# 基于维基百科的领域概念语义知识库的自动构建方法

| | |
|---|---|
| 作者 | 张巧燕，林民，张树钧 |
| 机构 | 内蒙古师范大学 计算机与信息工程学院 |
| | |
|---|---|
| 发表期刊 | 《计算机应用研究》 |
| 预排期卷 | 2018 年第 35 卷第 1 期 |
| 访问地址 | http://www.arocmag.com/article/02-2018-01-028.html |
| 发布日期 | 2017-01-19 14:55:41 |
| 引用格式 | 张巧燕，林民，张树钧. 基于维基百科的领域概念语义知识库的自动构建方法[J/OL]. [2017-01-19]. http://www.arocmag.com/article/02-2018-01-028.html. |
| | |
|---|---|
| 摘要 | 针对为检索服务的语义知识库存在的内容不全面和不准确的问题，提出一种基于维基百科的软件工程领域概念语义知识库的构建方法；首先，以 SWEBOK V3 概念为标准，从维基百科提取概念的解释文本，并抽取其关键词表示概念的语义；其次，通过概念在维基百科中的层次关系、概念与其他概念的解释文本关键词的关系构成概念语义知识库；接着，LDA 主题模型分别和 TF-IDF 算法、TextRank 算法相结合的两种方法抽取关键词；最后，对构建好的概念语义知识库用随机游走算法计算概念间的语义相似度；将实验结果进行人工对比后发现，… |
| | |
|---|---|
| 关键词 | 维基百科，语义知识库，关键词抽取，语义相似度计算，随机游走 |
| 中图分类号 | TP391.1 |
| 基金项目 | 国家自然科学基金资助项目（61562068）；内蒙古自然科学基金资助项目（2015MS0629，2014MS0617）；内蒙古民委蒙古文信息化专项扶持子项目（MW-2014-MGYWXXH-01）；内蒙古自治区高等学校科学研究项目（NJZY028）；内蒙古师范大学引进人才科研启动经费项目（2014YJRC036）；内蒙古师范大学校级基金资助项目（2015YBXM002） |

# 基于维基百科的领域概念语义知识库的自动构建方法 *

张巧燕，林　民，张树钧

(内蒙古师范大学 计算机与信息工程学院，呼和浩特 010022)

**摘　要：**针对为检索服务的语义知识库存在的内容不全面和不准确的问题，提出一种基于维基百科的软件工程领域概念语义知识库的构建方法；首先，以 SWEBOK V3 概念为标准，从维基百科提取概念的解释文本，并抽取其关键词表示概念的语义；其次，通过概念在维基百科中的层次关系、概念与其他概念的解释文本关键词的关系构成概念语义知识库；接着，LDA 主题模型分别和 TF-IDF 算法、TextRank 算法相结合的两种方法抽取关键词；最后，对构建好的概念语义知识库用随机游走算法计算概念间的语义相似度；将实验结果进行人工对比后发现，本方法构建的语义知识库语义相似度准确率能够达到 84%以上；充分验证了所提方法的有效性。

**关键词：**维基百科；语义知识库；关键词抽取；语义相似度计算；随机游走

**中图分类号：**TP391.1

## Research on automatic construction of domain concepts on Wikipedia semantic knowledge base

Zhang Qiaoyan, Lin Min, Zhang Shujun

*(College of Computer & Information Engineering, Inner Mongolia Normal University, Hohhot 010022, China)*

**Abstract:** The problem of incomplete and inaccurate content for the retrieval of semantic knowledge base exists,. propose a method of constructing the concept semantic knowledge base in the field of software engineering based on Wikipedia. First, Taking the concept of SWEBOK V3 as the standard, the interpretation of the concept is extracted from Wikipedia and the keywords are extracted to represent the semantic meaning of the concept. Second,Through the concept of hierarchical relationships in Wikipedia, the concept of the relationship with other concepts of the interpretation of the text of the concept of semantic knowledge base. Then, The LDA topic model is combined with the two methods that are called TF-IDF algorithm and TextRank algorithm respectively serve the keywords extraction. Finally,The semantic similarity between concepts is calculated by the random walk algorithm for the construction of the concept semantic knowledge base. The experimental results were compared with the manual work , The semantic similarity of knowledge base constructed by this method can reach more than 84%, It effectiveness of the proposed method is verified.

