doi:10.11835/j.issn.1000-582X.2020.012

面向维基百科的概念依赖关系挖掘方法

周 洋,肖 奎,曾 诚
(湖北大学 计算机与信息工程学院,武汉 430062)

摘要:在互联网技术高度发达的时代,网络上的学习资源呈现出指数型增长态势,面对各种学习对象、概念之间存在的多样化和无序性,如果能识别出之间的依赖关系,将有可能对计算机教育产生重要影响。针对该问题,提出一种面向维基百科的概念依赖关系识别方法,利用概念在维基百科中的特点,设计出一套识别概念依赖关系模型,在公共数据集上采用基于机器学习的分类算法进行测试。实验结果表明,该模型具有较高准确率和召回率,能够有效发现概念之间的依赖关系。

关键词:维基百科;概念依赖关系;机器学习;自然语言处理
中图分类号:TP311 **文献标志码**:A **文章编号**:1000-582X(2020)07-111-10

Concept dependency mining method for Wikipedia

ZHOU Yang, *XIAO Kui*, *ZENG Cheng*
(School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, P. R. China)

Abstract: In the era of highly developed Internet technology, the learning resources on the Internet show an exponential growth trend. With the diversification and disorder between various learning objects and concepts, the recognition of the dependencies between them will have a major impact on computer education. Aiming at the solution to this problem, this paper proposed a concept dependency recognition method for Wikipedia. Using the characteristics of the concept in Wikipedia, a set of recognition concept dependency model was designed, and the machine learning based classification algorithm was used to test on the public data set. The experimental results show that with high accuracy and recall rate, the model can effectively discover the dependencies between concepts.

Keywords: Wikipedia; concept dependency; machine learning; natural language processing

随着信息技术的飞速发展,网络学习作为一种新常态的学习方式得到了长足发展。利用互联网进行在线学习已成为年轻一代获取知识的重要途径,然而相比于传统的面授方式,网络课堂所存在的一个最大问题是如何保证学习者能够充分理解网络课程所学知识。那些出现在电子文档或者课程视频中的各种知识概念,如果可以搞清楚之间的语义关系,将帮助学习者更好进行理解和学习。例如当谈到“人工智能”话题时,

收稿日期:2020-01-18
基金项目:湖北省教育厅人文社科研究资助项目(19Q011)。
Supported by the Humanities and Social Sciences Foundation of Hubei Provincial Department of Education (19Q011).
作者简介:周洋(1998—),男,主要从事自然语言处理、机器学习方向研究,(Tel)17371277706;(E-mail)zhouyang1024cs@hotmail.com。
通讯作者:肖奎,男,博士,副教授,主要从事数据挖掘、众包模式行为分析方向研究,(E-mail)xiaokui1024@hotmail.com。

不可避免会介绍“机器学习”和“深度学习”等相关概念。对于一个刚刚入门人工智能的学生而言可能还不完全清楚“机器学习”与“深度学习”之间的关系，但是如果通过某种方式让其知道概念“深度学习”与“机器学习”的相关学习顺序，就会帮助学生更快理解和掌握与“人工智能”的相关概念。这种类似于课程之间的“先修关系”落实到概念层面就是所述的“依赖关系”。概念间依赖关系是建立依赖关系网络的基础，这种依赖关系网络其实就代表了这些概念知识的学习路径和学习顺序。而对概念依赖关系的识别能够为构建领域知识网络，学习对象排序、学习路径设计、课程计划的安排提供有效支持。

1 相关研究

早在2012年Talukdar and Cohen，(2012)[1]介绍了一种如何运用维基百科来挖掘概念之间的依赖关系，他们使用在社交媒体和生物计算等复杂网络分析上都有着广泛应用的随机游走(random walk)以及PageRank算法来计算概念在网络中的重要程度值，最后通过最大熵分类器进行分类，找到熵值最大的条件，但其平均正确率只达到60%左右。

随后越来越多的学者开始了概念依赖关系方面更深层次的研究：Vuong 等，2011[2]；Talukdar and Cohen，2012[1]；Liang 等，2015[3]；Wang 等，2016[4]；Scheines 等，2014[5]；Liu 等，2016[6]；Pan 等，2017[7]。其中有代表性的是Liang 等，2015 提出的一种量化概念依赖关系的方法，通过划分－1 至1 的连续区间来确定依赖关系。

