第34卷 第1期 北京信息科技大学学报 Vol.34 No.1
2019年2月 Journal of Beijing Information Science & Technology University Feb.2019

文章编号:1674-6864(2019)01-0068-07 DOI: 10.16508/j.cnki.11-5866/n.2019.01.013

文本相似度计算研究进展综述

王寒茹,张仰森

(北京信息科技大学 计算机学院,北京 100192)

摘 要: 相似度计算是自然语言处理工作的基石。随着自然语言处理技术的发展,相似度计算的研究价值和应用价值突显。现有的计算方法因其复杂度和精确度的问题,与现实应用的需求并不匹配。针对现有需求,对于不同粒度的文本,研究出一套适合大规模实际应用的相似度计算方法体系迫在眉睫。从方法论的角度,对目前主流的相似度计算方法进行总结,介绍了不同粒度的文本相似度计算的差别以及近几年的研究进展,总结了目前相似度计算方向存在的问题,并对发展趋势进行了展望。

关 键 词: 距离公式;相似度计算方法;词语相似度;句子相似度;篇章相似度

中图分类号: TP 391.1 **文献标志码:** A

A survey on research progress of text similarity calculation

WANG Hanru, ZHANG Yangsen

(Computer School, Beijing Information Science & Technology University, Beijing 100101, China)

Abstract: Similarity calculation is the cornerstone of natural language processing. With the development of natural language processing technology, the research value and application value of similarity calculation become more and more important. However, the existing calculation methods do not match the requirements of real-world applications due to their complexity and accuracy. It is urgent to study a set of similarity calculation method system suitable for large-scale practical application for different granularity texts. From the perspective of methodology, this paper firstly expounds the current mainstream similarity calculation method, and then introduces the difference of text similarity calculation with different granularity and the research progress in recent years. Finally it summarizes the problems existing in the current similarity calculation direction and provides an outlook of development.

Keywords: distance formula; similarity calculation method; word similarity; sentence similarity; text similarity

0 引言

文本相似度计算是自然语言处理任务的基石,对后续的文本处理起着非常关键的作用。文本相似度一般指文本在语义上的相似程度,被广泛应用于自然语言处理任务的各个领域。在机器翻译领域,它可以作为翻译精确度的评价准则;在搜索引擎领域,可用于衡量检索文本与被检索文本之间的相似程度;在自动问答领域,可用来评定问题与答案之间的语义匹配度;在抄袭检测领域,通过相似度计算可以检测出两段文本的抄袭程度;在文本聚类方面,相似度阈值可以作为聚类标准;在自动文摘中,相似度可以反映局部信息拟合主题的程度。

根据相似度计算方法的特点,文本相似度可以分为字面匹配相似度、语义相似度和结构相似度。字面相似度一般采用 Jaccard 距离、最小编辑距离、最长公共子串等基本方法进行文本相似度计算。语义相似度可以从基于统计和基于规则两方面进行考虑;结构相似度计算的关键在于分析文本的句法结构。

收稿日期: 2018-09-17

基金项目: 国家自然科学基金项目(61772081)

第一作者简介: 王寒茹,女,硕士研究生;通讯作者:张仰森,男,博士,教授。

1　基于字面匹配的方法

基于字面匹配的相似度算法只是单纯从词形上考虑文本的相似度,认为“形似即义似”。车万翔等[1]采用编辑距离计算相似度,用词语代替单个汉字或字符作为基本编辑单元; 俞婷婷等[2]根据 k(n-gram 窗口的大小) 个字符在文本中出现的频率及其所占权重,用 Jaccard 距离计算 2 个文本间的相似度; 李圣文等[3]利用公共字符串的信息熵评价文本相似度。

实际上基于字面匹配的文本相似度计算方法具有很大的局限性,原因包括:

1) 语言的多义同义问题。同一个词在不同的语境下,可以表达不同的语义,例如“苹果”既可以表示水果,也可以表示科技公司; 同理,相同的语义也可以由不同的词表达,例如“的士”、“计程车”都可以表示出租车。