**Key Words:** wikipedia; semantic knowledge base; keywords extraction; semantic similarity computation; random walk

## 0　引言

在现今大数据时代的环境下,用户希望检索到的内容能更加符合自己的需求,为了使得检索的内容更加全面和准确,构建为检索服务的语义知识库是很必要的，由领域专家手工创建的语义知识库，具有较高的质量和权威性，但更新和调整不易，内容具有静态性和有限性。并且人工构建语义知识库的过程需要投入大量的人力和时间，难以满足大规模网络文本信息检索的需要。

以维基百科为代表的网络百科知识库具有知识覆盖范围广，描述详细全面可靠、内容更新迅速、文档质量上和数量上都有其他资源无法比拟的优势，为构建面向检索的语义知识库提供了良好的基础数据资源。随着以 MOOC 为代表的网络化教学模式的兴起，在 MOOC 平台上需要检索各类学习资源，为了更好地满足学习者对学习资源的细粒度、高质量的检索要求，需要构建面向教学的领域语义知识库支持 MOOC 系统对学习资源按语义进行检索。因此，构建面向教学的易于更新内容的领域概念语义知识库具有重要意义。

---

**基金项目：**国家自然科学基金资助项目（61562068）；内蒙古自然科学基金资助项目（2015MS0629，2014MS0617）；内蒙古民委蒙古文信息化专项扶持子项目（MW-2014-MGYWXXH-01）；内蒙古自治区高等学校科学研究项目（NJZY028）；内蒙古师范大学引进人才科研启动经费项目（2014YJRC036）；内蒙古师范大学校级基金资助项目（2015YBXM002）

**作者简介：**张巧燕（1990-），女，内蒙古乌兰察布人，硕士研究生，主要研究方向为自然语言处理；林民（1969-），男，内蒙古呼和浩特人，教授，博士，主要研究方向为自然语言处理、人工智能；张树钧（1979-），男，内蒙古呼和浩特人，实验师，硕士，主要研究方向为自然语言处理.

## 1 相关工作

### 1.1 语义知识库构建

目前已经有很多自动构建的语义知识库，苏小康[1]提出对于每个概念从他自身的解释文本中抽取关键词并计算关键词对概念的贡献度形成语义指纹描述概念的语义，但概念之间并没有进一步的层次关系。陈千[2]对文本流中的多粒度主题进行建模,提出了基于语义层次树的主题语义描述方法，通过计算概念间的 sim 散度对主题语义进行了细粒度的表示。张琪[3]提出了四层树状层次结构的主题本体树，自顶向下有文档层、根主题层、叶子主题层、概念层，从根主题层到叶子主题层，每一层逐层都可以用一个 dirichlet 分布来表示；伍成志[4]构建了概念之间的网状结构的语义关系，但是对于概念本身并没有选取关键词来描述它的语义。万亿[5]用图模型来表示概念之间的语义关系。张涛[6]构建了基于维基概念的知识图谱。刘巧玲[7]提出了基于维基百科的语义搜索，利用维基百科本身的特性挖掘概念间隐含语义关系。知识图谱本质上是一种语义网络，相对于传统的本体和语义网络而言，实体覆盖率更高，语义关系也更加复杂而全面。

### 1.2 关键词抽取

关键词抽取方法可以抽取出概念解释文本最能表示概念语义的关键词，常用的关键词抽取方法有以下三种：Mihalcea 等提出了一种基于图的文本词语重要性排序算法 TextRank[8]，TextRank 认为一个词的重要程度取决于与它相邻的词的重要程度，利用 PageRank 算法计算词的重要性，并按照词的重要性排序；通过 TextRank 抽取出的概念解释文本关键词权重值区分度较好。隐含狄利克雷分布(Latent Dirichlet llocation,LDA) 模型是一种利用文档主题分布抽取关键词的主题模型[9],LDA 将“词项-文档”分布矩阵分解成“词项-主题”“主题-文档”矩阵，再从文档的主题中识别关键词，LDA 在抽取关键词时考虑了文档的语义信息；但是关键词的权重值相互之间的区分度不是很明显。TF-IDF 是一种统计方法，用以评估词项对于一个文档集或一个语料库中的其中一个文档的重要程度；字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。通过 TF-IDF 抽取出的概念解释文本关键词权重值区分度亦较好。所以本文将 LDA 主题模型和 TF-IDF 相结合、LDA 主题模型和 TextRank 相结合的结果进行实验对比，结果较好的作为最终的关键词抽取结果。

### 1.3 概念语义相似度计算

常用的概念语义相似度计算方法有:基于距离的相似度计算方法、基于信息内容的相似度计算方法、基于属性的语义相似度计算方法[10-12]；基于语义距离的方法根据概念层次树中的位置来计算概念语义相似度。基于信息内容的方法以信息熵计算为基础，不依赖于层次结构的词典，只要有概率模型存在，这种方法就可以应用。Tversky 模型是典型的特征匹配方法，通过特征匹配过程来计算概念语义相似度；相似度计算不仅由两个概念的相同属性决定，而且由它们的不同属性决定。在此基础上，安建成[13]在传统语义相似度计算方法的基础上，综合考虑了边的深度、密度、强度及两个概念的语义重合度、层次差等主要影响因素，提出了一种基于语义树的概念相似度计算方法。刘宏哲[14][15]等人提出了一个基于相关概念节点密度的概念向量模型来计算概念间语义相似度和相关度（简称 RNCVM 方法）。何夏燕[16]将词语的释义项转换为内涵概念图的形式，然后计算两个内涵概念图之间的相似程度，从而求得词语语义相似度的值。张琪[3]提出了四层树状层次结构的主题本体树，从根主题层到叶子主题层，每一层逐层都可以用一个 dirichlet 分布来表示；由于主题是采用分布来计算的,采用信息论的 KL 散度计算主题之间的相似度，但是从主题 $p$ 到主题 $q$ 的相似度并不等于从主题 $q$ 到主题 $p$ 的相似度，文献[3]对 KL 散度进行了修正，得到一个对称的测度 *sym-KL*,使得从主题 $p$ 到主题 $q$ 的相似度等于从主题 $q$ 到主题 $p$ 的相似度。

