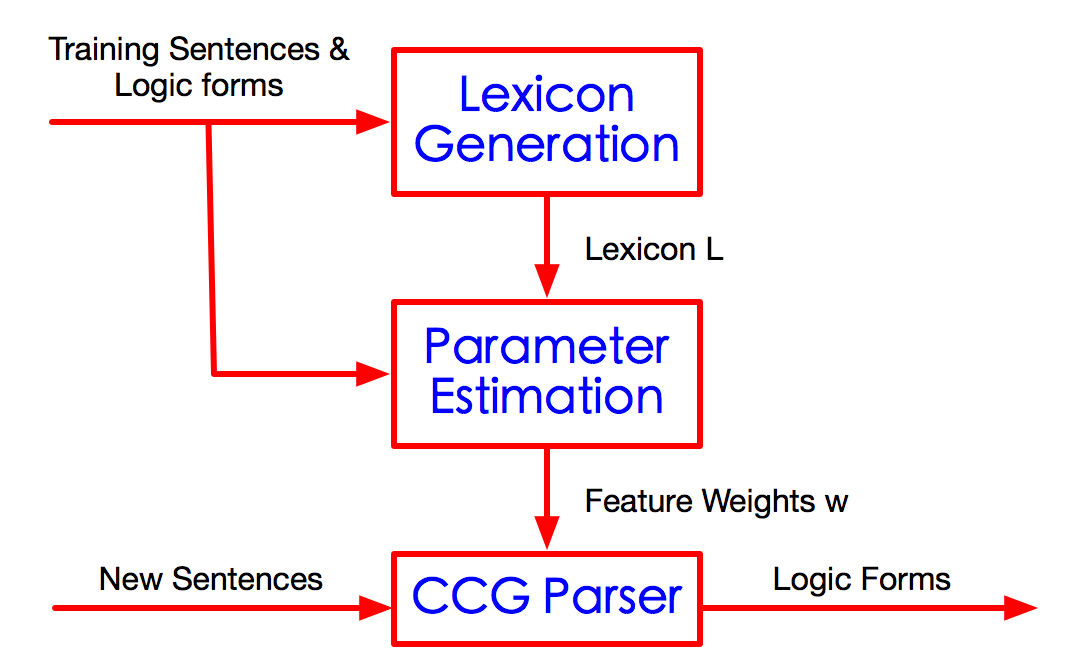
Lambda项，表示使得后面表达式为真的项，比如λx.zone(x, 五道口)表示存在(x, zone, 五道口)这种关系的实体。

逻辑表达式示例：

λx.restaurant(x)∧zone(x, 五道口)∧hasCuisine(x, 麻辣小龙虾)，表示五道口卖麻辣小龙虾的餐厅。

# **三、模型设计**

利用组合范畴语法进行语义分析的过程如下图所示。



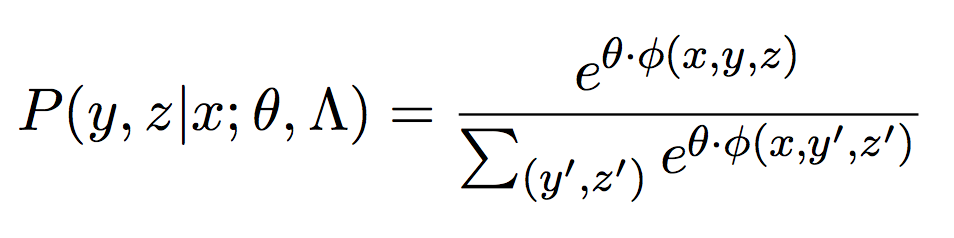
利用训练数据，即自然语句和对应的逻辑表达式，生成词典，并利用词典和训练数据进行参数估计，进行数轮迭代，最终得到一个组合范畴语法的语义分析器，包含一个词典和特征权重。

对于新的句子，通过语义分析器分析，可以得到对应的逻辑表达式。

## 3.1对数线性模型

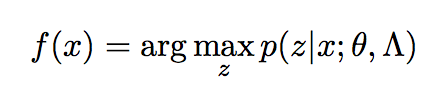
给定一个词典Λ和语法规则，对一个句子的语义分析可能产生很多结果。我们使用对数线性模型来选择可能性最高的结果。

