语义文本相似度计算方法研究综述

李 莹[1],伍 胜[1],徐 聪[1],尹 刚[2],张 锦[1]

(1. 长沙理工大学 计算机与通信工程学院,湖南 长沙 410076;2. 头歌教学研究中心,湖南 长沙 410205)

摘 要: 语义文本相似度计算是自然语言处理领域一个关键任务,旨在衡量两个文本之间的语义相似程度。对以往经典和当前主流的语义文本相似度计算方法进行归纳和总结,将这些方法划分为传统的方法和基于深度学习的方法两大类。传统的方法又划分为基于字面匹配、基于统计和基于规则的方法。基于深度学习的方法又划分为基于词嵌入、基于句向量和基于预训练模型的方法。在进一步细分每个类别的基础上,详细介绍了各子类的典型方法,并对各种方法的基本思想、优点和局限性进行了深入分析和总结。最后,对语义文本相似度计算方法可能的发展方向进行了展望。

关键词: 文本相似度;语义相似度;自然语言处理;深度学习;预训练模型

DOI: 10. 11907/rjdk. 231965 **开放科学(资源服务)标识码(OSID):**

中图分类号: TP391.1 **文献标识码:** A **文章编号:** 1672-7800(2024)011-0001-11

A Review of Semantic Text Similarity Calculation Methods

LI Ying[1], WU Sheng[1], XU Cong[1], YIN Gang[2], ZHANG Jin[1]

(1. *School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha* 410076, *China*; 2. *EduCoder Teaching and Research Center, Changsha* 410205, *China*)

Abstract: Semantic text similarity calculation is a key task in the field of natural language processing, which aims to measure the degree of semantic similarity between two texts. Based on the summary of the traditional and current mainstream semantic text similarity calculation methods, these methods are divided into traditional methods and deep learning-based methods. The traditional methods are divided into literal matching, statistics and rule-based methods. The methods based on deep learning are further divided into the methods based on word embedding, sentence vector and pre-trained model. On the basis of further subdivision of each category, the typical methods of each subcategory are introduced in detail, and the basic ideas, advantages and limitations of each method are deeply analyzed and summarized. Finally, the possible development direction of semantic text similarity calculation is prospected.

Key Words: text similarity; semantic similarity; natural language processing; deep learning; pretrained model

0 引言

随着信息化时代的到来,越来越多的文本数据贯穿日常生活的方方面面,经常需要找到与某个查询相似的其他文本,或者判断两个文本之间的相似程度。在当今信息时代,语义文本相似度计算具有极其重要的意义,且在各领域都有广泛应用。首先,语义文本相似度计算对于信息检索非常关键。在搜索引擎中,它可以用于对搜索结果进行排序或排名,以提供与用户查询意图最相关的文本结果,有助于提高搜索引擎的用户体验和搜索结果的质量。其次,语义文本相似度计算在文本匹配和语义分析任务中发

收稿日期: 2023-12-06

扫描二维码阅读全文:

基金项目: 湖南省自然科学基金项目(2021JJ30456、2021JJ30734);工业控制技术国家重点实验室开放研究项目(ICT2022B60);国防科技重点实验室基金项目(2021-KJWPDL-17);国家自然科学基金项目(61972055)

作者简介: 李莹(1997-),女,CCF会员,长沙理工大学计算机与通信工程学院硕士研究生,研究方向为自然语言处理;伍胜(1991-),男,博士,长沙理工大学计算机与通信工程学院讲师,研究方向为计算机视觉;徐聪(1990-),女,博士,长沙理工大学计算机与通信工程学院讲师,研究方向为类脑计算;尹刚(1975-),男,博士,头歌教学研究中心高级工程师,研究方向为软件工程、实践教育等;张锦(1979-),男,博士,CCF会员,长沙理工大学计算机与通信工程学院教授、博士生导师,研究方向为软件工程、人工智能等。本文通讯作者:张锦。

挥重要作用。例如,在问答系统中,可以使用语义文本相似度计算判断问题和候选答案之间的相似度,从而找到最相关的答案[1]。此外,语义文本相似度计算还在文本分类[2]、机器翻译[3]、抄袭检测[4]、文本摘要[5]和推荐系统[6]等任务中发挥着关键作用。在文本分类中,它有助于将文本归类到正确的类别中;在机器翻译中,它可以用来衡量机器翻译结果的准确度;在抄袭检测中,可以通过语义文本相似度计算得到两个文本的抄袭程度;在文本摘要任务中,它可以用于提取和选择与原始文本相似的重要摘要内容;在推荐系统中,可以使用语义文本相似度计算以推荐与用户喜好或历史浏览记录相似的文本内容。总体而言,语义文本相似度计算在多个自然语言处理任务中具有重要意义和价值,系统地分析和研究语义文本相似度计算方法尤为必要。

然而,计算语义文本相似度是一项复杂且具有挑战性的任务。文本的多样性、歧义性以及上下文信息的处理都给相似度计算带来了困难。为了解决这些困难,许多研究者提出了一系列语义文本相似度计算的经典模型和算法。也有学者对这些计算方法进行了归纳和总结[7-8]。本文将语义文本相似度计算方法分为传统的方法和基于深度学习的方法两大类。在已有研究基础上对传统的方法重新进行了归纳和总结,并对目前基于深度学习的主流模型和方法作了补充。本文讨论这些方法的原理、优点和缺点,通过对这些方法的综合分析,提供了一个更全面的视角,对推动进一步研究和发展以及提高语义文本相似度计算的准确性、适用性具有一定作用。本文对语义文本相似度计算方法的总体分类框架如图1所示。

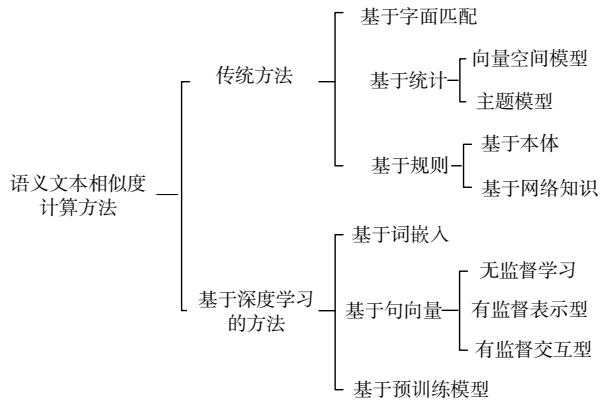

Fig. 1 Classification of semantic text similarity calculation methods

图1 语义文本相似度计算方法分类

1 传统的语义文本相似度计算方法

传统的语义文本相似度计算方法可以分为基于字面匹配的方法、基于统计的方法和基于规则的方法。基于统计的方法又包括向量空间模型和主题模型两种。基于规则的方法又分为基于本体和基于网络知识两种。