基于以上的相关工作，本文提出了一种基于维基百科的领域概念语义知识库自动构建方法。本文首先选取 SWEBOK V3 软件工程领域核心概念，用 JWPL 从维基百科提取概念的解释文本；对每个概念的解释文本抽取关键词，依次按权重值从高到低选 5 个关键词表示概念本身的语义；根据以下规则构建有向概念图：1.概念本身在维基百科中的上下位关系；2.概念本身在维基百科中的下上位关系；3.概念 *A* 在概念 *B* 的解释文本关键词中出现；4.概念 *A* 和概念 *B* 的解释文本中有相同的关键词；根据构建好的图，利用图的随机游走算法来确定概念之间的相似度。

本文的组织结构如下：第二节介绍软件工程领域概念语义知识库的构建流程；第三节介绍概念语义相似度的计算；第四节介绍实验；第五节总结全文并对未来工作进行展望。

## 2 概念语义知识库构建流程

本文首先以 SWEBOK V3 概念为标准，SWEBOK 由 IEEE Computer Society 制定的软件工程知识体系 SWEBOK，全名是 Guide to the Software Engineering Body of Knowledge，定义了软件工程学科的内涵，自发布至今，有效地推动了软件工程专业教育的发展；SWEBOK 第 3 版[17](SWEBOK V3）增加到了 15 个知识域，共有 102 个知识点，扩充了近十年来的软件工程研究与实践的最新成果。本文从 SWEBOK V3 中选取软件工程领域概念集，对概念集里的每个概念用 JWPL(Java Wikipedia Library)[18]从维基百科提取其解释文本，对概念解释文本进行分词预处理,分词包括去停用词和加领域词典,并对其解释文本抽取关键词表示概念本身的语义；根据制定好的规则构建概念图。

构建流程如图 1 所示:

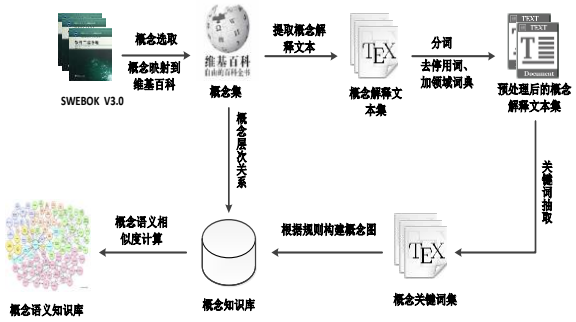


图 1　语义知识库构建流程

## 2.1　概念解释文本关键词抽取

对预处理后的概念解释文本分别用 TextRank 算法、LDA 主题模型、TF-IDF 算法进行关键词抽取。

### 2.1.1 TextRank 算法抽取关键词

TextRank 是一种基于图的排序算法，通过词语在文档中的位置关系， 采用投票机制构建有向图 $G=(V,E)$，其中：$V$ 是节点集合，由词语组成；$V$ 为边集合， $E\subseteq V*V$，对于句子中的每个词语 $v_i$ 与 $v_j$，若 $j$-$i$<常数（窗口大小），则 $\langle v_i,v_j\rangle\in E$。循环计算各词语的重要性直到收敛，其迭代计算公式如下：

$$WS(V_i)=(1-d)+d*\sum_{v_j\in In(V_i)}\frac{w_{ji}}{\sum_{v_k\in Out(V_j)}w_{jk}}*WS(v_j) \quad (1)$$

其中：$In(v_i)$表示节点 $v_i$ 的入度，$Out(v_j)$表示节点 $v_i$ 的出度，$w_{ji}$ 表示节点 $v_i$ 与 $v_j$ 的“关联强度”。$d$ 为阻尼系数，一般设为 0.85。

### 2.1.2 LDA 主题模型抽取关键词

LDA (Latent Dirichlet Allocation，LDA)是一种利用文档主题分布抽取关键词的主题模型，它将“词项-文档”分布矩阵分解成“词项-主题 ”和“主题-文档”矩阵， 再从文档的主题中识别关键词，LDA 主题模型抽取关键词：设置主题个数 K = 1 时，只有一篇文档，这时 LDA 模型相当于是一种有监督的主题模型；计算公式如下：

$$p(Z_i=k\mid Z_{-i},d,w)\propto\frac{n_{k,-i}^{(t)}+b_t}{\sum_{t=1}^{V}(n_{k,-i}^{(t)}+b_t)}*\frac{n_{m,-i}^{(k)}+\alpha_k}{\sum_{k=1}^{K}(n_{m,-i}^{(k)}+\alpha_k)} \quad (2)$$

