

# Graph Model of the Descriptive Semantic Relationship for the Simple Semantic Units and Its Solution

Yuntong Liu\*, Shijia Chen

School of Computer and Information Engineering, Nanyang Institute of Technology, Nanyang Henan  
Email: [liuyt\\_liuyt@126.com](mailto:liuyt_liuyt@126.com)

Received: Feb. 14<sup>th</sup>, 2020; accepted: Mar. 2<sup>nd</sup>, 2020; published: Mar. 9<sup>th</sup>, 2020

---

## Abstract

In order to use semantics more effectively in natural language processing, a graph model of the descriptive semantic relationship for the simple semantic units was proposed and its solution was researched. Firstly the definition of the simple semantic unit was given and its characteristics were discussed. Then the graph of descriptive semantic relations could be created for a simple semantic unit, and the maximum spanning tree of the graph could be got by an approximate algorithm. And the simple semantic unit could be parsed and the word sense could be disambiguated. Finally experiments were finished and the results suggested that the method might be feasible.

## Keywords

Semantic Relevancy, Simple Semantic Units, Graph of the Descriptive Semantic Relations, The Variable of the Maximum Spanning Tree

---

# 简单语义单元的语义修饰关系图模型及其求解

刘运通\*, 陈世佳

南阳理工学院计算机与信息工程学院, 河南 南阳  
Email: [liuyt\\_liuyt@126.com](mailto:liuyt_liuyt@126.com)

收稿日期: 2020年2月14日; 录用日期: 2020年3月2日; 发布日期: 2020年3月9日

---

## 摘要

为了有效地使用语义信息来进行自然语言处理, 提出了一个简单语义单元的语义修饰关系图模型并对其

\*通讯作者。

求解方法进行了研究; 首先提出了简单语义单元的概念, 分析了简单语义单元的特点; 在生成其语义修饰关系图的基础上, 通过近似算法, 求取语义修饰关系完全图的最大生成树; 从而实现对简单语义单元的语法分析和语义消歧。实验表明, 该方法具有一定的可行性。

## 关键词

语义相关度, 简单语义单元, 语义修饰关系图, 最大生成树变量

Copyright © 2020 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

当前, 自然语言处理研究日益成为人工智能研究的热点, 研究的主要技术手段是基于统计学或基于神经网络机器学习这两种技术, 并取得了许多非常有价值的研究成就。但是, 这两种方法没有有效地利用语句中的语义信息, 没有语义分析计算环节, 影响了其分析处理的效果。

为了在自然语言处理中更加有效地利用语义分析, 取得更好的研究效果, 不少学者在原有技术的基础上, 通过引入语义分析计算的方法来进行自然语言处理的研究。陈谦研究了基于循环神经网络的自然语言语义表征方法, 利用了注意力机制来关注篇章的特定区域、内容, 同时设计了分散力机制去遍历篇章的不同内容, 从而更好地为摘要获取篇章的全局信息[1]; 彭敏等研究了一个基于双向 LSTM 语义强化的概率主题模型来实现文档宏观语义的嵌入表示, 采用文档 - 主题和词汇 - 词汇双语义强化机制以及 LSTM 来刻画参数推断过程中的吉布斯采样过程, 可以提高文本语义特征表达方面的有效性[2]; 贾玉祥等研究了采用神经网络技术的语义选择知识获取模型, 设计了引入预训练词向量的单隐层前馈网络和两层 maxout 网络, 在汉语和英语的伪消歧实验中神经网络模型取得了较好的效果[3]; Mikolov 等通过词向量模型工具 Word2vec 来描述语义信息, 具有较好的语言学特性, 可在很多应用和任务改善自然语言处理效果[4]; Mikilov 等人对词向量的研究非常具有开创性, 引起了围绕词向量训练的热潮。Word2vec 基于词频和上下文建立语言模型和训练词向量, 有研究者考虑用更复杂的方式来训练, 利用统计出的语言学规律, 给词向量赋予更多的语义信息[5]; Pennington 等提出 GloVe 模型, 综合运用词的上下文统计信息, 用共现矩阵来生成对数线性语言模型和词向量表示[6]; 孟凡擎等通过依存句法分析, 获取语料上下文知识, 构建上下文词汇语义消歧图, 进行依存句法分析, 完成歧义词的消歧处理[7]; 唐善成等研究了采用 Seq2Seq 模型的非受限词义消歧方法, 输入词上下文序列, 经过编码器编码得到潜在语义向量, 再经过解码器解码输出词义序列[8]; 夏小强等通过将 PV-DM 模型作为句向量训练基本模型, 添加关系信息知识约束条件, 能够学习到文本中词语之间的关系, 进一步整合语义关系约束信息, 能够更好地表示文本[9]。