1.1 基于字面匹配的方法

基于字面匹配的方法是一种基于字符串的文本相似度计算方法,主要关注文本中字符级别的匹配度和相似度,主要包括编辑距离、汉明距离、最长公共子序列(Longest Common Subsequence,LCS)、N元模型(N-Gram)、Dice系数、Jaccard系数和Overlap等方法。

编辑距离和汉明距离都是用于度量字符串之间差异程度的方法。编辑距离更灵活,可以处理不等长的字符串,且允许插入、删除和替换操作。汉明距离更适用于比较等长的字符串,仅考虑字符替换操作。车万翔等[9]提出改进编辑距离的文本相似度计算方法,在编辑距离的基础上考虑了词语的顺序和语义信息,并且赋予了不同编辑操作不同的权重信息,使得其具有便于扩展、准确率高等优点。Rani等[10]通过消除停用词的方法改进了Levenshtein距离算法的性能,Levenshtein距离是一种常见的编辑距离算法,它主要计算一个文档转换到另一文档所需的工作量。实验结果表明,如果从文档中移除20%的停用词,则可以减少一半的计算时间。

LCS是两个字符串中最长公共字符子序列的长度。最长公共子序列不要求字符连续出现,只要字符的相对顺序保持一致即可。通过计算两文本之间的最长公共子序列以衡量文本间的相似度。Kawamitsu等[11]基于最长公共子序列计算代码之间的相似度,通过结果度量两个代码库之间源代码的重用关系。

N-Gram模型是指将文本划分为连续的n个字符组成的子序列,通过计算两个文本的N-Gram的重叠度以衡量文本之间的相似度[12]。常见的N-Gram方法包括unigram、bigram和trigram等。

Jaccard系数、Dice系数、和Overlap方法都是通过计算集合的交集和并集的比例衡量集合之间重叠程度的方法。它们的取值范围都是0~1区间,值越接近1表示集合之间的相似度越高。Niwattanakul等[13]提出利用Jaccard系数度量用户键入的关键字和索引项之间的相似度,然后根据相似度结果返回给搜索者所需要的结果。实验表明,Jaccard系数在词汇相似度度量中具有一定的适用性,但是在输入过程中,由于按键错误导致一个单词中出现额外字符时,通过Jaccard系数计算的准确度和稳定性都会下降。表1列出了基于字面匹配的代表方法,其中A、B分别代表文本A和文本B的集合。

基于字面匹配的文本相似度计算方法的优点是简单直观,易于实现和理解。这些方法关注字符串的字符级别匹配度,适用于处理插入、删除和替换等简单操作的情况。这些方法可以快速计算文本之间的相似度,并且对于短文本相似度计算任务具有较好的性能。然而,基于字符串的文本相似度计算方法也存在一些缺点。首先,这些方法通常忽略了文本的语义和上下文信息,无法捕捉到文本的深层含义;其次,对于长文本或包含复杂结构的文本,基于字

Table 1　Representation methods based on literal matching

表 1　基于字面匹配的代表方法

类型	方法	基本思想
基于字面匹配	编辑距离	通过插入、删除、替换等基本操作,计算将一个字符串转换成另一个字符串所需要的最少编辑操作次数
	汉明距离	通过计算两个等长字符串在相同位置上不同字符的个数衡量它们之间的差异或距离
	LCS	两个序列中存在的最长公共子序列
	N-Gram	两个文本所共有的N元组数量与N元组总数量的比值
	Dice 系数	$S_{dice}=\frac{2*\lvert A\cap B\rvert}{\lvert A\cup B\rvert}$,基于集合的思想
	Jaccard 系数	$S_{jaccard}=\frac{\lvert A\cap B\rvert}{\lvert A\cup B\rvert}$,基于集合的思想
	Overlap	$S_{overlap}=\frac{\lvert A\cap B\rvert}{\min(\lvert A\rvert,\lvert B\rvert)}$,基于集合的思想

符串的方法受限于字符级别的匹配,无法准确地反映文本之间的相似度。此外,这些方法对于拼写错误、同义词和词序调整等情况可能表现不佳。

1.2　基于统计的方法

基于统计的语义文本相似度计算方法是一种传统的计算语义文本相似度的方法,这类方法源自一种分布假设,认为在相似上下文中出现的词语应该具有相似的语义信息。它通过利用大规模文本语料库中的统计信息捕捉词语的上下文语境,从而更好地理解其语义。基于统计的语义文本相似度计算方法可以被视为是对基于字面匹配的方法的一种改进和扩展。

1.2.1　向量空间模型

向量空间模型(Vector Space Model,VSM)的原理是将文本表示为向量,并通过向量之间的距离或相似度衡量文本之间的相似度[14]。VSM 将文本内容看成是由相互独立的特征项所组成的集合,特征项是最小的不可分的语言单元。文本可被表示为$D=(t_1,t_2,\cdots,t_n)$,t_i代表特征项,其中每个特征项都被赋予一个权重,对应的权重向量为$W=(w_1,w_2,\cdots,w_n)$,w_i表示每个特征项在文档中的重要程度,这样由特征项组成的文本就转化成了对应的空间向量。将文本转换成向量之后,利用向量之间的距离或相似度度量方法计算文本之间的相似度。常用的度量方法包括余弦相似度、欧氏距离、曼哈顿距离等,其计算公式如式(1)—式(3)所示,其中A和B分别代表两个n维的向量。

$$cosine(A,B)=\frac{\sum_{i=1}^{n}(A_i\times B_i)}{\sqrt{\sum_{I=1}^{n}(A_i)^2}\times\sqrt{\sum_{I=1}^{n}(B_i)^2}} \tag{1}$$