随着依赖关系的判断在概念上取得的成果，人们又将目光逐步转移到具体的文本学习对象。尽管单一的概念依赖关系识别只是学习对象识别的一种基础，但任何研究依然离不开有关概念的识别。Yang 等.(2015)[8]在研究中利用各个大学课程之间所具有的“先修关系”来找到概念间的“依赖关系”，通过 Concept graph learning (CGL)构建通用的概念空间(universal concepts space)，使得任何的先决条件关系都可以使用该通用概念空间得出。落脚点依然在识别概念之间的关系层面上。Fabio 等[9-11]将概念在维基百科中具有的链接和分类特点得到特征向量，并结合自然语言处理在语义识别技术上的成果去识别具体的学习对象间的依赖关系，最后建立 web 网站实现他们的想法，但是实现较为复杂。随着大型开放式网络课程，即MOOC(massive open online courses)的兴起，网络上的课程视频资源丰富了学习对象的来源，例如 Pan 等.(2017)[7]将 MOOC 中内容当作分析对象，从每个学习视频的字幕或者课程描述中抽取其中的关键概念来设计特征去分析学习视频之间的依赖关系。因为一个或多个这些概念可以用来标注一个 MOOC，教材章节等这样的学习对象，所以根据概念间的依赖关系，可以分析出学习对象间的依赖关系，进而确定学习对象的学习路径和学习顺序。Liang 等.(2018)[8]运用集合上的二元关系来改进主动学习的查询和选择策略。但这种方法对数据集有较高要求，如果数据集比较松散或者概念总数量与要计算的概念对数量比值接近 1，那么这种策略将不会有很大作用。

相比较于前人的研究，笔者贡献主要在于充分考虑了概念在维基百科当中的 4 个特点而提出一种更有效、方便的概念依赖关系挖掘方法，并且在文献[1]的数据集上测试了所提出的方法。实验结果表明提出的模型可以很好地找出概念之间的依赖关系。

2 方 法

2.1 数据集

实验使用 Talukdar and Cohen，(2012)[1]当中来自 5 个不同领域数据集(global warming、meiosis、newton's laws、parallel postulate、public-key cryp)，将数据集分别命名为 D_1、D_2、D_3、D_4、D_5。对于每份数据集按照作者原意进行人工投票来得出概念对(A，B)之间正确的依赖关系，其间的关系分为 2 类，一类是概念 B 是概念 A 的依赖关系，即学习概念 A 之前有必要学习概念 B 作为预备知识，其余的所有情况分为另外一类，数据集的详细参数见表 1。

表 1　实验数据统计

Table 1　Experimental data statistics

Domain	# Concepts	# Pairs	# Prerequisites
D_1	351	400	43
D_2	256	346	55
D_3	324	400	44
D_4	161	200	25
D_5	180	200	27

在进行概念之间的依赖关系识别过程当中所依赖的数据来源于整个维基百科,不存在针对特定领域使用不一样的数据库文件或者模型,因此得出的模型通用性将会在后文关于跨领域分析当中得到验证。

2.2　特征描述

在进行特征描述前,有必要说明一下所涉及的概念关于维基百科的参数。在维基百科中一个概念所对应的维基页面主要包含 4 大部分:标题、摘要、正文和分类,其中在摘要和正文部分中都包含有丰富的超链接以及部分重定向概念,超链接是指从一个维基网页指向另一个目标概念网页的连接关系,而重定向概念是指概念的一个别名,例如概念“Computer machine learning”在维基百科里没有以它为标题的网页,因此通过维基百科访问此概念得到的结果最终会定位到概念“Machine Learning”的网页上,前者是概念对象的别名,后者是概念对象的正式名称。公式所用到的符号见表 2。

表 2　用到的符号及其说明

Table 2　Symbols used and their descriptions

符号	符号说明
In(C)	引用了概念 C 的概念所组成的集合
Out(C)	概念 C 所引用的概念所组成的集合
Cat(C)	概念 C 的对应分类所组成的集合
Sym(C)	概念 C 的同义词(重定向概念)所组成的集合
Sum(C)	概念 C 在维基百科中的摘要部分
Tm(C)	概念 C 的创建时间
Mid(C')	分类 C'距离根节点的最短距离
W	维基百科所有概念组成的集合

2.2.1　基于引用的特征

在维基百科中不同概念之间是可以通过网页间的超链接进行关联。如果将之间的关系映射到网络中,就可以得到以下特征。

对于给定的一个概念对(A, B),首先简单考虑那些引用概念 A、B 以及被其引用的概念数量(后文如果不做特殊说明所有的概念都来自维基百科的概念词条),通俗来讲就是输入链接数与输出链接数,表达式见式(1)、(2)、(3)及(4)。

$$\mathrm{In}\ A = |\mathrm{In}(A)|, \tag{1}$$

$$\mathrm{In}\ B = |\mathrm{In}(B)|, \tag{2}$$

$$\mathrm{Out}A = |\mathrm{Out}(A)|, \tag{3}$$

$$\mathrm{Out}B = |\mathrm{Out}(B)|, \tag{4}$$

再考虑概念 A 与概念 B 各自所引用的集合之间相似性和差异性，计算方法采用 jaccard 相似度(jaccard similarity)。对于给定的 2 个集合 $S1$ 和 $S2$，jaccard 系数定义为 $S1$ 与 $S2$ 交集的大小与并集大小的比值，jaccard 值越大说明 2 个集合之间的相似度越高，概念之间关系就越紧密，当集合都为空时 jaccard(A, B)=1。考虑概念 A 与概念 B 的 In(A)与 In(B)和 Out(A)与 Out(B)集合之间的相似度，公式定义如式(5)、(6)。