2) 语言的组合结构问题。词是自然语言中的最小语义单位,由词可以组成句子和篇章,不同的词序可以表达不同的语义,如“深度学习”和“学习深度”; 更进一步,还存在句法结构问题,例如“从北京到上海高铁”和“从上海到北京高铁”虽然含有的词语完全相同,但其语义完全不同。

文本相似度的计算不能只停留在字面匹配的层面,更需要语义层面的匹配,这涉及到语义的表示和计算的问题。现有的算法分别从统计和规则两方面进行考虑。

2　基于统计的经验主义方法

基于统计的经验主义思想源于 Harris 在 1954 年提出的分布假设(distributional hypothesis) 。这个假设认为具有相似上下文的词,应该具有相似的语义。其计算完全依赖于语料库,根据词汇在文本中的共现频率衡量其语义相似度。目前,根据语料将文本表示成计算机可操作的向量形式,是利用统计方法计算文本相似度的主要思路。基于构建向量的方式不同,有向量空间模型(vector space model, VSM) 、主题模型以及神经网络模型 3 种表示方式。

2.1　基于向量空间模型

VSM 将文档看成相互独立的特征项组 $(T_1,T_2,\cdots,T_n)$,并根据其在文档中的重要程度赋予其一定的特征项权重 W; 将 $(T_1,T_2,\cdots,T_n)$ 看作一个 n 维坐标系中的坐标轴, $(W_1,W_2,\cdots,W_n)$ 为相应的坐标值。这样由特征项组 $(T_1,T_2,\cdots,T_n)$ 构成了一个文档向量空间,采用空间向量间的余弦相似度计算文本相似度。

VSM 的缺陷在于: ①对于大规模语料,VSM 会产生高维稀疏矩阵,导致计算复杂度增加; ②VSM 假设文本中的各个特征词独立存在,割裂了词与词之间的关系以及段落间的层次关系。因而用向量空间进行文本相似度计算时,通常改进 TF-IDF 的计算方法以提高精确度。例如,张奇等[4]将文本用 3 个向量(V_1,V_2,V_3) 表示, V_1 中的每一维代表特征词的 TF-IDF 值, V_2 根据一个 bi-gram 是否出现取值 0 或 1, V_3 使用 tri-gram 信息,取值同 V_2, 用回归模型将 3 对向量相似度综合得到句子的相似度; 华秀丽[5]等利用 TF-IDF 选择特征项,利用知网计算文本的语义相似度。

2.2　基于主题模型

针对 VSM 中高维向量空间,一词多义和多词一义的问题,学者们提出了各种主题模型。如潜在语义分析模型和潜在狄利克雷分布模型,在词和文档之间加入主题的概念,对文本隐含主题进行建模。两篇文档是否相关不仅仅取决于字面上的词汇重复,更重要的是挖掘文字背后的语义关联。

Deerwester 等[6]于 1990 年提出潜在语义分析模型 (latent semantic analysis,LSA) ,该算法的基本思想是对大型语料库中的词语进行统计分析产生词条-文档矩阵,并采用奇异值分解(SVD) 技术剔除不重要的奇异值,从而去除文本的“噪音”,将文本从稀疏的高维词汇空间映射到低维的潜在语义空间,在低维语义空间上使用余弦距离计算文本相似度。这样做的优点在于两个相关的文本即使没有相同的词汇也能获得相似的向量表示,更加符合文本本身的关系。由于 LSA 算法过高的计算成本,LSA 并没有得到大规模的应用。

Blei 等[7]于 2013 年提出隐含狄利克雷分布模型(latent dirichlet allocation,LDA) 。它是一种对离散数据主题信息进行建模的方法,可以用来识别大规模文档集或语料库中的主题信息。文本的相似度通过计算与之对应的主题概率分布来实现。由于短文本的代表词少,LDA 对于短文本的主题挖掘并不一定能达到预期效果,因而更适用于长文本。例如王振振等[8]利用 LDA 建立文本主题空间,增强文本的向量表示。LDA 对文档的主题建模,仅保留本质信息,有助于高效处理大规模文档。