$$D_{Euclidean}(A,B)=\sqrt{\sum_{i=1}^{n}(A_i-B_i)^2} \tag{2}$$

$$D_{Manhattan}(A,B)=\sum_{i=1}^{n}|A_i-B_i| \tag{3}$$

常用的特征项权重计算方法为TF-IDF。TF(词频)表示一个词语在文本中的出现频率,IDF(逆文档频率)衡量一个词语在整个语料库中的重要性,TF-IDF 通过将词语的TF值和IDF值相乘,得到一个综合的特征权重。其主要思想是如果一个词语在文本中出现频率较高(高TF值),同时在整个语料库中出现的文档较少(高IDF值),则认为该词语在文本中具有较高的重要性。TF、IDF、TF-IDF 的计算公式如式(4)—式(6)所示。

$$TF=\frac{\text{词在文档中出现的次数}}{\text{文档的总词数}} \tag{4}$$

$$IDF=log\left(\frac{\text{语料库中文档的总数}}{\text{包含该词的文档数}+1}\right) \tag{5}$$

$$TF-IDF=TF*IDF \tag{6}$$

李连等[15]提出传统的VSM没有对文本间相同特征词数量进行统计会导致计算结果不准确的问题,因而提出一种改进VSM的算法,引入两个文本相同特征词的覆盖程度这个衡量参数,提高了计算的有效性和准确性。Xu 等[16]提出 VSM 的改进方法 VSM-Cilin,该方法考虑了词语之间的语义关系,通过同义词聚类减少特征维度,在计算文本相似度时充分考虑特征项的权重,达到了较好的性能表现。Alodadi 等[17]为了解决论坛中存在重复问题和询问得不到回复的情况,提出通过 TF-IDF 算法将论坛帖子转化为对应的向量,然后通过余弦相似度计算帖子之间的相似度,将相似的现有帖子连接到新帖子的方法以解决此问题。

VSM 的原理充分利用了词语在文本中的频率和权重信息,将文本表示为高维空间中的向量。该模型具有直观易懂、可灵活调节的优点。然而,该模型也存在一些局限性:①当特征项较多即语料库较大时,VSM 会产生高维的稀疏矩阵导致维度灾难,高维稀疏矩阵需要更多的存储空间和计算资源进行处理,会增加计算复杂性和时间成本;②VSM 模型使用词语作为特征项,每个词语在向量空间中表示为一个维度,然而同义词和近义词在 VSM 中通常被视为不同的词语,具有不同的维度表示,存在一词多义和多词一义的问题;③VSM 假设文本中的各特征项(如词语、短语等)之间是独立的,没有关联或依赖关系,忽略了特征项之间的依赖性,从而可能导致文本相似度计算不准确。

1.2.2　主题模型

主题模型(Topic Model)在 VSM 模型的基础上引入了潜在主题的概念,考虑了词语的分布和上下文信息,并能够处理一词多义的问题。其基本思想是假设文本由一组潜在的主题构成,每个主题都代表了一种语义概念或话题。通过主题模型,可以了解文本中的主题分布以及每个

主题与词语之间的关系。

潜在语义分析模型(Latent Semantic Analysis,LSA)是一种基于矩阵分解的方法,与传统的VSM一样用向量表示词语和文档,不同的是LSA通过对文本数据进行奇异值分解(SVD),将高维的词语—文档矩阵经过处理转至一个更低维的语义空间中[18]。这样可以消除噪声和冗余信息,提取文本数据中的主要语义信息。在这个语义空间中,LSA可以计算文本之间的相似度,用于衡量文本的语义相关性。LSA的本质是通过降维提高文本相似度计算准确度,但该算法的计算复杂度很高且移植性较差。

概率潜在语义分析模型(Probabilistic Latent Semantic Analysis,PLSA)是一种概率生成模型,将文档生成过程建模为概率分布[19]。相比于LSA的基于矩阵分解的非概率模型,PLSA更贴近真实的数据生成过程。它使用期望最大化算法(EM)进行参数估计,通过迭代优化似然函数学习模型参数。这样可以更准确地估计文档—主题分布和词语—主题分布,提高模型性能和准确性。PLSA引入了主题分布作为文档和词语的表示,通过计算文档和词语在主题分布上的概率衡量它们之间的关联性,这使得PLSA能够更好地捕捉文本的语义信息和主题结构。

潜在狄利克雷分布模型(Latent Dirichlet Allocation,LDA)是一种基于概率图模型的主题模型,它引入了Dirichlet先验分布以建模文档的主题分布和词语的主题分布,通过变分推断或Gibbs采样学习模型参数[20]。相比PLSA,LDA在生成过程中引入了先验分布,这使得模型更稳定,更适应真实世界的文本数据。LDA能够自动地确定主题数量,而PLSA需要预先指定主题数量。

Shao等[21]发现通过LDA对文本集进行主题建模,并使用Jensen-Shannon(JS)距离计算文本相似度的方法无法区分文本主题词的语义关联,于是提出一种基于隐藏主题模型和词语共现的文本相似度算法,基于LDA模型从词语共现的角度分析文本特征词的语义相关性,增强文本对主题信息的利用,该方法能有效提高文本聚类效果。

Wang等[22]提出一种基于主题模型的中文裁判文书相似度度量方法,该方法基于LDA和标记潜在狄利克雷分布(Labeled Latent Dirichlet Allocation,LLDA),通过这种方法可以帮助法官通过案件的基本情况找到相似的裁判文书,有助于裁判过程的顺利进行。实验证明,LLDA比LDA准确率更高。

Lau等[23]提出主题驱动语言模型(Topically-Driven Language Model,TDLM),它由语言模型和主题模型两部分组成。语言模型用于捕获句子中的单词关系,主题模型学习文档中的主题信息。通过对词嵌入矩阵进行卷积操作生成文档向量,通过注意力机制将文档向量与主题相关联,最终得到文档的主题向量。

Wang等[24]将LDA与TF-IDF相结合,提出文本相似度混合模型(LDA-TF-IDF Hybrid Model,L-THM),该模型既考虑到具有不同权重的词对文本语义表示的影响,又充分利用了文本之间存在的语义信息。实验结果表明,混合模型比单一模型性能表现更好,提高了文本相似度计算准确性。

主题模型在一定程度上解决了VSM中存在的一词多义和多词一义的问题。但主题模型无法提供细粒度的语义表示,无法捕捉词语之间的精确关系。此外,在训练主题模型时,需要大量的文本数据以获取准确和全面的主题分布,否则可能存在漏洞和模糊性。