其中 $n_{k,-i}^{(t)}$ 表示去除词项 $w_i$ 后，文本集中被分配主题 $k$ 的词项 $t$ 的数量；$\sum_{t=1}^{V}n_{k,-i}^{(t)}$ 表示去除词项 $w_i$ 后，文本集中被分配主题 k 的词项总数量；$n_{m,-i}^{(k)}$ 表示去除词项 $w_i$ 后，文本 $m$ 中被分配主题 $k$ 的单词数量；$\sum_{k=1}^{K}n_{m,-i}^{(k)}$ 表示去除词项 $w_i$ 后，文本 $m$ 中的单词数量；$\frac{n_{k,-i}^{(t)}+b_t}{\sum_{t=1}^{V}(n_{k,-i}^{(t)}+b_t)}$ 表示去除单词 $w_i$，其他的单词 $w_i$ 属于主题 $k$ 的概率；$\frac{n_{m,-i}^{(k)}+a_k}{\sum_{k=1}^{K}(n_{m,-i}^{(k)}+a_k)}$ 表示去除单词 $w_i$，主题 $m$ 属于当前文档的概率。

### 2.1.3 TF-IDF 算法抽取关键词

在给定的文档中，词频（TF）是指某一具体给定的词语在这个文档中出现的次数。对于在某一特定文档里的词语 $t_i$，其词频可以表示为：

$$tf_{i,j}=\frac{m_{i,j}}{\sum_k m_{k,j}} \quad (3)$$

其中，$m_{i,j}$ 是该词在文档 $d_j$ 中出现的次数，分母 $\sum_k m_{k,j}$，$j$ 是在文档 $d_j$ 中所有字词出现次数之和。为了防止它偏向长文件，一般需要进行归一化。

逆向文件频率(IDF）即对一个词语普遍重要性的度量。某一特定词语的 IDF 值，可由总文件数除以包含该词语的文件的数目，再将二者的商取对数得到，公式如下 ：

$$idf_i=log\frac{|M|}{|\{j:t_i\in d_j\}|} \quad (4)$$

其中，$|M|$表示的是语料库中总的文档数目，分母$|\{j:t_i\in d_j\}|$表示包含词语 $t_i$ 的文件数目。由式（3）和式（4）可得到单词的权重公式为：

$$tfidf_{i,j}=tf_{i,j}*idf_i \quad (5)$$

用三种方法对同一概念解释文本抽取 10 个关键词表示概念语义，表中小数表示每种方法中关键词对应的权重值，结果如表 1 所示。

表 1 LDA、TFIDF 和 TextRank 抽取概念关键词结果对比

| **软件生命周期模型**：软件生命周期同任何事物一样，一个软件产品或软件系统也要经历孕育、诞生、成长、成熟、衰亡等阶段，一般称为软件生命周期。 | | |
|---|---|---|
| **LDA 主题模型** | **TF-IDF 算法** | **TextRank 算法** |
| 软件生命周期 0.1373 | 软件生命周期 0.0944 | 周期 100 |
| 事物 0.0719 | 孕育 0.0852 | 孕育 6 |
| 软件产品 0.0719 | 成熟 0.0852 | 成熟 6 |
| 软件系统 0.0719 | 成长 0.0852 | 成长 6 |
| 经历 0.0719 | 经历 0.0852 | 经历 6 |
| 孕育 0.0719 | 衰亡 0.0852 | 衰亡 6 |
| 诞生 0.0719 | 生存 0.0777 | 诞生 6 |
| 成长 0.0719 | 诞生 0.0777 | 软件生命周期 4 |
| 成熟 0.0719 | 周期 0.0772 | 软件系统 4 |
| 衰亡 0.0719 | 软件产品 0.0772 | 软件 2 |

表 1 结果显示：LDA 主题模型对概念的解释文本抽取出的

关键词比较符合人的判断，但是关键词的权重值相互之间的区分度不是很明显,如果想要抽取权重值最高的前 5 个关键词表示概念的语义,这时使用 LDA 主题模型将会存在很多不便,TF-IDF 和 TextRank 相对于 LDA 主题模型而言关键词的权重值区分度比较好；所以将 LDA 主题模型分别和 TF-IDF 算法、TextRank 算法相结合对两者的结果进行比较。

2.1.4 LDA 主题模型分别与 TF-IDF 算法、TextRank 算法相结合

具体结合方法如下:对于同一篇概念解释文本,本文利用 LDA 主题模型抽取关键词,用 TF-IDF 算法的计算公式计算关键词在解释文本中的权重值,LDA 主题模型与 TextRank 算法相结合亦如此；对每个概念解释本文选取权重值最高的 5 个关键词来支持概念的语义；相结合之后的结果如表 2 所示。