给出一个句子x，对这个句子进行语义分析，经过解析过程y得到逻辑表达式z的概率为：

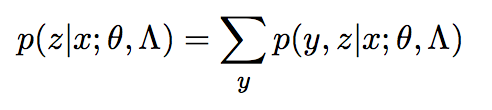


其中，φ是特征向量，θ是参数向量。

对于句子x，找出可能性最高的逻辑表达式，即：



其中，逻辑表达式z的概率为所有得到此表达式的分析过程的概率之和：



## 3.2特征

给出句子x，经过语义分析过程y得到逻辑表达式z，特征向量为使用了词典中哪些词条。参数向量为词典中每一个词条的权重。

## 3.3参数估计

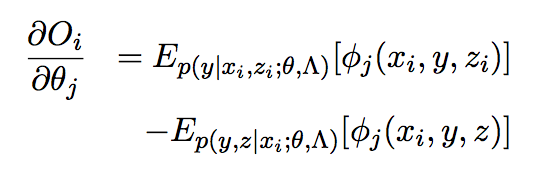
我们使用随机梯度下降法进行参数估计，对每个样本更新一次。

给定n对训练数据(xi, zi)，i=1…n。每轮迭代，对于每对训练数据，都更新参数。

每个样本的最大化目标函数为：



对θj求偏导计算梯度Δ，计算公式为：



更新参数：

θ=θ+αΔ

α为学习速率，随迭代增加而减小。

# **四、词典生成**

## 4.1词典初始化

首先，初始化词典。将训练数据中每一对样本<xi, zi>以词条的形式加入词典：

xi :- S : zi

然后，将固有名词加入词典，例如：

五道口 :- NP : 五道口:s

全聚德 :- NP : 全聚德:r

这些名词词条将有助于句子中名词实体的识别。

初始化词典中的词条权重初始值都为10。

由于中文不像英语一样有分词，需要对中文进行分词，将一句话或短语变成一串token，一遍后面进行拆分。

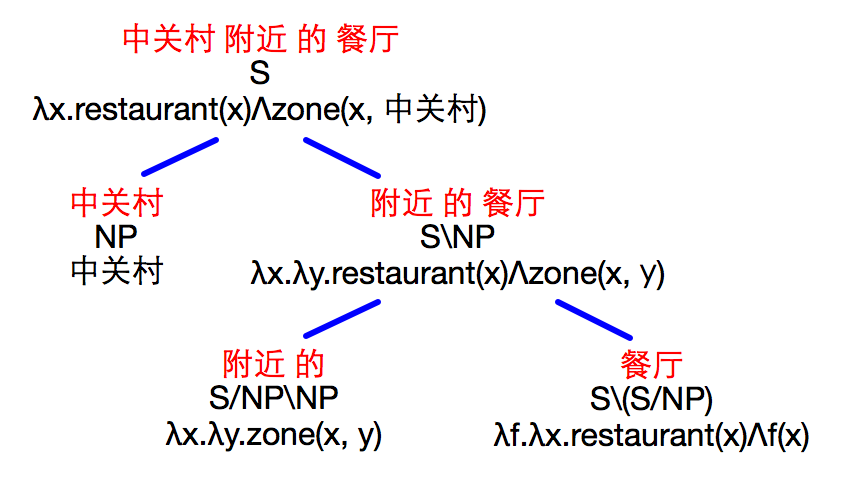
## 4.2 新词条生成

训练过程中，每轮迭代，对于每一个样本，根据已有词典，对自然语句x进行语义分析，产生当前得分最高的一个解析树y\*。

y\*中的叶节点即语义分析过程中用到的词条，而非叶结点是过程中产生的新词条。将非叶结点加入词典，并考虑其他可能产生此节点的解析：对非叶非根结点进行拆分，将得到的新词条也加入词典。

以下图“中关村附近的餐厅”一种解析树为例，利用“中关村”、“附近的”和“餐厅”三个词条，解析得到逻辑表达式。

解析树中有非叶非根节点“附近的餐厅”，将此词条加入词典，并对这个词条进行拆分，生成新的词条，这些新的词条都是可以产生“附近的餐厅”的词条，将这些新词条加入词典，并扩展特征向量。



## 4.3 词条拆分

语句是由短语构成的，组合范畴语法的本质就是根据词典理解每个短语的语义，并将短语组合的句子语义解析出来。

训练数据中的逻辑表达式是对一句完整的语句进行语义标注的，但是为了能够解析更多灵活变化的句子，需要得到每个短语的语义。这就是为什么要对解析树的非叶非根节点进行拆分得到新词条。