虽然这些研究已经获得了许多重要的学术成果, 但迄今为止, 并没有形成一个系统、有效的自然语言处理语义模型, 其计算方法, 依然有很大的改进余地。为了在自然语言处理更有效地使用语义信息, 本文作者已经做了一些研究, 可以通过语义计算, 获取语句的语义修饰关系图[10], 在该文中, 提出了简单语义单元的概念, 但并没有得到很好地解决其语义修饰关系图的求解问题。为了解决这个问题, 本文对简单语义单元的语义修饰关系图的求解方法进行了进一步的研究, 设计近似算法, 可以求出语义修饰关系图的最大生成树; 实现了简单语义单元的语法、语义分析, 并实现语义消歧。

## 2. 理论基础

### 2.1. 简单语义单元的定义

假设有一个短文本  $L$ , 其词汇序列为  $L = W_m \dots W_0$ , 且  $W_i$  是  $L$  中的一个实词, 词汇  $W_{Gi}$  是词汇  $W_i$  的语义修饰目标, 函数  $match(W_i, W_{Gi})$  来表示  $W_i$  和  $W_{Gi}$  之间的语义相关度, 假如  $L$  满足如下的语义特征, 则称  $L$  为一个简单语义单元:

- i)  $\forall W_i ((W_i \in L) \cap (W_{Gi} \in L))$  (if  $i \neq 0$ );
- ii)  $L$  中不包含从句, 也不包含谓语;
- iii)  $L$  中不包含形式为  $W_{L1} \dots W_{L2} \dots$ , 或  $\dots W_{L3} \dots$  的虚词  $W_{L1}$ 、 $W_{L2}$ 、 $W_{L3}$ ; 这些虚词具有连接、分段作用。

### 2.2. 简单语义单元所具有的语义特征

通过简单语义单元的定义进行分析, 可以知道简单语义单元具有如下的语义特征:

- i) 除了最后一个词汇  $W_0$ , 任何词汇  $W_i$  的语义修饰目标  $W_{Gi}$ , 都在简单语义单元内部;
- ii) 简单语义单元内部, 没有复杂的内部语法结构需要处理;
- iii) 进行语法、语义分析时, 可以把简单语义单元作为一个整体来处理, 其内部语法、语义结构对语句其余部分的分析结果没有影响。

### 2.3. 简单语义单元在自然语言处理中的作用

在自然语言处理过程中, 可以把语句转化为多个简单语义单元, 再进行处理, 其方法如下:

#### 2.3.1. 分割语句并获取简单语义单元

假设语句  $C_S$  的语义、语义结构可以表示为:  $C_S \rightarrow LSLLOLVL$ , 并假定每个分段  $L$  是已经被分割成为“不包含从句”的程度, 就可以根据分段  $L$  内各部分语义修饰目标的不同, 将  $L$  划分为: 前置定语、后置定语、状语(分别用  $A_B$ 、 $A_A$ 、 $P_D$  表示)。就可以把  $L$  就转化为三个简单语义单元:  $A_B + S$ 、 $A_A + S$ 、 $P_D + V$  (见图 1)。