表 2 结合结果对比

| LDA 与 TF-IDF 相结合 | LDA 与 TextRank 相结合 |
|---|---|
| 软件生命周期 0.0980 | 孕育　6 |
| 孕育　0.0885 | 成熟　6 |
| 成熟　0.0885 | 成长　6 |
| 成长　0.0885 | 经历　6 |
| 经历　0.0885 | 衰亡　6 |
| 衰亡　0.0885 | 诞生　6 |
| 诞生　0.0807 | 软件生命周期　4 |
| 软件产品　0.0750 | 软件系统　4 |
| 事物　0.0567 | 软件产品　1 |
| 软件系统:　0.0567 | 事物　0 |

由于 LDA 主题模型在抽取关键词时考虑了文档的语义信息，TF-IDF 算法的区分度相对较好，根据表 2 显示结果得出：LDA 主题模型与 TF-IDF 相结合的方法抽取出的解释文本关键词在权重值和区分度上能更好的表示概念语义；所以将 LDA 主题模型与 TF-IDF 算法相结合作为本文抽取概念解释文本关键词的方法。

## 2.2 概念语义知识库构建

维基百科[19]是目前全世界最大的多语种开放式的在线百科全书，允许用户编辑，内容丰富且跟新速度快；维基百科的分类页面是一种具有层次结构的图[20]，能反映出概念之间的层次关系，它包含该分类的所有子分类、父分类和属于该分类的概念。根据 2.1 抽取好的解释文本关键词,本文结合概念在维基百科原有的层次关系并且新加入每个概念与其他解释文本关键词之间的关系构建概念语义图，具体构图规则如下：

a) 概念在维基百科中存在上位到下位关系，边的权重值初始化为 0.9。

b)概念在维基百科中存在下位到上位关系，边的权重值初始化为 1。

c) 概念 A 在概念 B 的解释文本关键词中出现，把概念 A 在概念 B 中的权重值赋值给概念 A 到概念 B 的边。

d) 概念 A 和概念 B 的解释文本有相同的关键词 C，概念 A 到概念 B 的边的权重值赋值为$(w_1+w_2)/2$。

其中，$w_1$ 表示关键词 *C* 在概念 *A* 的解释文本中作为关键词的权重值，$w_2$ 表示关键词 *C* 在概念 *B* 的解释文本中作为关键词的权重值。

考虑到下位概念在语义上更能具体表示上位概念，所以把上位到下位概念的边的权重值定义为 0.9，把下位到上位概念边的权重值定义为 1。

加边的优先级由高到底依次是 1)→2)→3)→4)。

构建有向图$G=(V,E)$),其中 *V* 代表概念集合，*E* 代表边集合，$E \subseteq V*V$ 。

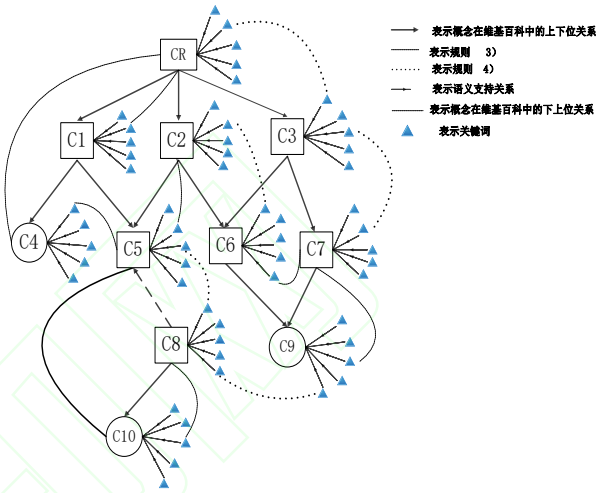


图 2　概念语义知识库结构图

根据以上四条规则构建的概念知识图如图2, 其中*CR*表示根概念节点, □表示此概念有下位概念, ⬭表示叶子节点。

# 3　概念间语义相似度计算

本文利用图的随机游走算法[21]计算概念间的语义相似度，图的随机游走算法用来捕捉图中两个节点之间的相似度，根据 2.1 的构图规则，直接的上下位关系和有间接的链接关系的两个节点之间在语义上有某种相似度，如图 2，*CR* 和 *C3* 有上下位关系，*C3* 和 *C7* 有上下位关系，则 *CR* 和 *C7* 在语义上有某种相似度，只不过关联强度相对 *CR* 和 *C3*、*C3* 和 *C7* 要弱一些。