$$\mathrm{JsIn}(A,B) = \frac{|\mathrm{In}(A) \cap \mathrm{In}(B)|}{|\mathrm{In}(A) \cup \mathrm{In}(B)|}, \tag{5}$$

$$\mathrm{JsOut}(A,B) = \frac{|\mathrm{Out}(A) \cap \mathrm{Out}(B)|}{|\mathrm{Out}(A) \cup \mathrm{Out}(B)|}, \tag{6}$$

还有就是概念 B 出现在概念 A 的 Out(A)集合中，这可能表示理解概念 A 之前需要理解概念 B。为了描述概念对(A, B)之间的这种关系设计如下式(7)，并将结果标准化。

$$\mathrm{Linkin}(A,B) = \mathrm{Lin}(A,B) + \mathrm{Lin}(B,A) - 1, \tag{7}$$

$$\mathrm{Linkin}(A,B) \in \{-1,0,1\},$$

$$\mathrm{Lin}(A,B) = \begin{cases} 1 & \text{if } \exists c \in \{B\} \cup \mathrm{Sym}(B)\ c \in \mathrm{Out}(A), \\ 0 & \text{else}, \end{cases}$$

Liang 等[12]提出可以通过计算概念在维基百科当中链接的 reference distance(Ref D)值来找出依赖关系，见式(8)。

$$\mathrm{Ref}\ D(A,B) = \frac{\sum_{i=1}^{n} r(c_i,B) \cdot w(c_i,A)}{\sum_{i=1}^{n} w(c_i,A)} - \frac{\sum_{i=1}^{n} r(c_i,A) \cdot w(c_i,B)}{\sum_{i=1}^{n} w(c_i,B)}, \tag{8}$$

$$c_1, c_2, \cdots\cdots, c_n \in \mathrm{In}(A) \cup \mathrm{In}(B) \cup \mathrm{Out}(A) \cup \mathrm{Out}(B),$$

$$w(c,A) = \begin{cases} 1 & \text{if } c \in \mathrm{Out}(A), \\ 0 & \text{else}, \end{cases}$$

$$r(c,B) = \begin{cases} 1 & \text{if } B \in \mathrm{Out}(c), \\ 0 & \text{else}, \end{cases}$$

上式中 $w(c_i, A)$是概念 c_i 对概念 A 的重要程度，$r(c_i, A)$表示概念 A 是否存在于概念 c_i 所引用的集合。

概念之间的依赖关系很明显还与概念之间的语义关联度有关。一般来说概念之间的语义关联度越高，他们之间的就越有可能存在依赖关系。Witten and Milne (2008)[13]类比测量谷歌搜索引擎上的网页之间的相似度 NGD(the normalized google distance)为基础，将其中网页的超链接关系换为维基百科中的引用关系，新的测量方法命名为 NLD(normalized link distance)。Ratinov 等. (2011)[14]也通过计算概念之间 PMI (the pointwise mutual information)值来计算概念之间的相似度，计算表达式见式(9)、(10)。

$$\mathrm{NLD}(A,B) = \frac{\log \dfrac{\max(|\mathrm{In}(A)|, |\mathrm{In}(B)|)}{|\mathrm{In}(A) \cap \mathrm{In}(B)|}}{\log \dfrac{|W|}{\min(|\mathrm{In}(A)|, |\mathrm{In}(B)|)}}, \tag{9}$$

$$\mathrm{PMI}(A,B) = \log \frac{|W| \cdot |\mathrm{In}(A) \cap \mathrm{In}(B)|}{|\mathrm{In}(A)| \cdot |\mathrm{In}(B)|}。 \tag{10}$$

除此之外，还来计算概念之间引用的权重熵值，借鉴 TF-IDF(term frequency-inverse document frequency)思想以及香农熵(shannon entropy)，计算概念对之间的信息熵(Entropy)见式(11)。

$$\mathrm{Wlk}(A,B) = \begin{cases} \log \dfrac{|W|}{|\mathrm{In}(B)|} & \text{if } A \in \mathrm{In}(B), \\ 0 & \text{else}。 \end{cases} \tag{11}$$

2.2.2　基于分类的特征

每个维基百科概念都会被人为分配若干个分类,这些分类通常出现在每个维基页的底端,它可以帮助发现相似概念或者上下义词。另外维基百科的所有分类都是从一个根结点开始 Milne D, Witten I H (2013)[15],某个概念对应的分类如果距离根节点越近,说明该概念就越抽象,反之就越具体。直观上越抽象的概念就越应放在具体概念前解释。