1.3 基于规则的方法

基于规则的语义文本相似度计算方法是利用人工构建的具有规范组织体系的规则库评估文本之间的相似度。基于规则的方法主要包括基于本体和基于网络知识两大类。基于本体的方法根据概念之间的上下位关系,同义反义关系和同位关系计算相似度,这类方法更侧重于在事先定义好的本体中进行语义分析和计算。基于网络知识的方法通过网页词条之间的超链接形成上下位关系进行相似度计算,它更侧重于从互联网中获取丰富的语义信息,注重实时和灵活的知识获取,并能够捕捉到更广泛的语义关联。

1.3.1 基于本体的方法

本体是指一个定义了领域特定概念的结构化知识表示模型。本体用于组织和描述领域中的语义知识,并提供一种统一的方式以表示和推理文本的语义。基于本体的语义相似度计算方法常用的语义词典有《知网》(HowNet)、WordNet和《同义词词林》等。

《知网》是一个包含中文和英文的词语语义知识库,基于《知网》的语义文本相似度计算主要分为词语相似度计算、概念相似度计算和义原相似度计算这3个部分。一个词语可以有多个不同的含义或解释,每个解释对应一个概念,一个概念又由有限个义原集合表示。义原是《知网》中对概念进行描述的最基本、不可再分割的最小单元,用于表示概念的语义特征。词语之间的相似度用它们所属概念之间相似度的最大值表示,概念之间的相似度计算可以利用各义原之间相似度的加权和计算得到。

WordNet是著名的英文语义词典,其主要特点是以词语的概念为基础,将词汇组织成一个层次结构。每个词汇条目都对应一个概念,即一组具有相似意义的词语。这些概念之间通过上下位关系、同义词关系或反义词关系等多种关系相互连接。

《同义词词林》是一种中文语义词典,它以词语为基本单位,通过上下位关系、同义关系、反义关系等语义关系将词语进行组织和分类。它在结构上与WordNet十分相似,在不同语言环境下提供了相应的词语和语义信息。

基于本体的语义文本相似度计算方法可分为基于路径距离、基于内容、基于属性和混合方法4种。表2列出了基于本体的一些代表方法。

Table 2　Ontology-based representation methods

表 2　基于本体的代表方法

分类	代表方法	基本原理
基于路径距离	Rada 等[25]、Wu 等[26]、Hirst 等[27]、Li 等[28]、Kim 等[29]	通过概念节点在层次结构树上的路径长度,并加入深度、密度和层次等以计算语义文本相似度
基于内容	Resink 等[30]、Lin 等[31]、边振兴等[32]	通过不同概念共享信息的程度衡量语义文本相似度
基于属性	Lesk 等[33]、Pedersen 等[34]	利用概念释义中共现词语数量衡量语义文本相似度
混合方法	Li 等[35]、李文清等[36]、郑志蕴等[37]	综合基于路径距离、基于内容和基于属性的方法计算语义文本相似度

基于本体的方法利用本体的结构将知识进行组织和表示,在一定程度上能够捕捉到词语之间的语义信息,避免了基于表面形式的字符串匹配的局限性。但是基于本体的方法还存在一些不足:①构建和维护一个准确且完备的本体需要专业知识和领域专家的参与,需要耗费大量的时间和精力;②并没有考虑到整个句子或文档中词语的顺序和句法结构,会导致文本相似度计算准确率不高;③依赖于事先构建好的语义词典,当新词不断出现时,人工构建的语言知识库面临需要不断更新和扩充的挑战。

1.3.2　基于网络知识的方法

基于网络知识的方法可以利用互联网上的大规模文本数据和知识资源,这些资源包含了丰富的语义信息和关系。因此,相比基于本体的方法,它覆盖更广泛的领域和主题,更新速度也较快,能够从大量的数据中学习到更全面的语义模型。基于网络知识计算语义文本相似度的方法分为基于维基百科和基于百度百科两种。

维基百科是一个半结构化的知识库,包含了多个语言版本。它可以看作是一个由网页和类别构成的巨大网络。在网页网络中,每个网页对应一个节点,节点之间的链接表示页面之间的关联,可以将链接看成是边。在类别网络中,每个类别对应一个节点,节点之间的连接表示类别之间的层次关系,可将这两个网络看成有向图模型结构。

Strube 等[38]提出的 WikiRelate! 方法是最早的基于维基百科的代表算法,其基本思想是首先检索给定单词对所引用的维基百科页面,然后提取页面所属类别以找到类别树,最后根据提取的页面和对应类别分类找到路径以计算相关性。

Gabrilovich 等[39]提出显示语义分析(Explicit Semantic Analysis,ESA)方法,该方法将文本的含义表示为基于维基百科的概念的加权向量,然后通过余弦相似度等常用的度量方法计算语义相似度。

Milne 等[40]提出一种基于维基百科链接的测量方法(Wikipedia Link-based Measure,WLM),通过在维基百科文章间的链接计算术语之间的相关性,并未考虑文本内容。该方法比 ESA 计算简单,但在准确性上不如 ESA。

Camacho-Collados 等[41]提出一种结合 WordNet 和维基百科的向量表示方法 NASARI。引入词汇特异性改进了 ESA 中使用 TF-IDF 的加权方式,并提出一种新的降维技术,实现了比 ESA 更好的性能。

Qu 等[42]提出概念表示模型(Concept Representation Model,CORM)和类别表征模型(Category Representation Model,CARM),两个模型都致力于将概念的离散属性融合到一个组件中,以弥补属性缺失并挖掘潜在语义特征。实验表明,该方法在计算概念之间的语义相似度方面比 NASARI 更有效。

Wu 等[43]针对维基百科存在语义空间变大导致生成概念向量效率降低的问题,提出一种基于维基百科的高效语义匹配方法(Wikipedia Approach to Document Classification,WMDC)。该方法定义了启发式选择规则以快速挑选相关概念,使得概念向量的生成效率提高。然后,根据文本的语义表示计算相似度实现文档准确分类,在保证文档分类准确率的情况下提高了分档分类效率。