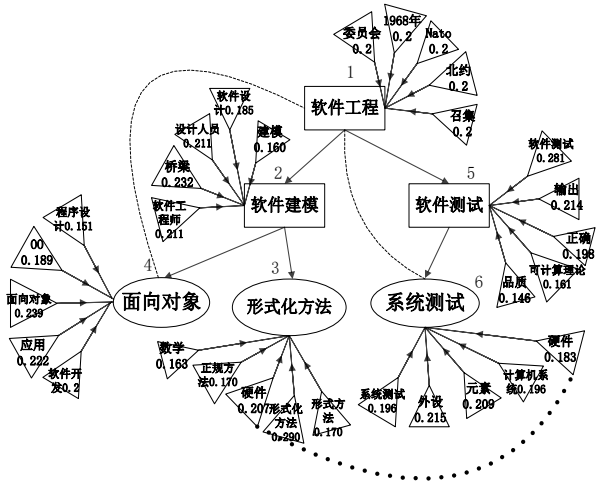


图 3　部分概念间的语义关系

本文用 2.1 构建好的有向图映射到矩阵，构成一个 *m***m* 的概率转移矩阵 *P*，*m* 表示图中概念节点的数量；对矩阵 *P* 的每

一行元素进行归一化处理，即每个元素除以所在行的不为 0 的所有元素的总和。

在利用随机游走计算概念相似度之前，需要建立概念图的初始分布，假设从节点 $i$ 开始随机游走，则概念图的初始分布为式（6）

$$v_0 = \begin{cases} 1, \text{如果从节点i开始随机游走} \\ 0, \text{否则} \end{cases} \tag{6}$$

给定了初始分布后，则可以根据图的随机游走算法进行游走，通过多次迭代可以得到一个稳定的概率分布，稳定的概率分布可以表示初始节点与其他节点间的语义关联强度。

图的随机游走具体算法流程如下：

a) 给定初始化矩阵 $v_0$，并令 $v=v_0$。

b) 根据图中概念间的转移概率，生成矩阵 $P$。

c) $v_{new} = \alpha * p^T * v + (1-\alpha) * v_0$。

d) $v = v_{new}$。

e) 重复步骤 c)d)，直到 $v_{new}$ 达到稳定状态或者迭代次数超过某个阈值。

从节点 i 开始随机游走，到达稳定状态后，图中的每个节点都有一个概率值，该节点的概率值反映了该节点与节点 i 的语义相似重要程度,通过实验证明在迭代 20 次各节点概率值达到稳定状态。同文献[22]$\alpha$ 的取值为 0.5。

如果从图 3 的 1 号节点“软件工程”进行随机游走，按照图的随机游走流程 3)的计算公式，第一次迭代的结果如下：

$$\begin{bmatrix} 0.500 \\ 0.000 \\ 0.000 \\ 0.250 \\ 0.000 \\ 0.250 \end{bmatrix} = 0.5 * \begin{bmatrix} 0.000 & 1.000 & 0.000 & 0.000 & 1.000 & 0.000 \\ 0.000 & 0.000 & 0.757 & 0.797 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.243 & 0.000 & 0.000 & 0.000 \\ 0.500 & 0.000 & 0.000 & 0.203 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.826 \\ 0.500 & 0.000 & 0.000 & 0.000 & 0.000 & 0.174 \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 0.5 * \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

经过 20 次迭代,图的各个节点的值达到稳定状态,概率分布满足如下关系:

$$\begin{bmatrix} 0.563 \\ 0.062 \\ 0.000 \\ 0.157 \\ 0.064 \\ 0.154 \end{bmatrix} = 0.5 * \begin{bmatrix} 0.000 & 1.000 & 0.000 & 0.000 & 1.000 & 0.000 \\ 0.000 & 0.000 & 0.757 & 0.797 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.243 & 0.000 & 0.000 & 0.000 \\ 0.500 & 0.000 & 0.000 & 0.203 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.826 \\ 0.500 & 0.000 & 0.000 & 0.000 & 0.000 & 0.174 \end{bmatrix} * \begin{bmatrix} 0.563 \\ 0.062 \\ 0.000 \\ 0.157 \\ 0.064 \\ 0.154 \end{bmatrix} + 0.5 * \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

同样，如果从 3 号节点“形式化方法”进行随机游走，经过 20 次迭代达到稳定状态，概率分布满足如下关系：

$$\begin{bmatrix} 0.121 \\ 0.229 \\ 0.569 \\ 0.033 \\ 0.014 \\ 0.033 \end{bmatrix} = 0.5 * \begin{bmatrix} 0.000 & 1.000 & 0.000 & 0.000 & 1.000 & 0.000 \\ 0.000 & 0.000 & 0.757 & 0.797 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.243 & 0.000 & 0.000 & 0.000 \\ 0.500 & 0.000 & 0.000 & 0.203 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.826 \\ 0.500 & 0.000 & 0.000 & 0.000 & 0.000 & 0.174 \end{bmatrix} * \begin{bmatrix} 0.121 \\ 0.229 \\ 0.569 \\ 0.033 \\ 0.014 \\ 0.033 \end{bmatrix} + 0.5 * \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