2.3　基于神经网络模型

随着深度学习在图像、语音方面取得的进展,学

者们又把目光转向了利用深度学习模型进行自然语言处理的工作。如 DSSM、ConvNet、Tree-LSTM、Siamese LSTM [9-13] 都是在对词语或者句子建模的基础上得到词向量或者句向量，并选择合适的距离公式进行相似度计算。

利用神经网络模型进行文本的相似度计算有 2 个思路。以句嵌入为例，一是直接将句子表示成句向量，如 Ryan Kiros 等 [14] 采用 seq2seq 框架，借鉴 word2vec 的 skip-gram 的方法，通过一句话来预测这句话的上一句和下一句，在模型的 encoder 层生成句向量，decoder 进行上下文的向量预测；二是从词的角度出发，组合句子中的词向量得到句向量，如 Arora 等 [15] 对一个句子中所有的词向量进行加权平均得到句向量，并采用 SVD 或 PCA 方法进行修正，在句子的相似度计算方面取得的效果比较好；Kusner 等 [16] 最小化 2 个句子中词向量的全局距离之后，用 EMD 算法来计算句子的相似度；肖和等 [17] 利用神经网络模型结合上下文信息，学习单词在语境中的向量表示，在依存句法树中分析句子中各个词语的依存关系，得到整个句子的句义表示。

3 基于规则的理性主义方法

基于规则的理性主义方法是采用人工构建的、具有规则体系的知识库进行文本相似度计算。根据知识库中定义的规则，将词汇分解成概念，这样词汇间的相似性度量就可以转化为相似性最高的概念间的相似度。

知识库中概念的组织形式，如概念间的上下位关系、同义、反义关系以及树状概念层次体系中的不同要素（节点之间的路径长度、局部网络密度、节点在树形图中的深度、节点包含的信息量等）都可以作为词汇的特征项进行相似度计算。按照知识库的种类划分，常用的语义词典包括《知网》（HowNet）、《同义词词林》、WordNet 等，常用的 web 语料库有维基百科、百度百科等。

3.1 基于《知网》的词语相似度计算

《知网》是一个以汉语和英语所表示的概念为描述对象，以揭示概念间的关系、概念所具有的属性间的关系为基本内容的常识知识库。《知网》采用嵌套式结构，把复杂概念层层分解，直到能用一组义原来表述。《知网》本质上是一种概念树结构，这个结构比较符合人的思维方式，近些年来得到学者们的广泛研究和应用。基于《知网》的词语相似度计算思想如下：

1）词语的整体相似度计算。对于 2 个词语 w_1、w_2，w_1 对应的 m 个义项（概念）分别为 s_{11}，s_{12}，…，s_{1m}，w_2 对应的 n 个义项（概念）分别为 s_{21}，s_{22}，…，s_{2n}，词语之间的相似度可以用词语分解所得概念之间相似度的最大值来表示：

$$\mathrm{sim}(w_1,w_2)=\max_{\substack{i=1\cdots m,\\ j=1\cdots n}}\mathrm{sim}(s_{1i},s_{2j}) \tag{1}$$

2）概念相似度计算。在知网中，一个概念可以用 4 种特征来描述，分别为第一基本义原描述、其他基本义原描述、关系义原描述、关系符号描述。基于“整体相似度等于部分相似度之和”的思想，概念相似度等于各个特征相似度的加权和。由于各个特征对概念的影响程度不同，部分相似性在整体相似性中所占的权重也不一样，概念相似度计算方法为

$$\mathrm{sim}(s_1,s_2)=\sum_{i=1}^{4}\left(\beta_i\prod_{j=1}^{i}\mathrm{sim}(p_{1i},p_{2j})\right) \tag{2}$$