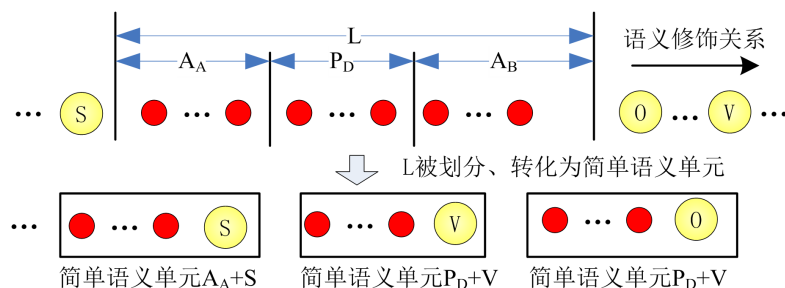


Figure 1. The method of turning sentences into the simple semantic units  
图 1. 把语句转化为简单语义单元的方法

#### 2.3.2. 含有分段、连接词的简单语义单元

经过 2.3.1 中方法的处理, 假如某个简单语义单元中还包含有形式为  $W_{L1} \dots W_{L2} \dots$ , 或  $\dots W_{L3} \dots$  的虚词。则可以用如下的方法将其转化为多个简单语义单元。

假设简单语义单元内包含了形式为  $W_{L1} \dots W_{L2} \dots$  的虚词; 那么, 简单语义单元内就被划分出两个语义块  $B_1$ 、 $B_2$ ; 语义块  $B_1$  的范围显而易见,  $B_2$  的后界限  $W_i$ , 可以通过式(1)来确定(见图 2):

$$\text{match}(W, W_i) = \max \{ \text{match}(W, W_1), \text{match}(W, W_2), \dots, \text{match}(W, W_i), \dots, \text{match}(W, W_p) \} \quad (1)$$

其中的  $W_1 \sim W_p$  与  $W$  的词性相同。

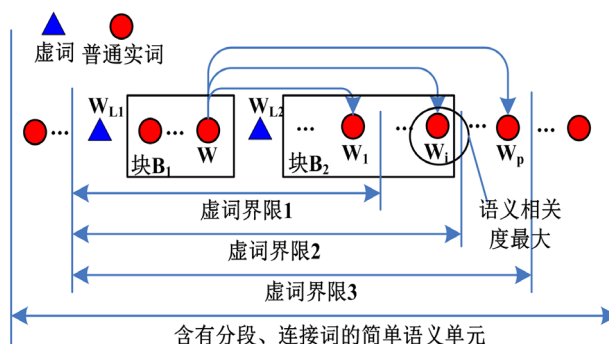


Figure 2. The processing method of the function words with segmented function  
图 2. 分段功能的虚词的处理方法

先将  $B_1$ 、 $B_2$  当作两个简单语义单元处理；然后，再把  $W_{L1}$   $B_1$   $W_{L2}$   $B_2$  的当作一个整体，用词汇  $W$  来替代即可。

## 2.4. 简单语义单元的语义修饰关系图模型

### 2.4.1. 简单语义单元的语义修饰关系图

自然语言语句中的词汇，大多数都是多义词，对简单语义单元来说，只要构造出它的语义修饰关系图，就求出了它的语法、语义分析结果，并实现了语义消歧。

假设简单语义单元的词汇序列为  $L = W_m \dots W_i \dots W \dots W_p \dots W_0$ 。L 有许多不同的语法分析方案  $\{A_n \dots A_t \dots A_0\}$ ，图 3 描述了它的第  $t$  语法分析方案  $A_t$ ，多义词  $W_i$  的第  $j$  个义项是  $W_i^j$ ，多义词  $W_p$  的第  $q$  个义项是  $W_p^q$ ，并且义项  $W_i^j$  语义修饰于义项  $W_p^q$ ；则在图中的义项  $W_i^j$  和义项  $W_p^q$  之间，画一条有向边，该边的权值等于  $\text{match}(W_i^j, W_p^q)$ 。