与计算 2 个概念引用集合间的 jaccard 系数类似,得到关于概念之间分类集合的 jaccard 系数见式(12)。

$$\mathrm{JsCat}(A,B)=\frac{|\mathrm{Cat}(A)\cap\mathrm{Cat}(B)|}{|\mathrm{Cat}(A)\cup\mathrm{Cat}(B)|}。\tag{12}$$

考虑每个概念分类中距离根节点最近的分类对应的距离,其大小代表概念的抽象程度,并定义分类间的最短距离为 1,计算概念对(A, B)的分类距离差。为使得数据之间具有可比性,将该差值除以领域内最大的那个概念所对应的分类距离,详细表达式见式(13)。

$$\begin{gathered}\mathrm{Cat}D(A,B)=\frac{\min(\mathrm{Con}(A))-\min(\mathrm{Con}(B))}{\max(\mathrm{CoD})},\\ \mathrm{CoD}=\mathrm{Con}(C_1)\cup\mathrm{Con}(C_2)\cup\cdots\cup\mathrm{Con}(A)\\ \cup\mathrm{Con}(B)\cup\cdots\cup\mathrm{Con}(C_n)\\ C_1,C_2,\cdots,A,B,\cdots,C_n\in D,\\ \mathrm{Con}(A)=\{\mathrm{Mid}(C_1'),\mathrm{Mid}(C_2'),\cdots\cdots,\mathrm{Mid}(C_m')\}\\ C_1',C_2',\cdots,C_m'\in\mathrm{Cat}(A)。\end{gathered}\tag{13}$$

2.2.3　基于文本的特征

将维基百科中概念的摘要内容考虑进来,由于摘要部分具有高度的概括性和精简性,因此出现在该部分的概念会对概念依赖关系的判断具有促进作用,与式(7)的原理类似,计算概念出现在摘要部分的情况,表达式见式(14)。

$$\begin{gathered}\mathrm{Cont}(A,B)=\mathrm{Cnt}(A,B)+\mathrm{Cnt}(B,A)-1,\\ \mathrm{Cont}\in\{-1,0,1\},\\ \mathrm{Cnt}(A,B)=\begin{cases}1 & \text{if } \exists c\in\{B\}\cup\mathrm{Sym}(B)\ c \text{ 出现在 } \mathrm{Sum}(A)\text{中},\\ 0 & \text{else}。\end{cases}\end{gathered}\tag{14}$$

2.2.4　基于创建时间的特征

每个概念在维基百科中的创建时间不尽相同,但基本遵循着后创建的概念要用当前概念来进行编辑的原则,将概念的“诞生时间”考虑到挖掘依赖关系上来,表达式见式(15)。

$$\mathrm{Tim}(A,B)=\begin{cases}1 & \mathrm{Tm}(A)<\mathrm{Tm}(B),\\ 0 & \mathrm{Tm}(A)\geqslant\mathrm{Tm}(B)。\end{cases}\tag{15}$$

3　实　验

3.1　分类评价

实验选择那些具有二元分类能力的分类器进行分类,使用五种在 Weka 3.8.3(waikato environment for knowledge analysis)工具当中具有不同分类特性且广泛使用的二元分类器:J48、朴素贝叶斯 NB (naivebayes)、多层感知机 MLP(multilayerperceptron)、随机森林(RF,randomforest)和逻辑回归(RF,logistic regression),参数使用 Weka 的默认值。对于每份数据集采用 10 倍交叉验证,评估标准采用在信息检索和统计学分类领域广泛使用的准确率 P(precision)和召回率 r(recall),综合评价指标采用 F 值(F-measure),而 F-measure 是 precision 和 recall 加权调和平均,对应的计算公式见式(16)。

$$F_\beta=(1+\beta)\frac{p\cdot r}{\beta^2\cdot p+r}。\tag{16}$$

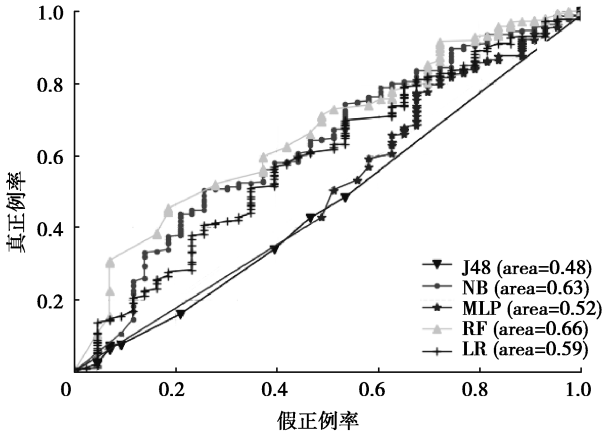
公式当中的 β 值一般为 1。另外还计算每个分类器的 AUC(area under roc curve)来度量分类器的好坏[16],通常其 AUC 的值越大,分类模型的分类效果就越好,对应不同分类器分类情况以及 ROC(receiver operating