式中 β 为权重。

3）义原相似度计算。对于 2 个义原的相似度，刘群等 [18] 提出的义原相似度的计算方法为

$$\mathrm{sim}(p_1,p_2)=\frac{\alpha}{\alpha+\mathrm{dis}(p_1,p_2)} \tag{3}$$

式中，$\mathrm{dis}(p_1,p_2)$ 为 p_1、p_2 之间的路径长度；α 为一个可调节参数，表示相似度为 0.5 时的路径长度。吴健等 [19] 提出节点深度对义原的相似度有一定的影响。义原相似度计算方法为

$$\mathrm{sim}(p_1,p_2)=\frac{\alpha\times\min(d_{p_1},d_{p_2})}{\alpha\times\min(d_{p_1},d_{p_2})+\mathrm{dist}(p_1,p_2)} \tag{4}$$

式中 d_{p_1}、d_{p_2} 分别为节点 p_1、p_2 的节点深度。

3.2 基于 WordNet 的词语相似度计算

WordNet 以同义词集合为基本构建单位，每一个同义词集合代表一个词汇的基本概念，并在概念之间建立了上下位关系、同义关系、反义关系以及整体部分关系。

目前基于 WordNet 的词汇语义相似度计算方法如表 1 所示。

3.3 基于《同义词词林》的词语相似度计算

《同义词词林》将所有的词组织在 1 个或几个树状的层次结构中，类似于 WordNet 的组织形式。由于国外已经有很多专家对 WordNet 做了详细研究，因而与其结构相似的《同义词词林》未来得到广泛应用的潜力很大。

表 1　基于 WordNet 的词语相似度计算方法

原理	典型方法	描述
基于路径距离	Path [20]	2 个概念节点在 WordNet 层次结构树上最短路径长度
	Lch [21]	对 Path 方法的改进,引入 2 个概念节点在 WordNet 层次结构树上的深度
	Wup [22]	查找 2 个概念的最近公共祖先节点,从根节点到该节点的路径长度作为相似度
	HSO [23]	在分类树中,2 个概念之间的词汇链越长,发生转向的次数越多,相似度越低
	CP/CV [24]	由概念的层次结构树中的层次决定
基于信息含量	Resink [25]	根据 2 个概念所共有的最深父节点的信息量计算相似度
	Lin [26]	相似度取决于不同概念所包含信息的共性和个性
	Jcn [27]	将最短路径的方法和基于概念节点的信息量方法融合
基于属性特征	Vector_pairs [28]	基于 WordNet 层次结构信息和语料库共现信息,每个概念释义中词语的共现词语,为其构建释义向量(Gloss Vectors) ; 根据不同词义的释义向量之间的余弦夹角衡量两者的词义相关度
	Lesk [29]	将 2 个词汇概念的释义的重合词语数量作为二者的相似度
混合方法	Li [30]	综合考虑最短路径,公共父节点在分类树中的深度以及局部密度信息
	Shi [31]	基于局部密度,信息量以及概念深度

陈宏朝等[32]使用《同义词词林》基于路径与深度的方法进行词语相似度计算,在 MC30 测试集上得到皮尔森相关系数为 0. 856。彭琦等[33]基于信息内容的方法,在 MC30 测试集上得到皮尔森相关系数为 0. 899。

3.4　基于维基百科/百度百科的相似度计算

维基百科是目前最大的百科全书,每个页面都有一个主题,页面之间通过链接相互访问。相对于《知网》、WordNet 等知识库,维基百科知识描述全面,覆盖范围广泛,更新速度迅速,因而得到学者们的青睐。

维基百科具有很好的结构化信息,可以将维基百科看作 2 个巨大的网络: ①由页面构成的网络(页面网) ,每个节点代表一个页面,节点之间的连接线代表页面之间的链接; ②由类别组成的网络(类别网) ,每个节点代表维基百科的一个类别,连接线代表 2 个节点之间存在子类和父类的关系。