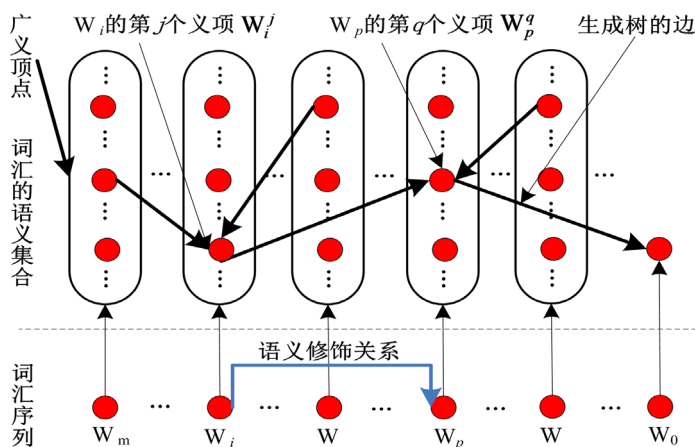


Figure 3. The graph of descriptive semantic relations for a simple semantic unit  
图 3. 简单语义单元的语义修饰关系图

在图 3 中，每个词的所有义项构成一个义项集合，每个义项都是其中的一个元素；每个词的义项集合都可以被看成一个“广义顶点”，每个义项都可以被看成一个“义项顶点”，则有如下特点：

- i) 最后的词汇  $W_0$  只具有一个义项, 不存在语义修饰目标, 其义项可以通过由语句级别处理过程确定;
- ii) 每个“广义顶点”只具有一个语义修饰目标, 集合的总出度是 1;
- iii) 每种语法分析方案, 本质上都是一个语义修饰关系图, 该图必然是“广义顶点完全图”的一颗“生成树”;
- iv) 在任意“义项集合”中, 其“生成树”的每条边, 必定关联于同一个“义项顶点”。

#### 2.4.2. 图语义修饰关系图最佳解的求解原理

自然语言语句中的词汇, 大多数都是多义词, 对简单语义单元来说, 只要构造出它的语义修饰关系图, 就求出了它的语法、语义分析结果, 并实现了语义消歧。

假设某个简单语义单元, 具有  $n$  种不同的语法分析方案  $\{A_n \cdots A_t \cdots A_0\}$ , 在它的第  $t$  语法分析方案  $A_t$  中,  $W_i^j$  是  $W_i$  的第  $j$  个义项,  $W_{G_i}^{G_j}$  是  $W_i^j$  的语义修饰目标义项,  $W_{G_i}^{G_i}$  是  $W_i^j$  的语义修饰目标义项; 用  $\text{match}(W_i^j, W_{G_i}^{G_j})$  表示它们之间的语义相关度, 则针对语法分析方案  $A_t$ , 总的语义相关度  $f_{A_t}$  可以用式(2)来计算:

$$f_{A_t} = \sum_{i=1}^m \text{match}(W_i^j, W_{G_i}^{G_j}) \quad (2)$$

其中,  $\text{match}(W_i^j, W_{G_i}^{G_j})$  用来表示义项  $W_i^j$ 、 $W_{G_i}^{G_j}$  之间的语义相关度, 针对一个词汇语义库(本文使用 hownet 词汇语义库), 可用式(3)计算。

$$\text{match}(W_i^j, W_{G_i}^{G_j}) = \max \{ \alpha * \text{sim}(W_i^j, W_{G_i}^{G_j}) + \beta * \text{rel}(W_i^j, W_{G_i}^{G_j}) \} \quad (3)$$

其中,  $\text{sim}(W_i^j, W_{G_i}^{G_j})$  表示义原之间的最大语义相似度,  $\text{rel}(W_i^j, W_{G_i}^{G_j})$  表示义原之间的最大语义关联度, 且系数  $\alpha + \beta = 1$ 。