characteristic)曲线分别见表3和图1。

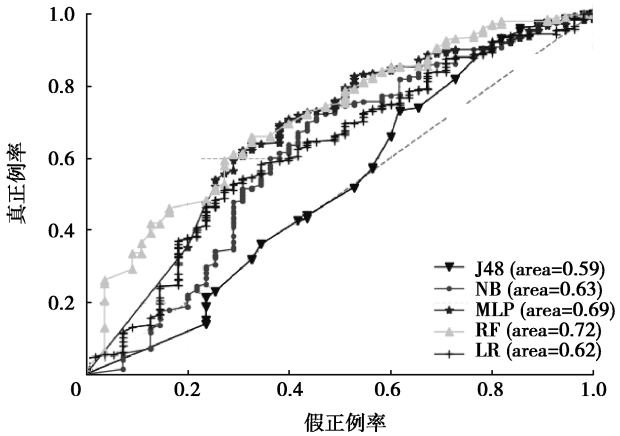
表3 所用到的符号及其说明

Table 3 Experimensal result

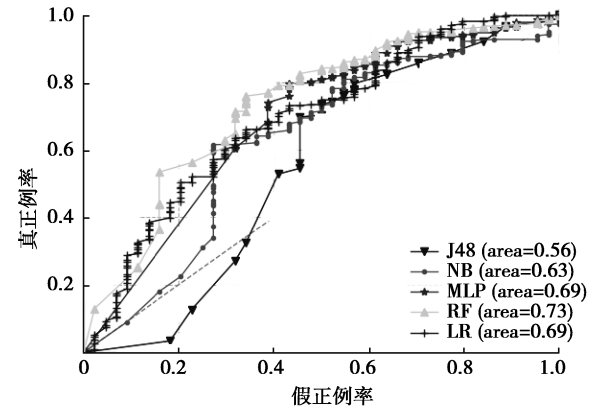
Classifier	Metric	D_1	D_2	D_3	D_4	D_5
J48	p	81.2	78.1	**85.7**	85.3	76.2
	r	88.0	82.4	87.5	85.5	83.0
	$F1$	84.0	79.4	**86.4**	**85.4**	79.2
	AUC	47.9	59.3	56.0	55.0	47.3
NB	p	82.7	77.8	85.4	**85.7**	77.3
	r	82.0	78.3	83.5	81.0	80.5
	$F1$	82.3	78.1	84.3	82.8	78.8
	AUC	63.3	63.2	63.0	63.0	57.9
MLP	p	82.9	79.8	82.6	81.6	**81.2**
	r	83.8	81.5	84.0	82.5	81.5
	$F1$	83.3	80.6	83.3	82.0	**81.4**
	AUC	51.5	68.7	69.9	73.8	57.2
RF	p	**83.5**	**81.3**	82.1	84.4	74.6
	r	**88.8**	**84.7**	87.3	**87.0**	**85.0**
	$F1$	**84.8**	**81.0**	84.0	85.2	79.5
	AUC	**66.1**	**71.6**	**73.1**	75.5	61.0
LR	p	79.6	75.5	81.9	84.5	77.8
	r	88.8	83.2	87.8	85.0	83.5
	$F1$	83.9	77.4	84.0	84.7	80.1
	AUC	58.8	62.5	68.9	**78.4**	**61.9**