根据 20 次迭代得到的概率值,从 1 号节点随机游走，可以得出“面向对象”到“软件工程”的语义相似度为 0.157，而“形式化方法”没有到“软件工程”的路径，所以语义相似度概率值为 0，从 3 号节点开始随机游走，“软件工程”到“面向对象”的语义相似度为 0.121，得出下位概念到上位概念的语义相似度大于上位概念到下位概念的语义相似度；其余节点都有到“形式化方法”的路径，“软件建模”是“形式化方法”最近的上位概念，所以除了“形式化方法”本身“软件建模”到“形式化方法”的语义相似度是最大的，其余的概念到“形式化方法”路径越长语义相似度概率值越小，这样的结果和本文构建概念语义图的初始规则是相吻合的。

## 4 实验结果与分析

### 4.1 实验过程

实验下载了维基百科概念页面的解释文档内容、分类之间的所属关系、概念页面之间的内部链接 3 个数据包，用 JWPL(Java Wikipedia Library) 的 DataMachine 解析后导入 MySQL 数据库，得到 11 个 jwpl_tables.sql 表；依据 SWEBOK V3 标准选取软件工程领域核心概念 200 个，用 JWPL 接口从 MySQL 数据库中提取出每个概念的解释短文本，用中科院分词系统对其分词，为了提高准确率本文不仅去除了停用词，还加了领域词典，领域词典来源于文献[23],加了领域词典后分词时会把如“软件”和“工程”两个词合并成“软件工程”，本文得到概念和解释文本关键词共 1541 种。

对分词后的概念解释文本进行关键词抽取，本文把 LDA 主

题模型、TF-IDF 算法、TextRank 算法作为基准，采用 LDA 主题模型与 TF-IDF 算法相结合的方法、LDA 主题模型与 TextRank 算法相结合的方法进行对比,最终选择实验效果较好的 LDA 主题模型与 TF-IDF 算法相结合的方法作为本文的关键词抽取方法。

概念在维基百科中的层次关系：首先找到概念所属的分类，然后根据每个分类在维基百科的上下位层次关系用 JWPL 进行提取并存入实体三元组。

概念间的语义相似度计算实验分别采用文献[3]的对称测度 sym-KL 和图的随机游走算法在同一数据集上的结果进行了对比，与人工标注对比结果如表 3 所示。

表 3 随机游走、sym-KL 与人工标注结果对比

| 实验方法 | 完全一致 | 基本一致 | 不太一致 | 完全不一致 |
|---|---|---|---|---|
| 随机游走 | 32% | 52% | 15% | 1% |
| sym-KL | 3% | 12% | 53% | 32% |

文献[3]的 sym-KL 计算公式和 KL 计算公式分别如式(7) (8)所示。

$$sym-KL(p\,||\,q)=\frac{1}{2}(KL(p\,||\,q)+(KL(q\,||\,p))) \tag{7}$$

$$KL(p\,||\,q)=\sum_i p(x_i)\log\frac{p(x_i)}{q(x_i)} \tag{8}$$

### 4.2 实验结果分析

本文在构建领域概念语义知识有向图谱时，不仅采用了概念在维基百科中的上下位和下上位关系，还加入了两种间接的概念和其他解释文本关键词的关系：概念 *A* 在概念 *B* 的解释文本关键词中出现，概念 *A* 到概念 *B* 建立一条边；概念 *A* 和概念 *B* 的解释文本有相同的关键词 *C*，概念 *A* 到概念 *B* 建立一条边；通过这四种关系，使得概念间的语义关系更加全面和精确。

概念语义相似度计算，本文采用图的随机游走算法主要是因为随机游走算法是经过多次迭代计算后得到的稳定值，对整个概念图上的概念节点进行全局的平衡，鲁棒性较强。对称测度 *sym-KL* 方法中当两个随机分布相同时，它们的相对熵为零，当两个随机分布的差别增大时，它们的相对熵也会增大，造成 *sym-KL* 方法不如随机游走准确的原因是：只要两个概念解释文本关键词概率分布相同则 *sym-KL* 概率值为 0，两个概念的 *sym-KL* 值只与关键词的概率分布相关，而造成概念间的的语义关联较弱；然而,本文实验运用图的随机游走算法,依据节点间的连通性和转移概率进行随机游走，以传递节点间的关联关系，经过多次迭代，概念间的语义关系概率值能较准确的表示出来。

## 5 结束语

本文提出了一种面向教学的基于维基百科的领域概念语义知识库的构建方法，知识库不仅有概念间的层次语义关系，对概念本身也通过抽取概念解释文本中的关键词来表示概念的语义，在抽取概念解释文本关键词时采用了 LDA 主题模型与 TF-IDF 算法相结合、LDA 主题模型与 TextRank 算法相结合的方法较好的前者，通过概念在维基百科的上下位关系和解释文本的关键词的两种链接关系作为构建概念语义知识库的规则，最后通过图的随机游走算法对构建好的概念语义知识库计算概念的语义相似度。