**最佳解选择规则:** 在简单语义单元的  $n$  种不同的语法、语义分析方案  $\{A_n \cdots A_t \cdots A_0\}$  中, 满足式(4)中条件的语法、语义分析方案  $A_t$  就是最佳方案:

$$A_t = \text{argmax}(f_{A_t}) \quad (4)$$

最佳语法分析方案的语义修饰关系图是简单语义单元“广义顶点完全图”的“最大生成树”。

值得注意的是: 代词、副词、量词、数词被按照实词计算, 虚词不参与计算。

### 3. 语义修饰关系图模型的近似求解

假定简单语义单元具有  $m$  个词汇, 一个词汇平均具有  $t$  个义项。假如采用穷举法来求取最优解, 其计算复杂度将是  $O(m^2 * t^m)$ , 属于指数级别时间复杂度, 实际上不可行。

为此, 我们提出了一种近似算法, 可以降低求解的计算复杂度, 较快地求出近似解。

#### 3.1. 最大生成树变量

采用动态规划方法, 可以求出模型的近似解。先定义一个“最大生成树变量”(见图 4), 其含义为: 针对词汇  $W_i$ , 其第  $j$  个义项  $W_i^j$  有多个生成树, 其中各边权值之和最大的生成树, 用  $M_T(W_i^j)$  表示。

#### 3.2. 求最大生成树的近似算法

计算过程中的所涉及的两个变量:

集合  $V = \{ \text{已经计算出最大的生成树 } M_T \text{ 的“广义顶点”} \}$

目标词汇  $W_i$ : 即将被加入  $V$  的下一个词汇;

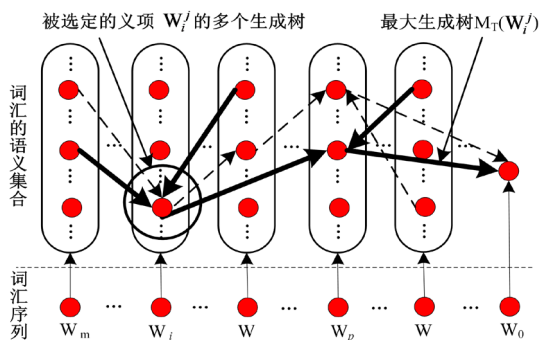


Figure 4. The variable of the maximum spanning tree for a simple semantic unit  
图 4. 简单语义单元的最大生成树变量

动态规划算法(算法 1):

步骤 1: 设定  $V$  的初始值

设置  $V = \{W_1, W_0\}$ , 则很容易算出:

$$M_T(W_0^i) = \max \{(W_0^i, W_1^j), 0 \leq j \leq \text{Num}(W_1)\};$$

$$M_T(W_1^j) = \max \{(W_0^i, W_1^j), 0 \leq i \leq \text{Num}(W_0)\}.$$

其中 Num 是词汇的义项个数,  $(W_0^i, W_1^j)$  表示义项  $W_0^i$  和  $W_1^j$  之间的边。

步骤 2: 计算目标词汇  $W_i$  的最大生成树变量

针对  $W_i$ , 其每个义项  $W_i^j$  与  $V$  中的任意义项  $W_p^q$  之间, 都可以计算出一个语义相关度, 按照式(5), 可以计算出  $W_i^j$  的最大生成树变量(见图 5)。

$$M_T(W_i^j) = \max \{M_T(W_p^q) + (W_i^j, W_p^q)\} \quad (5)$$

$(W_i^j, W_p^q)$  表示义项  $W_i^j$  和  $W_p^q$  之间的边。

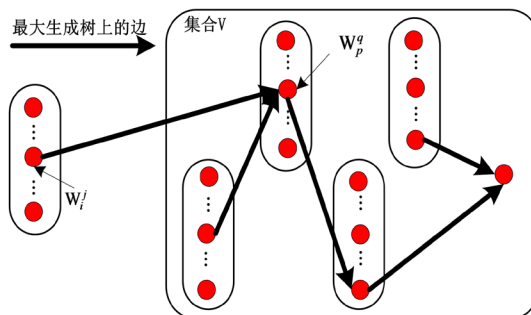


Figure 5. Computing the variable of the maximum spanning tree for semantic item  $W_i^j$