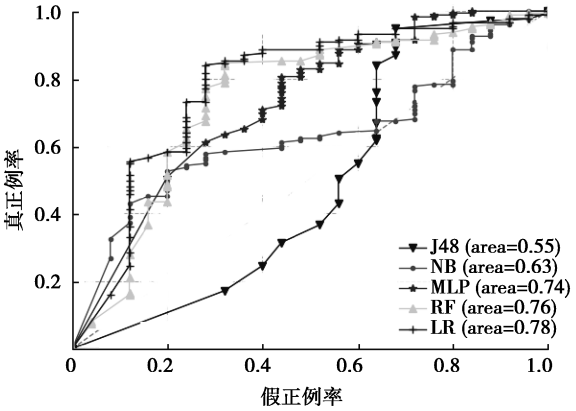
(a)全球气候变暖



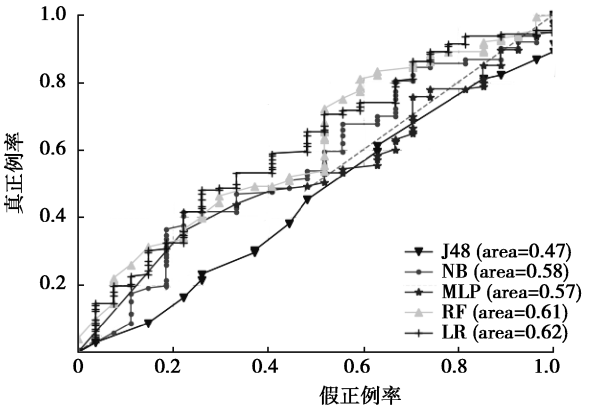
(b)减数分裂



(c)牛顿定律



(d)平行假设



(e)公钥密码

图 1　5 个数据集上的不同分类器得到的关于 ROC 曲线情况

Fig. 1　ROC curve obtained by different classifiers on five datasets

总体来看,不同的分类器之间的分类效果还是有所区别,逻辑回归相比其他分类器在准确率上表现最差,由于逻辑回归形式非常简单(类似于线性模型),因此很难去拟合出真实数据分布情况。并且研究的特征值多达 15 个,想达到一个很好的分类效果很难做到。在召回率和 $F1$ 值上朴素贝叶斯分类器要稍稍落后于其他模型,这主要由于它与其余的分类算法去学习出特征之间的关系不同,该分类器采用的是生成方法,是基于概率的参数估计方法,也就是找出特征 F_1 与特征 F_2 之间的联合分布 $P(F_1 | F_2)$,然后用 $P(F_1 | F_2) = P(F_1, F_2) / P(F_1)$ 计算得出分类,因此它在实现过程当中是假设特征之间相互独立。显然设计的多个特征相互之间并不独立,例如基于引用的特征之间是相互影响的,并且特征 JsIn、JsOut、Ref D 还包含有部分特征的结论来进行分类(In A、In B、Out A、Out B)。

另外比较不同分类器的 AUC 值可以看出 J48 的分类效果非常不理想,J48 是基于 C4.5 实现的决策树算法,看来在这里并不适用。在 5 种分类模型当中性能最好且最稳定的当属随机森林,这在 AUC 值上体现最为明显,在 $F1$ 值上分别高于最差的分类模型 2.5%、3.6%、0.7%、3.2%、0.7%,在 AUC 值上分别高于最差的 18.2%、12.3%、17.1%、20.5%与 13.7%。这可能是由于随机森林基于 Bagging 的集成学习方法[17]更适合区分概念之间的依赖关系,为了证明提出特征的有效性,使用随机森林继续进行特征分析方面的实验。

3.2　特征分析

可以将所设计的特征根据其特点分为 4 大类,其中基于引用的特征 11 个,基于分类的特征 2 个,基于文本的特征 1 个以及基于创建时间的特征 1 个,实验每次去掉其中的一类特征来观察结果的变化,数据集保持不变,分类算法采用随机森林,实验与未剔除特征的实验结果当中的 AUC 值进行比较,结果见表 4。

表4 研究方法的各个特征组的贡献情况

Table 4 Contribution of each characteristic group of research method %

特征	D_1	D_2	D_3	D_4	D_5
Link-based Features	53.7(−12.4)	62.6(−9.0)	51.9(−21.2)	59.4(−16.1)	47.3(−13.7)
Category-based Features	64.4(−1.7)	69.1(−2.5)	71.5(−1.6)	68.4(−7.1)	61.3(+0.3)
Content-based Feature	64.6(−1.5)	71.0(−0.6)	72.3(−0.8)	73.6(−1.9)	58.6(−2.4)
Time-based Feature	64.2(−1.9)	69.5(−2.1)	71.8(−1.3)	75.3(−0.2)	58.6(−2.4)
All Fearures	66.1	71.6	73.1	75.5	61.0

由上表可以明显看出在撤掉对应特征组之后，分类的 AUC 值都出现一定的下降，这就说明设计的特征是有效的，特别是基于引用的特征组当中最高的下降比高达 29%，这说明在判别概念之间的依赖关系上引用关系是一个不可忽视的重要因素，可以说它直接影响到概念依赖关系的分类，它与基于分类或者文本等特征不同，引用关系是在维基百科当中最直接明显的特征之一，在模型当中的特征 Ref D 以及 NLD 和 PMI 都是完全基于引用关系来设计，相比较于分类和文本等，基于这些特点设计出来的方法或多或少是通过间接方式挖掘出概念之间的依赖关系。