百度百科是由百度公司推出的中文在线百科全书,相比维基百科,其覆盖领域更小,只重点关注中文内容,用于中文文本相似度计算。目前,基于百度百科的研究方法还比较少。詹志建等[44]通过计算百度百科两个词条所对应的各部分相似度,将各部分相似度计算结果加权得到整体相似度。尹坤等[45]将百度百科看成是一个大型的有向图结构,每个词条代表图中一个节点,节点之间的链接看成是边,然后引入 SimRank 算法通过链接关系计算词条之间的语义相似度。

基于网络知识的方法具有覆盖范围广、更新速度快等优点。但网络知识库的质量和完整性对计算结果具有重要影响,如果知识库中存在错误、缺失或不完整的信息,可能会导致计算结果不准确。网络知识库通常规模庞大,相似度计算涉及对知识库的检索和处理,需要较高的计算资源和时间。

2　基于深度学习的语义文本相似度计算方法

随着深度学习技术的不断发展,基于深度学习的方法在文本相似度计算任务中表现出极高的性能。这种方法利用深度学习模型的强大表达能力和自动特征学习能力,能够更好地捕捉文本之间的语义关系。

2.1　基于词嵌入的方法

基于词嵌入的方法主要关注将单词映射为向量表示,它通过对文本中的单词向量进行平均或加权平均等方式得到文本的向量表示,进而进行文本相似度计算。最早期

无监督学习词向量的表示方法有 Word2Vec[46] 和 GloVe[47]。Word2Vec 分为 Skip-Gram 和 CBOW 两种训练方式。Skip-Gram 模型的原理是通过训练神经网络使得模型能够对给定目标词语预测其正确的上下文词语；CBOW 模型与 Skip-Gram 模型相反，其目标是根据上下文词语预测中心词。GloVe 通过共现矩阵中的词频信息生成单词的向量表示，Word2Vec 在训练过程中只考虑了每个词语的局部语义信息，而 GloVe 在训练过程中考虑了全局上下文信息。Joulin 等[48]提出 FastText 方法，是一种用于生成词向量和文本分类的模型，它是 Word2Vec 的拓展，具有更高的训练和推理速度。

Le 等[49]提出了 Doc2Vec 模型，一种用于学习可变长度文本片段的固定长度特征表示模型。它与 Word2Vec 一样有两种训练方式，分别是 PV-DM 模型和 PV-DBOW 模型。PV-DM 在预测中心词时，不仅使用了上下文词向量，还引入了一个特殊的段落向量，用于表示该文本片段。这样，段落向量相当于文本片段的固定特征，而上下文词向量会根据上下文动态改变。PV-DBOW 中同样会引入段落向量，然后模型将预测目标词与段落向量联合训练，以捕捉整个文本片段的语义信息。

Kusner 等[50]提出一种新的度量一个文档的嵌入单词移动到另一文档嵌入单词所需移动的最小词移距离方法（Word Mover's Distance，WMD），利用 Word2Vec 产生高质量的词嵌入，并使用 EMD 检索算法计算两个文本之间的语义距离。WMD 是第一个将高质量词嵌入与 EMD 检索算法联系起来的方法。

Arora 等[51]提出平滑逆频率算法（Smooth Inverse Frequency，SIF），该算法首先使用预训练的词向量表示文本中的每个单词，然后通过平滑化技术降低高频词汇权重，最后通过计算平滑化的词嵌入的均值，并减去平均向量在其第一个奇异向量上的投影向量，得到代表整个句子的语义信息的最终句子向量。在文本相似度计算任务中，该方法提高了 10%~30% 的性能。

基于词嵌入的方法将离散的词语表示为连续的向量，捕捉了单词的语义信息，计算效率较高，比较适用于大规模数据。但基于词嵌入的方法通常只考虑单词本身的语义，忽略了上下文信息，在复杂语义任务中表现不佳。基于词嵌入的方法只能为每个单词生成一个固定的词向量，无法解决一词多义的问题。

2.2 基于句向量的方法

基于句向量的方法主要思想是将整个文本表示为向量，该方法捕捉了整个句子的语义信息，不再仅依赖于单词级别的匹配，有助于更全面地理解句子的语义相似度。基于句向量的语义文本相似度计算方法分为有监督学习和无监督学习两种。基于句向量的代表方法如表 3 所示。

Table 3 Representation method based on sentence vector

表 3 基于句向量的代表方法

分类		代表方法	特点
无监督学习		Skip-Though[52]、SDAE[53]、FastSent[53]、Qick-Though[54]	通过对大量文本数据进行自学习，以捕捉文本之间的相似度信息
有监督学习	表示型	DSSM[55]、CLSM[56]、LSTM-DSSM[57]、Siamese-LSTM[58]、CNN+BiLSTM[59]、ARC-I[60]	结构简单，具备出色的解释性以及易于实现的特点；缺点是缺乏文本对间的信息交互
	交互型	BiLSTM-SECapsNet[61]、ARC-II[60]、DecAtt[62]、ESIM[63]、DAM[64]	很好地考虑到了文本之间的交互关系和上下文信息

2.2.1 无监督学习

无监督学习的语义文本相似度计算方法不依赖于人工标注的数据，而是通过对数据的自学习发现语义和语境上的相似度，以便比较和度量文本之间的相似程度。

Kiros 等[52]训练了一个编码器—解码器模型 Skip-Thought，该模型依赖于一个有连续文本的训练语料库，通过对一个句子的编码预测其上下文，编码器将单词映射到句子向量，解码器用于生成周围句子。实验表明，Skip-Thought 模型能学到高度通用的句子表示，但其训练速度较慢，而且并未考虑到词语的顺序信息。

Hill 等[53]提出顺序去噪自动编码器（Sequential Denoising Autoencoders，SDAEs）和 FastSent 两个无监督短语或句子表示学习模型。SDAEs 解决了 Skip-Thought 模型对句子连贯性的依赖问题，可以处理任意顺序的句子集。FastSent 解决了 Skip-Thought 模型训练速度慢的缺点，给定一个句子的词袋模型表示，该模型能简单地预测相邻句子。

Logeswaran 等[54]提出一个简单而通用的框架 Qick-Thought 以学习句子表征，该框架将预测句子上下文的问题转换为分类问题，训练速度极大提高，同时也获得了更高的性能。

无监督学习方法不需要大量的标注数据，因此可以用于大规模文本相似度计算，而无需人工标注大量的训练数据。但由于无监督方法不考虑标签信息，它们可能对数据中的噪声比较敏感，从而导致性能下降。