基于维基百科的代表算法有以下 3 种:

Strube 等[34]提出 WikiRelate! 算法,它将基于 WordNet 的经典算法重新基于维基百科的类别网实现,用维基百科的文档类型结构、文档内容分别代替 WordNet 的概念层次结构、词汇定义。Gabrilovich 等[35]提出显性语义分析法(explicit semantic analysis,ESA) ,该方法类似于向量空间模型,首先构建语义解析器,将每个维基百科的概念页面用 TF-IDF(或其他特征抽取方法) 表示成一个概念向量,每个值表示相对应的词语与这个概念的相关程度,通过比较 2 个概念向量的相似性判断词汇的语义相似度。相比于 WikiRelate! 算法,ESA 效果更加突出。此外,Milne[36]利用了维基百科页面之间的链接信息,基于向量模型计算语义相关性,效果不如 ESA。

百度百科作为最大的中文百科全书,相似度计算方法有以下 2 种: 詹志建等[37]对百度词条的百科名片、词条正文、开放分类和相关词条 4 部分分别求相似度,通过部分相似度加权得到整体相似度; 尹坤等[38]将百度百科看成一个巨大的有向图,基于图论的思想计算相似度,通过 2 个词条所在文档之间的链接关系来衡量 2 个词条的相似度。

4　基于句法分析的方法

句法分析是一种句子结构分析方法,借助句子的依存关系进行句法分析。依存关系主张核心动词(支配成分) 为句子的中心成分,支配句子中的其他成分(从属成分) ,支配成分与从属成分之间形成某种依存关系。依存句法可以通过长距离的搭配信息,反映出句子中各成分间的语义修饰关系,与句子成分的物理位置无关。

4.1　基于骨架依存树的方法

穗志方[39]于 1998 年提出用骨架依存树的方法计算句子相似度,开辟了用骨架依存树进行相似度计算的先河。利用依存结构进行句子间的相似度计算,关键在于如何获得句子各成分间的语义依存信息。实际上这种方法并不需要考虑所有的依存关系,只需要判断对句子结构相似有决定性作用的依存关系即可,利用依存关系计算句子相似度的方法为

$$\mathrm{sim}(s_1,s_2)=\frac{\sum_{i=1}^{n}W_i}{\max(c_1,c_2)} \tag{5}$$

式中: c_1、c_2 分别为 2 个句子中的有效搭配对; $\sum_{i=1}^{n}W_i$ 为有效搭配对中匹配的总权重。匹配权重定义为: 存在 2 个有效搭配对 w_1、w_2 和 w'_1、w'_2,如果 w_1 =

w'_1 且 $w_2 = w'_2$，则匹配权重为 1；如果 $w_1 = w'_1$ 但 $w_2 \neq w'_2$ 或者 $w_1 \neq w'_1$ 但 $w_2 = w'_2$，则匹配权重为 0.5。

4.2 基于语义角色标注的方法

语义角色标注是一种浅层的语义分析技术，把句子中的某些语法成分标注为给定谓语动词的论元（语义角色），如施事、受事、事件、地点等。

田堃等[10]提出语义角色标注的汉语句子相似度算法。该方法以谓语动词为核心，在动词相似的基础上，比较相同标签下的角色相似度。计算方法如下：

1）整体相似度计算。对于包含 p 个谓词的句子 S_1 和包含 q 个谓词的句子 S_2，分别拥有包含 p 和 q 个标注句型，S_1 的标注句型集合为 $T(S_1) = \{T_{11}, T_{12}, \cdots, T_{1p}\}$，$S_2$ 的标注句型集合为 $T(S_2) = \{T_{21}, T_{22}, \cdots, T_{2q}\}$，2 个句子间的相似度计算方法为

$$\mathrm{sim}(S_1, S_2) = \frac{\sum_{(i,j)} \mathrm{sim}(T_{1i}, T_{2j})}{\max(p, q)} \tag{6}$$