但是分析单个特征对其影响时发现不同特征组之间存在特征分配不均的现象，这就很可能导致有些特征之间会发生重叠而出现一些不必要的计算，解决该问题的方法是考虑每个特征在分类当中的作用，对不必要的特征进行删除以达到降维的目的。因此，将基于引用的特征单独拿出，分析其中每一个特征的贡献情况，实验过程和上面类似，列出在基于引用特征当中的 11 个特征由大到小的贡献情况：JsOut、Linkin、InB、RefD、InA、NLD、JsIn、OutB、OutA、PMI、Wlk。可以看出通过计算各个特征的引用 jaccard 系数其效果要比传统的 RefD 要好，但是原来用来判别语义之间关系的 NLD 与 PMI 都表现不佳，这可能说明用来判别概念之间的语义关系可能在这里不太适合去判断依赖关系，同时也侧面反映出要想挖掘出概念之间的依赖条件关系盲目从语义关系下手可能不太合适。

3.3 跨领域分析

由于实验都集中在某一个数据集上进行，没有考虑不同数据集交叉的结果，考虑跨领域条件下特征的分类情况。一个好的分类模型不仅应该可以适应领域内的数据，还更应该推广到更大范围，这对模型的可扩展性以及适用性提出了更高要求。实验每次将其中一份数据集作为测试数据，其余的 4 个作为训练数据，分类算法仍然使用随机森林，实验结果与领域内的情况进行对比，结果见图 2。

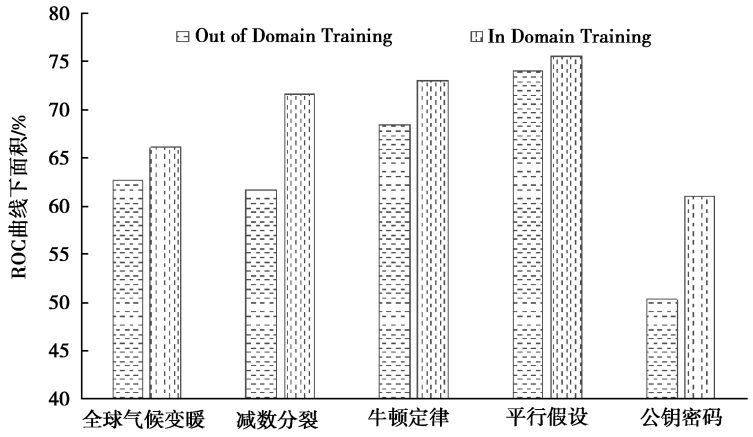


图2 跨领域分类 AUC 值(左边)与领域内的(右边)比较/%

Fig. 2 Comparison of AUC value (left) of cress domain classification with that in domain (right) %

和预想的类似，跨领域的分类效果不如领域内，造成这种现象原因是实验在进行测量的过程中都是以数据集为单位进行测量，这使得数据集与数据集之间没有过多的联系，另外由于数据集之间所涉及的领域相对来说较为孤立，因此在实际设计方法当中也很难做到统一。

3.4　方法比较

目前还没有研究者使用深度学习等较为先进的模型在依赖关系的挖掘上，因此为了说明模型的优越性，仅将上述结果与 Talukdar and Cohen[1] 和 Liang.[3] 进行比较（数据集保持一致）。为方便分析，将 Talukdar and Cohen 提出方法命名为 MaxEnt，Liang 提出的方法命名为 Ref D，另外还需说明的是在方法 Ref D 上存在 2 种形式去计算概念之间链接权重，第一种是认为概念之间链接权值相等，将此方法命名为 RefD-EQUAL，第二种链接之间的权重使用 TFIDF 方式来进行计算，将此方法命名为 RefD-TFIDF，最后方法命名为 LCCT（link-categories-content-time）。

将各种方法得到的正确率进行比较，LCCT 使用随机森林进行分类，实验结果见表 5。

表 5　不同方法正确率比较

Table 5　Comparison of accuracy of different methods　　%

数据类	MaxEnt	Ref D-EQUAL	Ref D-TFIDF	LCCT
D_1	56.8	57.4	60.1	88.75
D_2	51.0	53.0	55.7	84.68
D_3	53.9	63.7	64.6	87.25
D_4	64.7	70.5	67.9	87.0
D_5	67.1	55.1	57.7	85.0
AVG	58.7	60.0	61.2	86.74

可以看出方法性能要比其余 2 种都要好，所有的数据集正确率基本上可达到 85％以上，高于其余 2 种方法，并且还可以观察到方法在不同数据集上正确率的相互差值仅在约 4％的范围内浮动，这也说明方法具有更好的稳定性。但是这种高正确率的背后也不乏可能是过拟合所造成，在观察 weka 分类器分类结果后发现并没有出现该现象。