图 5. 计算义项  $W_i^j$  的最大生成树变量

步骤 3: 对所有的  $(W_i^j, W_p^q)$  进行逆排序

在步骤 2 的计算过程中, 可以计算出所有的  $(W_i^j, W_p^q)$  的值, 按由大到小的顺序, 将其保存到线性表 List 中。

步骤 4: 更新  $V$  中义项  $W_p^q$  的最大生成树变量

如图 6 所示, 已知  $V$  中义项  $W_p^q$  的最大生成树, 可定义变化函数  $f$ :



$$f(W_r^{s0}, W_r^{s1}) = \sum((W_r^{s1}, W) - (W_r^{s0}, W)) \tag{6}$$

其含义为: 假设把最大生成树进行一个义项改动, 由义项  $W_r^{s0}$  改为义项  $W_r^{s1}$ , 其权值之和的变化量。假定边  $(W_i^j, W_{r1}^{s1})$  所关联的顶点为  $W_{r1}$ , 把最大生成树上的义项  $W_{r1}^{s0}$  改为义项  $W_{r1}^{s1}$ ; 则其变化函数  $f(W_{r1}^{s0}, W_{r1}^{s1})$  的值必然小于 0。

假如按照由大到小的顺序, 处理 List 中的所有边  $(W_i^j, W_{r1}^{s1})$ , 有以下两种情况:

- i)  $W_{r1}^{s1}$  位于最大生成树上, 则 List 中后续的任何边都不可能是最大生成树的边, 因此就无需处理了。
- ii)  $W_{r1}^{s1}$  不位于最大生成树上, 则按照下属算法 2 求取义项  $W_p^q$  的最大生成树。

**算法 2:**

For all  $(W_i^j, W_{r1}^{s1})$  in List

{

    If ( $W_{r1}^{s1}$  不在最大生成树上)

    {

        计算变化函数  $f(W_{r1}^{s0}, W_{r1}^{s1})$  的值;

    (A) 找到  $W_{r1}$  所指的下一个顶点 ( $W_{r2}$ )

    {

        针对  $W_{r2}$  的所有义项  $W_{r2}^{s1}$ , 计算变化函数  $f(W_{r2}^{s0}, W_{r2}^{s1})$  的值;

        选取能够使得  $f(W_{r2}^{s0}, W_{r2}^{s1})$  最大的义项  $W_{r2}^{s1}$ ;

        将最大生成树上的义项由  $W_{r2}^{s0}$  修改为  $W_{r2}^{s1}$ ;

    }

    返回(A), 循环处理至最后一个顶点;

    按下式计算总变化量:

$$(W_i^j, W_{r1}^{s1}) + \sum_{r=1}^N f(W_{r1}^{s0}, W_{r1}^{s1}), \text{ 即对所有的变化函数求和, 并加上边 } (W_i^j, W_{r1}^{s1}) \text{ 的值。}$$

    }

}

选择能够使总变化量最大的边  $(W_i^j, W_{r1}^{s1})$ , 将其修改为最大生成树的边。

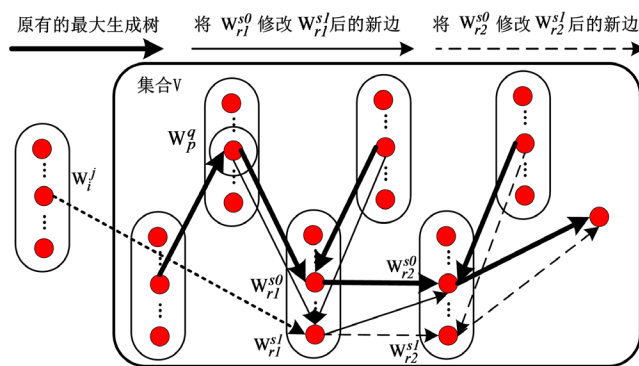


Figure 6. Updating the variable of the maximum spanning tree for semantic item  $W_i^j$