式中 (T_{1i}, T_{2j}) 为标注句型的匹配对。

2）标注句型的相似匹配算法。谓语动词是句子的核心，是动作的发出者，它的相似度虽然不能完全代替句型之间的相似度，但在很大程度上能够区分标注句型间是否具有一定的相似性，因而可以通过谓语动词的相似匹配来判断标注句型的相似匹配程度。

设句子 S_1 中第 i 个谓词和 S_2 中的第 j 个谓词之间的相似度为 sim_{ij}，谓词之间的相似度矩阵为

$$\boldsymbol{A} = \begin{bmatrix} \mathrm{sim}_{11} & \cdots & \mathrm{sim}_{1n} \\ \vdots & \ddots & \vdots \\ \mathrm{sim}_{m1} & \cdots & \mathrm{sim}_{mn} \end{bmatrix} \tag{7}$$

式中 m、n 是 2 个句子中的谓词个数，因而也是 2 个句子中的标注句型数。假设 $m \leqslant n$，2 个句子间的 m 对谓语匹配关系的算法分为以下 3 步：

①找到句子中最大的元素 $\mathrm{sim}_{pq} = \max(\mathrm{sim}_{ij} \mid 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n)$，这样得到 S_1 中第 p 个谓词和 S_2 中的第 q 个谓词之间的谓语匹配对。

②删除矩阵 $\boldsymbol{A}$ 中 sim_{pq} 所在的行和列；

③循环执行前 2 步直到矩阵 $\boldsymbol{A}$ 中的行数或列数为 0。

3）标注句型间的相似度计算。对于一个含有 m 个语义角色的标注句 T，用 v 表示它的动词，$e(S) = \{e_1, e_2, \cdots, e_m\}$ 表示 S 中所有论元成分的集合，$r(S) = \{r_1, r_2, \cdots, r_m\}$ 表示 S 中所有角色标签的集合。则标注句型 T 可以表示为一个 3 元组 $(v, e(S), r(S))$。

标注句型 T_1、T_2 的相似度为

$$\mathrm{sim}(T_1, T_2) = \beta \times \mathrm{sim}(v_1, v_2) + (1 - \beta) \times \frac{\sum_{(i,j)} \mathrm{sim}(e_i, e_j)}{\max(m, n)} \tag{8}$$

式中：m 和 n 分别为句型 T_1、T_2 中包含的标注句型数；$\mathrm{sim}(v_1, v_2)$ 为 2 个动词 v_1、v_2 间的词语相似度；$\mathrm{sim}(e_i, e_j)$ 为 2 个论元 e_i、e_j 间的相似度；β 为谓词相似度在全句中所占的权重，这里取 $\beta = 0.5$，即对谓词和语义角色的相似度各赋予 0.5 的权重。

5 结束语

虽然在学术界，相似度计算已经取得丰硕的研究成果，但是随着自然语言处理技术的发展，对相似度计算所能达到的精确度也提出了较高的要求。基于以上论述，未来的研究方向值得从以下两方面考虑：

第一，计算方法的单一导致计算结果非线性偏高，基于混合建模的相似度算法将日渐丰富。基于统计的方法能够反映文本在语义、语用方面的相似性和差异，但受语料库的质量影响较大，尤其是在对于特定领域进行文本相似度计算时，语料库的质量对结果的精确度至关重要。基于规则的相似度计算方法能够弥补语料库的数据稀疏和噪声问题，但规则制定受人的主观影响较大，如果规则库不能及时更新，规则的不完善将导致不能达到预期结果。基于句法驱动的方法从句法结构的方面刻画句子的相似度，但一般不适用于长句，随着句子长度的增加算法的准确率、复杂度均呈现下降趋势。目前已有的研究表明混合方法能在一定程度上弥补单一方法的不足，提高相似度计算方法的精确度。