目前，本文的领域概念规模不大，但本文旨在构建能够动态更新的语义知识库，如有新增概念，首先判断维基百科是否包含该概念，如果包含，用 JWPL 提取已有概念的所有父节点、子节点及兄弟姐妹节点，把新增概念放入其对应的位置；如果维基百科不包含此概念，则从百度百科提取其概念解释文本，并按照概率值从高到低依次抽取 5 个概念解释文本关键词，与知识库中其他概念解释文本关键词进行比较，如果新概念的解释文本关键词与知识库中某个概念 *D* 的解释文本关键词有大于等于两个相同，则判定此新增概念与概念 *D* 是兄弟姐妹关系，有相同的父亲节点。

在应用方面有如下三点：1.本文构建的语义知识库虽然面向软件工程领域，但该方法并不局限于该领域，同样也适合其他领域；2. 本文构建的语义知识库能够支持 MOOC 系统对学习资源按语义进行检索；3. 也能为非监督主题模型提供先验知识指导。在应用方面有如下三点：1.本文构建的语义知识库虽然面向软件工程领域，但该方法并不局限于该领域，同样也适合其他领域；2. 本文构建的语义知识库能够支持 MOOC 系统对学习资源按语义进行检索；3. 也能为非监督主题模型提供先验知识指导。

## 参考文献：

[1] 苏小康. 基于维基百科构建语义知识库及其在文本分类领域的应用研究[D]. 武汉: 华中师范大学, 2010.

[2] 陈千, 郭鑫, 王素格, 等. 文本流多粒度主题结构建模研究[J]. 中文信息学报, 2015, 29(1): 118-125.

[3] 张琪, 陈千, 郭鑫. 基于主题本体树的文本流层次主题检测技术[J]. 微电子学与计算机, 2013(07): 60-63.

[4] 伍成志. 基于维基百科的知识查找系统的研究与实现[D]. 广州: 华南理工大学, 2012.

[5] 万亿. 基于维基百科的概念图建模及其应用研究[D]. 武汉: 华中师范大学, 2014.

[6] 张涛, 刘康, 赵军. 一种基于图模型的维基概念相似度计算方法及其在实体链接系统中的应用[J]. 中文信息学报, 2015, 29(02): 58-67.

[7] 刘巧玲. 维基百科上的语义搜索[D]. 上海交通大学, 2009.

[8] Mihalcea R, Tarau P. TextRank: Bringing Order into Texts[C]//Proc of Conference on Empirical Methods in Natural Language Processing, EMNLP 2004, A Meeting of Sigdat, A Special Interest Group of the Acl, Held in Conjunction with ACL. 2004: 404-411.

[9] Blei D M, Ng A Y, Jordan M I. Latent dirichl*et a*llocation[J]. Journal of Machine Learning Research, 2003, 3: 993-1022.

[10] 韩欣, 等. 基于树状结构的语义相似度计算方法分析[J]. 微电子学与计算机, 2012, 29(05): 38-41.

[11] Fröhlich H. The GOSim package[J]. 2010.

[12] 黄果, 周竹荣. 基于领域本体的概念语义相似度计算研究[J]. 计算机工程与设计, 2007, 28(10): 2460-2463.

[13] 安建成, 武俊丽. 基于语义树的概念语义相似度计算方法研究[J]. 微电子学与计算机, 2011, 28(01): 138-141.

[14] Liu H Z, Bao H, Xu D. Concept vector for similarity measurement based on hierarchical domain structure[J]. Computing & Informatics, 2011, 30(5): 881-900.

[15] Liu H, Bao H, Xu D. Concept vector for semantic similarity and relatedness based on WordNet structure[J]. Journal of Systems & Software, 2012, 85(2): 370-381.

[16] 何夏燕. 基于汉语概念图的词汇语义相似度计算[D]. 上海: 上海交通大学, 2010.

[17] 马培军, 李东. 软件工程知识体 SWEBOK 的新进展——SWEBOK V3[J]. 计算机教育, 2013: 66-68.

[18] http: //code. google. com/p/jwpl/.

[19] http: //www. wikipedia. org/.

[20] Wang Z C, Wang Z G, Juan-Zi L I, *et al*. Knowledge extraction from Chinese wiki encyclopedias[J]. Journal of Zhejiang University Science C, 2012, 13(4): 268-280.

[21] Pearson K. The Problem of The Random Walk[J]. Nature, 1905, 72(1865): 318.

[22] Hu J, Wang G, Lochovsky F, *et al*. Understanding user's query intent with wikipedia[J]. Www Madrid! Track Search, 2009: 471-480.

[23] 赵佳鹏, 林民. 基于维基百科的领域历史沿革信息抽取[J]. 计算机应用, 2015, 35(4): 1021-1025.