图 6. 更新义项  $W_i^j$  的最大生成树变量

**步骤 5: 把词汇  $W_i$  加入集合  $V$** 

针对  $V$  中的所有义项, 更新其最大生成树变量, 完成后把  $W_i$  加入集合  $V$ 。

**步骤 6: 循环**

返回步骤 2, 循环处理, 当所有顶点都加入集合  $V$  后, 选出词汇  $W_m$  的最大生成树, 就是所求的近似解。

**3.3. 计算结果分析**

在算法 1 的步骤 4 中, 使用了贪心法, 只选择能够使得  $f(W_{r_2}^{s_0}, W_{r_2}^{s_1})$  最大的义项  $W_{r_2}^{s_1}$  进行计算, 而没有对所有  $f(W_{r_2}^{s_0}, W_{r_2}^{s_1}) > 0$  的所有义项进行穷举; 故此所求的解不是“理论最优解”, 只是近似解, 其时间复杂度约为  $O((m*t)^2)$ 。

**4. 实验结果及分析**

实验中, 以 hownet 作为语义计算的依据。在人民日报 2014 语料库中选取了 1200 个语句进行求解实验, 先将它们转化为 4729 个简单语义单元, 通过算法进行求解, 并与正确结果进行对比。实验环境为: windows 10, i5 9400F 六核, 主频 2.9 GHz, 32 G 内存。

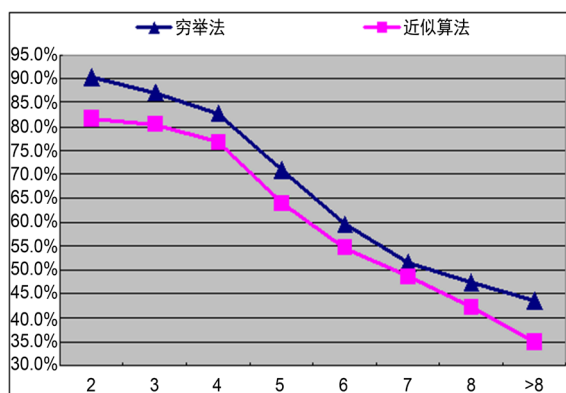
实验情况见表 1、图 7、图 8, 其中正确率标准有两个:

- 1) 语义消歧正确率大于阈值 0.5 (义项选择正确就是语义消歧正确);
- 2) 语法分析正确率大于阈值 0.8 (义项的语义修饰目标正确就是语法分析正确)。

**Table 1.** The resulting data of experiment

**表 1.** 实验数据表

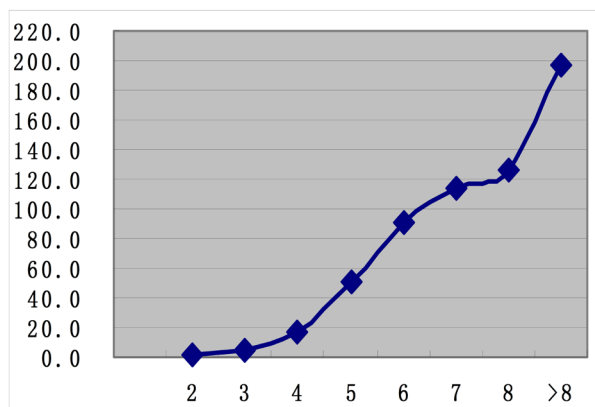
长度	个数	平均歧义个数	穷举法计算时间(ms)	穷举法正确率	近似算法计算时间(ms)	近似算法正确率
2	697	3.17	1848	87.7%	1838	79.2%
3	887	3.86	10,977	83.6%	2820	79.8%
4	1153	3.90	97,017	79.6%	5713	74.4%
5	769	3.59	370,284	68.1%	7303	60.8%
6	447	2.95	695,010	58.9%	7420	53.2%
7	349	2.47	815,967	49.9%	7289	47.6%
8	194	2.19	993,451	46.9%	7871	40.0%
>8	233	1.81	1,749,055	41.8%	8707	34.5%