对于多种不同的方法，融合方法主要为加权和回归，加权的方法对于权重的选择也是一个问题，采取回归的方法需要考虑不同方法的不同特征。融合方法的选择应该简洁高效，避免陷入单纯追求准确率的提高而忽略其复杂性和实用性的“黑洞”。

所以关于使用何种技术融合以及采用哪几种算法融合还有待深入研究，如果寻找到最佳结合点，混合方法在未来必定取代现有的方法，成为一大发展趋势。

第二，基于深度学习的建模方法将成为新的发展热点。随着神经网络在语音、图像等领域都大幅度超越传统算法，词向量、卷积神经网络、长短时记

忆以及注意力模型等都被用于文本相似度计算中，在训练的过程中挖掘文本的潜在语义特征，可以解决人工构造特征而造成的特征不足问题; 并且用向量表示文本符合人的认知，因此在大数据量的情况下利用神经网络的方法进行文本处理将会成为今后的又一大发展方向，其在篇章相似度的计算方面将会大展身手。

参考文献:

[1] 车万翔，刘挺，秦兵，等. 基于改进编辑距离的中文相似句子检索 [J]. 高技术通讯，2004，14(7)：15-19.

[2] 俞婷婷，徐彭娜，江育娥，等. 基于改进的 Jaccard 系数文档相似度计算方法 [J]. 计算机系统应用，2017，26(12)：137-142.

[3] 李圣文，凌微，龚君芳，等. 一种基于熵的文本相似性计算方法 [J]. 计算机应用研究，2016，33(3)：665-668.

[4] 张奇，黄萱菁，吴立德. 一种新的句子相似度度量及其在文本自动摘要中的应用 [J]. 中文信息学报，2005，19(2)：93-99.

[5] 华秀丽，朱巧明，李培峰. 语义分析与词频统计相结合的中文文本相似度量方法研究 [J]. 计算机应用研究，2012，29(03)：833-836.

[6] Deerwester S，Dumais S T，Furnas G W，et al. Indexing by latent semantic analysis [J]. Journal of the American Society for Information Science，1990，41(6)：391-407.

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

[8] 王振振，何明，杜永萍. 基于 LDA 主题模型的文本相似度计算 [J]. 计算机科学，2013，40(12)：229-232.

[9] Huang P S，He X，Gao J，et al. Learning deep structured semantic models for web search using clickthrough data [C]//ACM International Conference on Conference on Information & Knowledge Management. ACM，2013：2333-2338.

[10] He H，Gimpel K，Lin J. Multi-perspective sentence similarity modeling with convolutional neural networks [C]//Conference on Empirical Methods in Natural Language Processing. 2015：1576-1586.

[11] Tai K S，Socher R，Manning C D. Improved semantic representations from tree-structured long short-term memory networks [J]. Computer Science，2015，5(1)：36.

[12] Mueller J，Thyagarajan A. Siamese recurrent architectures for learning sentence similarity [C]//Thirtieth AAAI Conference on Artificial Intelligence. AAAI Press，2016：2786-2792.

[13] Neculoiu P，Versteegh M，Rotaru M. Learning text similarity with siamese recurrent networks [C]//Proceedings of the 1st Workshop on Representation Learning for NLP. 2016：148-157.

[14] Kiros R，Zhu Y，Salakhutdinov R，et al. Skip-thought vectors [C]//Advances in neural information processing systems. 2015：3294-3302.

[15] Arora S，Liang Y，Ma T. A simple but tough-to-beat baseline for sentence embeddings [C]//In International Conference for Learning Representations. ICLR 2017.

[16] Kusner M J，Sun Y，Kolkin N I，et al. From word embeddings to document distances [C]//International Conference on International Conference on Machine Learning. JMLR. org，2015：957-966.

[17] 肖和，付丽娜，姬东鸿. 神经网络与组合语义在文本相似度中的应用 [J]. 计算机工程与应用，2016，52(07)：139-142.