2.2.2 有监督表示型

有监督的语义文本相似度计算方法依赖于提供一组带有标签的文本对作为训练数据，这些文本对通常包含两个句子和对应的相似度标签，表示两个句子的相似程度。模型通过学习从句子到相似度得分的映射关系进行训练，以便在之后的相似度计算中预测未标记文本对的相似度。有监督表示型模型的主要任务是将每一对文本转化为语义向量，然后在最后一层对待匹配的两向量进行相似度计算，该类模型更侧重于语义表示层面的构建。

Huang 等[55]提出一种双塔结构模型(Deep Structured Semantic Model,DSSM),它是基于神经网络的最早的语义文本相似度计算模型之一,为后续更多优秀模型奠定了基础。DSSM 由输入层、表示层和匹配层组成。两个文本通过双塔结构将输入的文本转换为句向量,即将句子或文档表示为固定维度的向量,捕捉文本的语义信息,然后通过余弦相似度计算得到两文本的语义相似度。DSSM 模型使得相似文本对的向量在语义空间中更接近,而不相似文本对的向量则更远离。DSSM 模型结构简单、计算较快,但由于其用词袋模型表示文本,故在一定程度上忽略了语序信息。

Shen 等[56]将潜在语义分析与卷积神经网络(Convolutional Neural Network,CNN)相结合,得到了卷积潜在语义模型(Convolutional Latent Semantic Model,CLSM)。先通过卷积神经网络捕捉文本的局部特征,再通过最大池化捕获全局信息,最终得到更加丰富的特征用于表示文本向量。与 DSSM 相比,CLSM 中引入了卷积层,在一定程度上考虑了上下文信息,在性能上提高了近 10%。

Palangi 等[57]将长短期记忆网络(Long Short-Term Memory,LSTM)与 DSSM 相结合得到 LSTM-DSSM 模型,第一次将 LSTM 应用于信息检索任务[65]。通过 LSTM 编码器将文本序列逐词编码成连续的隐含语义表示,捕捉文本的长期依赖关系。两个文本分别通过 LSTM 编码器得到对应的语义向量,然后通过余弦相似度计算文本相似度。LSTM-DSSM 相比 CLSM 可以获取更加长距离的上下文语义信息,但计算更加复杂。

Bao 等[58]提出一种引入了注意力机制的 Siamese-LSTM 架构,用于计算语义文本相似度。Siamese-LSTM 架构有两个相同的 LSTM 子网络,两个子网络共享相同的权重,每个子网络用于处理句子对中的一个句子。实验证明,加入注意力机制的 Siamese-LSTM 能够获得更丰富的语义信息,取得更好的性能。

Yuan 等[59]提出一种 CNN 与 BiLSTM 结合的 Siamese 网络应用于数学领域的文本相似度计算方法[66]。首先,通过 CNN 和 BiLSTM 分别提取代表局部和全局的上下文特征;其次,将局部和全局特征进行融合和拼接,得到句子的丰富语义表示;最后,通过余弦相似度计算相似度值。

有监督表示型模型具有结构简单、解释性强和易于实现等优点。但也存在一些不足,其在计算相似度时仅抽取了各文本最后的语义向量,并没有考虑到文本对间的信息交互,容易出现语义偏移。

2.2.3　有监督交互型

有监督交互型模型基于表示型模型中缺乏文本对间词法、句法信息的交互而提出。这种模型考虑了文本间的信息交互,从而能提取到句子对间更丰富的交互信息。

Hu 等[60]提出 Architecture-I(ARC-I)和 Architecture-II(ARC-II)两种网络结构。ARC-I 在本质上采用了孪生网络架构,使用 CNN 分别对输入句子提取特征,但在两句子在特征提取过程中相互独立,并没有交互行为,到形成最终的句向量后再进行匹配运算,这会影响句子语义相似度计算结果。基于 ARC-I 在句子建模中的缺点,又提出了 ARC-II,让两个句子在形成最终句向量之前通过一维卷积操作对词进行交互,考虑了词序信息和交互信息,从而提取到更加完整的特征信息。

Zhang 等[61]将 BiLSTM、CapsNet[67]和 SENet[68]相结合,并引入互注意力机制得到混合模型 BiLSTM-SECapsNet。BiLSTM 用于提取文本的全局信息,在 BiLSTM 层之后引入互注意力机制得到文本特征间的注意力权重,增强两文本间的交互,得到更充分的语义信息。SECapsNet 由 CapsNet 和 SENet 组成,CapsNet 用于提取文本的局部特征,它用动态路由代替了池化操作,有效减少了语义信息丢失。SENet 通过学习得到各局部特征的重要性,将经过 BiLSTM 得到的特征矩阵与通过 SECapsNet 得到的特征矩阵进行融合,融合后的特征再通过 BiLSTM 层获取上下文信息,得到两个文本的相似度矩阵,然后将两相似度矩阵进行融合,池化和全连接等操作以度量语义文本相似度。该模型解决了 CNN 只能提取局部语义信息而不能获得上下文交互信息以及 RNN 在处理长文本时无法解决长距离依赖性等问题。

Parikh 等[62]提出一个具有更少参数的轻量级模型 DecAtt,该模型第一次在句子对建模中引入了注意力机制,文中提到了两种注意力机制,分别为文本间的注意力机制和文本内的注意力机制。通过文本间的注意力机制可以获得句子的交互表示。文本内的注意力机制将输入的向量表示进行自对齐作为新的输入表示,实验表明文本内的注意力机制可以提高模型表现性能,但该模型并没有考虑到词语的顺序和上下文语义信息。Chen 等[63]对 DecAtt 进行改进得到了增强时序推理模型(Enhanced Sequential Inference Model,ESIM),基于 DecAtt 的缺点,引入了 BiLSTM 层,BiLSTM 通过同时从前向和后向两个方向处理输入序列,将上下文信息捕捉得更全面,从而提高了模型对序列数据的理解能力。

Zhou 等[64]受 Transformer 的启发,提出深度注意匹配网络(Deep Attention Matching Network,DAM),它是一种采用自注意力和互注意力两种机制的模型[69]。自注意力机制使得句子能够聚焦于自身,从而捕捉词与词之间的内在依赖。通过堆叠多层自注意力机制,可以得到不同粒度的语义表示。而互注意力机制则用于捕捉上下文和回复之间潜在匹配片段的依赖关系。通过这种双重注意力机制,DAM 在文本匹配任务中表现出色,特别是在考虑句子内部和句子间的依赖关系时,能够提升性能。