4　结束语

互联网的高速发展使得网络上的学习资料越来越多，21 世纪是知识经济的时代，如何通过概念间的依赖关系来帮助高校制定教学计划以及帮助学生规划适合于自己的学习路线将会是人工智能领域发展的难点。针对这一问题，提出一种基于维基百科的概念依赖关系挖掘方法，利用概念词条在维基百科上的特点，构建出依赖关系推理模型。以其中的引用关系、分类关系为基础，并结合摘要信息及创建时间，利用机器学习算法有效地识别出概念之间的依赖关系，通过对比实验验证了提出方法的优越性和有效性。但是模型在真实的概念空间上表现力还需去验证，同时目前基于文本的分类模型例如 LDA[18]、TFIDF[19]、word2Vec[20] 等优秀的方法没有运用到研究当中，且原始数据集数据量本身较小，因此深度学习模型如 DNN 在训练阶段没办法得到好的模型，实验结果将不会有很强说服力。数据集扩展以及增加算法的鲁棒性和可扩展性将是下一步研究的重点。

参考文献：

[1] Talukdar P P, Cohen W W. Crowdsourced comprehension: predicting prerequisite structure in wikipedia[C]// Proceedings of the Seventh Workshop on Building Educational Applications Using NLP. New York, USA: Association for

Computational Linguistics,2012:307-315.

[2] Vuong A, Nixon T, Towle B. A method for finding prerequisites within a curriculum[C]// Proceedings of the 4th International Conference on Educational Data Mining. Eindhoven, Netherlands: DBLP,2011:211-216.

[3] Liang C, Wu Z, Huang W, et al. Measuring prerequisite relations among concepts[C]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. New York, USA: Association for Computational Linguistics,2015:1668-1674.

[4] Wang S, Ororbia A, Wu Z, et al. Using prerequisites to extract concept maps from textbooks[C]// Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. New York, USA: ACM, 2016: 317-326.

[5] Scheines R, Silver E, Goldin I M. Discovering prerequisite relationships among knowledge components[J]. Educational Data Mining,2014: 355-356.

[6] Liu H, Ma W, Yang Y, et al. Learning concept graphs from online educational data[J]. Journal of Artificial Intelligence Research,2016,55:1059-1090.

[7] Pan L, Li C, Li J, et al. Prerequisite relation learning for concepts in MOOCs[C]// Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. New York, USA: Association for Computational Linguistics, 2017:1447-1456.

[8] Yang Y, Liu H, Carbonell J, et al. Concept graph learning from educational data[C]// Proceedings of the Eighth ACM International Conference on Web Search and Data Mining. New York, USA:ACM,2015:159-168.

[9] Gasparetti F, Limongelli C, Sciarrone F. Exploiting wikipedia for discovering prerequisite relationships among learning objects[C/OL]. Interactive Environments & Emerging Technologies for Elearning. Piscataway, NJ: IEEE,2015(2015-08-24)[2020-04-05].https://doi.org/10.1109/ITHET.2015.7218038.

[10] Limongelli C, Gasparetti F, Sciarrone F. Wiki course builder: A system for retrieving and sequencing didactic materials from Wikipedia[C/OL]. International Conference on Information Technology Based Higher Education & Training. Piscataway, NJ: IEEE, 2015(2015-08-24)[2020-04-05].https://doi.org/10.1109/ITHET.2015.7218041.

[11] Gasparetti F, De Medio C, Limongelli C, et al. Prerequisites between learning objects: Automatic extraction based on a machine learning approach[J]. Telematics and Informatics,2018,35(3):595-610.

[12] Liang C, Ye J, Zhao H, et al. Active learning of strict partial orders: A case study on concept prerequisite relations[J/OL]. Machine Learning,2018[2020-04-15]. https://www.researchgate.net/publication/322634491_Active_Learning_of_Strict_Partial_Orders_A_Case_Study_on_Concept_Prerequisite_Relations.

[13] Witten I H, Milne D N. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links[J/OL]. Association for the Advancement of Artificial Intelligence,2008[2020-04-25].http://hdl.handle.net/10289/1777.

[14] Ratinov L A, Roth D, Downey D, et al. Local and Global Algorithms for Disambiguation to Wikipedia[C]// The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference. New York, USA: Association for Computational Linguistics,2011:1375-1384.

[15] Milne D, Witten I H. An open-source toolkit for mining Wikipedia[J]. Artificial Intelligence,2013,194:222-239.

[16] Davis J, Goadrich M. The relationship between Precision-Recall and ROC curves[C]// Proceedings of the 23rd International Conference on Machine Learning. New York, USA:ACM,2006:233-240.

[17] Liaw A, Wiener M. Classification and regression by randomForest[J]. R news,2002,2(3):18-22.

[18] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of Machine Learning Research, 2003,3: 993-1022.

[19] Salton G, Buckley C. Term-weighting approaches in automatic text retrieval[J]. Information Processing & Management, 1988,24(5): 513-523.

[20] Rong X. Word2vec parameter learning explained[J/OL]. Computer ence,2014[2020-04-15]. https://www.researchgate.net/publication/268226652_word2vec_Parameter_Learning_Explained

(编辑 侯 湘)