[18] 刘群，李素建. 基于《知网》的词汇语义相似度计算 [J]. 中文计算语言学，2002，7(2)：59-76.

[19] 吴健，吴朝晖，李莹，等. 基于本体论和词汇语义相似度的 Web 服务发现 [J]. 计算机学报，2005(04)：595-602.

[20] Rada R，Mili H，Bicknell E，et al. Development and application of a metric on semantic nets [J]. IEEE Transactions on Systems，Man and Cybern，1989，19(1)：17-30.

[21] Leacock C，Miller G A，Chodorow M. Using corpus statistics and WordNet relations for sense identification [J]. Journal of

Computational Linguistics, 1998, 24(1): 147-165.

[22] Wu Z, Palmer M. Verbs semantics and lexical selection [C]//Meeting on Association for Computational Linguistics. ACL, 1994: 133-138.

[23] Hirst G, St-Onge D. Lexical chains as representations of context for the detection and correction of malapropisms [J]. WordNet: An electronic lexical database, 1998: 305-332.

[24] Kim J W. CP/CV: concept similarity mining without frequency information from domain describing taxonomies [C]//ACM International Conference on Information and Knowledge Management. ACM, 2006: 483-492.

[25] Resnik, P. Using information content to evaluate semantic similarity in a taxonomy [C]// International Joint Conference on Artificial Intelligence. 1995: 448-453.

[26] Lin D. An information-theoretic definition of similarity [C]//International Conference on Machine Learning. 1998: 296-304.

[27] Jiang J J, Conrath D W. Semantic similarity based on corpus statistics and lexical taxonomy [J]. ROCLING, 1997: 11512-0.

[28] Pedersen T, Patwardhan S, Michelizzi J. WordNet: similarity-measuring the relatedness of concepts [C]//National Conference on Artifical Intelligence. AAAI Press, 2004: 1024-1025.

[29] Banerjee S, Pedersen T. An adapted lesk algorithm for word sense disambiguation using WordNet [C]//International Conference on Computational Linguistics and Intelligent Text Processing. Springer-Verlag, 2002: 136-145.

[30] Li Y, Bandar Z A, Mclean D. An approach for measuring semantic similarity between words using multiple information sources [J]. Knowledge & Data Engineering IEEE Transactions on, 2003, 15(4): 871-882.

[31] Shi B, Yan J Z, Wang P, et al. Ontology-based measure of semantic similarity between concepts [J]. Computer Engineering, 2009, 2(19): 109-112.

[32] 陈宏朝,李飞,朱新华,等.基于路径与深度的同义词词林词语相似度计算 [J].中文信息学报,2016,30(05):80-88.

[33] 彭琦,朱新华,陈意山,等.基于信息内容的词林词语相似度计算 [J].计算机应用研究,2018,35(02):400-404.

[34] Strube M, Ponzetto S P. WikiRelate! computing semantic relatedness using wikipedia [C]// National Conference on Artificial Intelligence. 2006: 1419-1424.

[35] Gabrilovich E, Markovitch S. Computing semantic relatedness using Wikipedia-based explicit semantic analysis [C]//Proc. International Joint Conference on Artificial Intelligence. 2016: 1606-1611.

[36] Milne D. Computing semantic relatedness using wikipedia link structure [C]//Proceedings of the New Zealand Computer Science Research Student Conference. NZ CSRSC'07, 2008.

[37] 詹志建,梁丽娜,杨小平.基于百度百科的词语相似度计算 [J].计算机科学,2013,40(6):199-202.

[38] 尹坤,尹红风,杨燕,等.基于 SimRank 的百度百科词条语义相似度计算 [J].山东大学学报,2014,44(03):29-35.

[39] 穗志方,俞士汶.基于骨架依存树的语句相似度计算模型 [C]//1998 中文信息处理国际会议.1998.

[40] 田堃,柯永红,穗志方.基于语义角色标注的汉语句子相似度算法 [J].中文信息学报,2016,30(06):126-132.