有监督交互型模型摒弃了表示型模型后匹配的思想,充分考虑文本间的信息交互,可以很好地把握语义焦点,有效减少了语义偏差。虽然交互型模型效果显著,但相比

表示型模型，其结构更为复杂，计算成本也更高。

2.3 基于预训练模型的方法

基于预训练的语义文本相似度计算方法是指利用预训练模型计算文本之间的语义相似度，该方法利用在大规模未标记数据上预训练的语言模型，学习到丰富的语义信息，并在文本相似度计算任务中进行微调或迁移学习。基于预训练模型可以捕捉单词、短语和句子之间的上下文关系和语义信息，从而提供更准确、更全面的文本表示。

Peters 等[70]提出一种语言模型嵌入表示方法（Embeddings from Language Models，ELMo），其主要特点是将多层双向 LSTM 网络堆叠在一起，形成了一个深层的双向语言模型。每一层都可以捕捉不同级别的上下文信息，从而生成多层的词嵌入，将多层的词嵌入进行加权平均得到最终的词向量。通过 ELMo 模型，可以动态地学习同一个单词在不同语境下对应的不同词向量，解决了一词多义的问题。虽然 ELMo 模型充分考虑了上下文信息，但是模型在计算时新的输入总是依赖之前的输出结果，导致无法并行从而影响运行速度。ELMo 中采用两个独立的单向 LSTM 拼接以实现对上下文的考虑，在本质上还不是真正意义上的双向语言模型。

Devlin 等[71]提出 BERT 模型，它是一个基于 Transformer 的深层双向语言模型，其整体框架由 Transformer 的多层 Encoder 部分堆叠组成，BERT 预训练过程中引入了掩码语言模型（Mask Language Model，MLM）和下一句预测（Next Sentence Prediction，NSP）两个任务。通过 MLM 任务能够捕捉到词语之间的关系和上下文信息，从而得到更具有语义表示能力的词向量，实现预训练模型的深层双向表示。通过 NSP 任务与 MLM 任务相结合，使得 BERT 模型在预训练阶段能够同时学习到单词级别和句子级别的语义信息。BERT 不仅具有强大的表征能力，在语义文本相似度计算和其他自然语言任务上都表现出良好的性能，而且解决了 ELMo 中无法并行以及避免了长序列容易产生梯度消失的问题。

Liu 等[72]对 BERT 模型进行改进得到了 RoBERTa 模型，RoBERTa 在更大的无监督文本数据中进行训练，并且移除了 BERT 中的 NSP 任务。RoBERTa 直接使用连续的文本块进行预训练，而不再需要预测文本是否为原始文档中的相邻句子，这样的改进有助于更好地学习文本之间的上下文关系。RoBERTa 中采用动态掩码机制，比 BERT 中的静态掩码更优。RoBERTa 在性能上相较于 BERT 有所提升，但其训练时间和计算资源也更多。

Reimers 等[73]提出 Sentence-BERT（SBERT）模型，也是基于 BERT 进行改进得到的模型。因为 BERT 在计算语义文本相似度时需要将两个句子进行拼接作为一个整体送进模型中，这会导致计算开销非常大。SBERT 使用两个 BERT 作为子网络，将句子对分别送进两个子网络，得到两个固定大小的句子向量表示，再通过余弦相似度、曼哈顿距离或欧式距离等方法计算得到句子的语义相似度。SBERT 极大减少了计算开销，并提高了计算效率。

Li 等[74]和 Su 等[75]发现通过 BERT 编码得到的句子向量存在各向异性和向量分布不均匀等缺点，提出了 BERT-flow 和 BERT-whitening 模型，这两个模型都是为了解决各向异性和向量分布不均匀的问题。BERT-flow 采用一种流式可逆变换将各向异性向量转换到一个标准的高斯分布空间，即各向同性且分布较均匀的空间，通过这种转换有效地提升了模型性能。BERT-whitening 在 BERT-flow 的基础上，采用白化操作将基于 BERT 的句向量准换成标准正交基。这种方法以更加简单的方式对向量进行空间分布转换，达到了与 BERT-flow 相当甚至更好的性能，并且优化了内存存储，加快了检索速度。

Gao 等[76]提出一个简单的对比学习框架 SimCSE，通过对比学习方式将句向量的各向异性空间正则化，使其变得更加均匀。与 BERT-flow 和 BERT-whitening 通过后处理方式解决各向异性的问题不同，SimCSE 通过仅使用 Dropout 构造正样本对的对比学习范式，方法简单且效果非常好，无论是有监督还是无监督的方式，模型性能都优于以往方法。但 SimCSE 也存在一些不足，由于 SimCSE 通过 Dropout 构造出的正例对的长度相同，而负例对的长度则不同，这会导致模型更倾向于认为长度相近的两个句子在语义上更相似。基于 SimCSE 存在的缺点，Wu 等[77]在 SimCSE 的基础上提出增强型 SimCSE（Enhanced SimCSE，ESimCSE），通过单词重复和动量对比两种方式分别对正例和负例进行重构，通过实验表明这种改进方式对性能提升具有显著效果。

Yan 等[78]提出一种自监督学习句子表征的对比学习框架 ConSERT。它通过对比学习以一种无监督的方式将原始句子通过多种数据增强的方式得到两个新句子，并让这两个句子相互靠近，同时让其他句子与其远离，从而学习到句子的有效表示。这样将 BERT 产生的句向量在空间上作了变换，成功解决了句向量空间坍塌问题。ConSERT 在性能上比 BERT-flow 和 BERT-whitening 更优，但比 SimCSE 要差。

Chuang[79]等提出一种基于句子间差异的无监督对比学习框架 DiffCSE，DiffCSE 是等变对比学习的一个实例，它通过将基于 Dropout 的数据增强方式作为不敏感变换学习对比学习损失，并通过将基于单词替换的增强方式作为敏感变换学习原始句子和编辑句子的差异，有利于编码器获得更好的句向量表示。DiffCSE 在语义文本相似度计算任务中取得了更先进的结果，相比无监督的 SimCSE 在性能上提升了 2.3%。