**Figure 7.** The comparison for the correct ratios

**图 7.** 正确率对比图





**Figure 8.** The exhaustive time/the approximate algorithm time  
**图 8.** 穷举法时间/近似算法时间

从实验结果可以看出:

- 1) 两种算法均可进行语法分析和语义消歧;
- 2) 穷举法的正确率略高于近似算法;
- 3) 正确率随着简单语义单元长度的增加有逐渐下降的趋势;
- 4) 当简单语义单元长度增加时, 近似算法所需时间, 在可接受的范围内, 属于实践可行的算法, 而穷举法计算所需的时间急剧增加, 实践上不可行。

## 5. 结语

本文对简单语义单元的语义修饰关系图模型及其求解方法进行了研究; 通过把语句转化为简单语义单元, 生成其语义修饰关系图, 通过近似算法对模型进行求解; 可以实现对简单语义单元的语法分析和语义消歧; 并通过实验, 验证了本方法的计算效果。由于问题自身难度所限, 本文没有能够找具有“非指数级别计算复杂度”的求取最佳理论解的方法。另外, 在语义计算时, 由于所依据的 hownet 词汇语义库的准确度、细致程度等有很多欠缺和不足, 使得语义相关度计算结果不够准确; 这也是计算效果不够理想的关键因素。这些问题, 将是我们下一步的研究的目标。

## 基金项目

河南省教育厅科学技术研究重点项目(18A520002)。

## 参考文献

- [1] 陈谦. 基于神经网络的自然语言语义表征方法研究[D]: [博士学位论文]. 合肥: 中国科学技术大学, 2018.
- [2] 彭敏, 杨绍雄, 朱佳晖. 基于双向 LSTM 语义强化的主题建模[J]. 中文信息学报, 2018, 32(4): 40-49.
- [3] 贾玉祥, 许鸿飞, 咎红英. 基于神经网络的语义选择限制知识自动获取[J]. 中文信息学报, 2017, 31(1): 155-161.
- [4] Mikolov, T., Chen, K., Corrado, G., *et al.* (2013) Efficient Estimation of Word Representations in Vector Space. [http://cn.bing.com/academic/profile?id=6b535fdb1fad7c57b8421d3a81b00af5&encoded=0&v=paper\\_preview&mkt=zh-cn](http://cn.bing.com/academic/profile?id=6b535fdb1fad7c57b8421d3a81b00af5&encoded=0&v=paper_preview&mkt=zh-cn)
- [5] Wan, S., Lan, Y., Guo, J., *et al.* (2016) A Deep Architecture for Semantic Matching with Multiple Positional Sentence Representations. [https://www.researchgate.net/publication/285271115\\_A\\_Deep\\_Architecture\\_for\\_Semantic\\_Matching\\_with\\_Multiple\\_Positional\\_Sentence\\_Representations](https://www.researchgate.net/publication/285271115_A_Deep_Architecture_for_Semantic_Matching_with_Multiple_Positional_Sentence_Representations)
- [6] Pennington, J., Socher, R. and Manning, C. (2014) Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, 25-29 October 2014, 1532-1543. <https://doi.org/10.3115/v1/D14-1162>

- [7] 孟凡擎, 鹿文鹏, 张旭, 等. 基于 HowNet 的图模型词义消歧方法[J]. 齐鲁工业大学学报, 2018, 32(6): 69-76.
- [8] 唐善成, 马付玉, 张镛月, 等. 采用 Seq2Seq 模型的非受限词义消歧方法[J]. 西北大学学报(自然科学版), 2019, 49(3): 351-355.
- [9] 夏小强, 邵堃. 基于语义关系约束和词语关系信息的句向量研究[J]. 计算机应用研究, 2019, 36(7): 2023-2026.
- [10] Liu, Y.T. and Sun, H. (2015) Word Sense Disambiguation for Chinese Based on Semantics Calculation. *Mathematical Problems in Engineering*, **2015**, Article ID: 235096. <https://doi.org/10.1155/2015/235096>