**4.3.1 拆分限制**

将逻辑表达式h拆分成f和g，f和g可以根据语法规则这，使得f(g)=h或者λx.f(g(x))=h。

如果不对f和g加以限制，可能会生成无数种可能性。例如:

h=五道口，f=λx.五道口，g可以是任何实体或关系，都能使得f(g)=h。再例如：

h=f1∧f2∧f3∧f4，h的任意子条件联结组合都可以分配给f或g。

为了限制拆分可能性的数量，对拆分词条作如下限值：

1、禁止无意义变量

f和g都不能是λx.e的形式，其中e是与变量x无关的逻辑表达式。

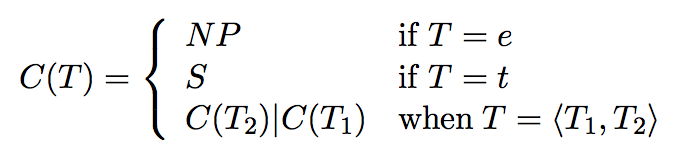
2、h=f1∧f2∧f3∧f4∧…时，g不能包含超过N个子条件。本项目中N为4。

3、f不能是f= λq.q(A)的形式，其中q是h中不存在的变量且q的输入是非变量。

**4.3.2 逻辑表达式句法成分拆分**

逻辑表达式h的句法成分为X，即X:h，将h拆分为f和g，f和g的句法成分取决于他们的类型。

以g为例，g的类型为T(g)，比如T(五道口)=e，T(λx.zone(x, 五道口))=<e, t>。类型T的句法成分为C(T)：



基本类型e的句法成分为名词成分NP，基本类型t的句法成分为句子S，所有其他类型的句法成分可以递归得到。例如C(<e, t>)=S|NP，C(<e, <e, t>>)=S|NP|NP。

分别对应四条语法规则：

X/Y:f Y/Z:g => X/Z:λx.f(g(x)) forward composition (FC)

X/Y:f Y:g => X:f(g) forward apply (FA)

Y\Z:g X\Y:f => X\Z:λx.f(g(x)) backward composition (BC)

Y:g X\Y:f => X:f(g) backward apply (BA)

以四种方式拆分逻辑表达式和句法成分：

FA(X:h) = {(X/Y:f, Y:g) | h=f(g) ∧ Y=C(T(g))􏰀

BA(X:h) = {(Y:g, X\Y:f) | h=f(g) ∧ Y=C(T(g))}

FC(X/Y:h) = {(X/W:f, W/Y:g) | h=λx.f(g(x)) ∧ W=C(T(g(x)))}

BC(X\Y:h) = {(W\Y:g, X\W:f) | h=λx.f(g(x)) ∧ W=C(T(g(x)))}

对A = X:h所有可能拆分SC(A):

SC(A) = {FA(A) ∪ BA(A) ∪ FC(A) ∪ BC(A)}

**4.3.3 词条拆分**

将逻辑表达式h和句法成分X拆分后，对短语（或句子）w进行拆分。

已分词的w由n个token组成，w0:n=<w0, w1, …, wn>。

对词条w0:n :- A的拆分S(w0:n :- A)：

SL(w0:n :- A) = {(w0:i :- B, wi+1:n :- C) | 0≤i<n ∧ (B,C)∈SC(A)}

即枚举短语w所有拆分方式，与逻辑表达式和句法所有可能拆分方式组合。

# **五、数据标注**

## 5.1标注方案

根据RDF知识库的设计方案，实体类型e的子类型有：

r，表示餐馆；

c，表示菜品；

s，表示字符串，餐馆和菜品的各种属性值类型都是s。

根据数据，定义以下关系：

(restaurant:t r t)，输入类型r，返回类型t，输入为餐馆时，返回真。

(name:t r s t)

(district:t r s t)

(zone:t r s t)

(street:t r s t)

(lat:i r i)

(lng:i r i)

(price:i r i)

(address:s r s)

(tel:s r s)

(time:s r s)

(label:t r s t)

(tasteScore:i r i)

(surroundingScore:i r i)

(serviceScore:i r i)

(shopScore:i r i)

(hasCuisine:t r c t)

(cuisinePrice:i c i)

(npeople:t r i t)

## 5.2数据说明

标注了407条餐饮领域的问题，每一个样本为<自然语句，逻辑表达式>对。例如：

西单附近的餐厅

(lambda $0 e (and (restaurant:t $0) (zone:t $0 西单:s)))

五道口有什么比较好吃的日本料理

(lambda $0 e (and (restaurant:t $0) (zone:t $0 五道口:s) (label:t $0 日本料理:s) (> (tasteScore:i $0) 8:i)))

# **六、实验**

用407条标注数据进行十折交叉验证，结果如下：

Average precision : 0.9337721208773841

Average recall : 0.8800000000000001

Average F1 : 0.9057170522127217