Zhou[80]等发现以往工作中采用批次内负采样或数据中随机负采样的方式可能会导致采样偏差，从而使得不恰当的负例被用来进行对比学习句子表示，最终损害表示空间的对齐性和均匀性等问题。为解决这些问题，提出一个

无监督对比学习句子表示框架(Debiased Contrastive Learning of unsupervised sentence Respresentations,DCLR),设计一种实例加权方法以惩罚假负例,有效提升语义空间的对齐性。并且,采用一种可梯度更新的噪声负例生成方法以提高语义空间的均匀性。

Lee[81]等提出一种新的基于最优传输距离的度量方法RCMD和一个基于RCMD的对比学习框架CLRCMD,将RCMD距离度量整合到最先进的对比学习框架中,将token-level的语义信息融合到句子表示中,解决了以往工作中普遍存在的基于平均池化计算相似度不足以有效捕捉句子间token-level的可解释性问题。CLRCMD准确地预测了句子对的相似度并获得了更好的可解释性。

Jiang等[82]发现各向异性不是导致原生BERT表现差的主要原因,提出静态词向量中的偏见和BERT中的无效层才是导致在句子语义相似度计算方面表现不好的原因。为解决静态词向量偏见和BERT中的无效层所带来的不良影响,提出一种基于提示的句子嵌入方法PromptBERT。该方法可以减少词嵌入偏见并提高句子表征的性能,并使原有的BERT层更有效。PromptBERT在有监督和无监督的情况下与SimCSE相比性能都有提升。

Zeng等[83]将原型对比学习的思想引入无监督句嵌入学习表示中,提出基于提示衍生虚拟语义原型的对比学习方法(Contrastive learning method with Prompt-derived Virtual semantic Prototypes,ConPVP),该方法基于SimCSE框架,并进一步利用了语义模型的概念。ConPVP方法为每个实例构造虚拟语义原型,并利用提示的否定形式推导否定原型。利用原型对比损失约束anchor句嵌入接近其对应的语义原型,远离否定原型和其他句子原型。实验表明,相比PromptBERT,ConPVP方法性能有一定提升,证明了原型对比学习方法的有效性。

基于预训练模型的方法已经在自然语言处理领域取得了显著成功,并在语义相似度任务中获得了很高的性能。基于预训练的方法提高了模型泛化能力和计算效率,降低了对标注数据的依赖。目前,基于预训练模型的方法已经成语义文本相似度计算的常用方法。

3　总结与展望

语义文本相似度计算方法研究在自然语言处理领域具有广泛应用和深远意义。本文对语义文本相似度计算的众多方法进行了梳理,并通过系统分析和总结对其进行了详细分类。将语义文本相似度计算划分为传统的方法和基于深度学习的方法两大类。传统方法在准确度方面因依赖于特征工程和规则,通常在特定任务上表现良好,但受限于领域知识,适用性有限。传统方法通常具有较高的效率,因为它们通常不需要大规模的数据和复杂的计算。基于深度学习的方法在准确度和应用广度上具有优势,能够自动捕捉语义信息,适用于多种任务。基于深度学习的方法虽然训练和计算复杂,需要大量数据,效率也较低,但适用于更广泛的应用领域。

未来,语义文本相似度计算方法研究将继续朝着更深入、更准确、更高效的方向发展。随着深度学习技术的不断进步,基于深度学习的语义文本相似度计算方法受到广泛关注和研究,尤其是基于预训练模型的方法成为近年来研究的热点和趋势,这些方法能够更好地捕捉文本之间的复杂语义信息,取得更出色的性能。近期,对比学习和提示学习的方法被提出并应用于语义文本相似度计算,是目前比较主流的方法,今后预计将会有越来越多的研究者基于这些方法开展创新性研究,以进一步提高语义文本相似度计算准确性和效率,为广泛的应用领域提供更多可能性。未来,这些方法将得以不断发展和完善,探索如何更有效地利用对比学习和提示学习方法计算语义文本相似度将是今后研究的一个关键方向。

参考文献:

[1] DAS A, MANDAL J, DANIAL Z, et al. A novel approach for automatic bengali question answering system using semantic similarity analysis[J]. International Journal of Speech Technology,2020,23(4): 873-884.

[2] STEIN A R, JAQUES A P, VALIATI F J. An analysis of hierarchical text classification using word embeddings[J]. Information Sciences, 2019, 471: 216-232.

[3] NGUYEN-SON H Q, THAO T, HIDANO S, et al. Machine translated text detection through text similarity with round-trip translation[C]//Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021: 5792-5797.

[4] ABDI A, IDRIS N, ALGULIYEV R M, et al. PDLK: plagiarism detection using linguistic knowledge[J]. Expert Systems with Applications,2015,42(22): 8936-8946.

[5] SARKAR A, HOSSEN M S. Automatic bangla text summarization using term frequency and semantic similarity approach[C]//Dhaka:2018 21st International Conference of Computer and Information Technology,2018.

[6] MAGARA M B, OJO S O, ZUVA T. A comparative analysis of text similarity measures and algorithms in research paper recommender systems[C]//Durban:2018 Conference on Information Communications Technology and Society,2018.

[7] WANG C L, YANG Y H, DENG F, et al. A review of text similarity calculation methods[J]. Information Science,2019,37(3): 158-168.
王春柳,杨永辉,邓霏,等. 文本相似度计算方法研究综述[J]. 情报科学,2019,37(3): 158-168.

[8] HAN C C, LI L, LIU T T, et al. Approaches for semantic textual similarity[J]. Journal of East China Normal University (Natural Science),2020,66(5): 95-112.
韩程程,李磊,刘婷婷,等. 语义文本相似度计算方法[J]. 华东师范大学学报(自然科学版),2020,66(5): 95-112.

[9] CHE W X, LIU T, QIN B, et al. Chinese similar sentence retrieval based on improved editing distance[J]. High Technology Letters,2004,14(7): 15-19.
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.

[10] RANI S, SINGH J. Enhancing Levenshtein's edit distance algorithm for evaluating document similarity[C]//Proceedings of the First International Conference on Computing, Analytics and Networks,2018: 72-80.

[11] KAWAMITSU N, ISHIO T, KANDA T, et al. Identifying source code reuse across repositories using lcs-based source code similarity[C]//Proceedings of the 14th International Working Conference on Source Code Analysis and Manipulation, 2014: 305-314.

[12] KONDRAK G. N-gram similarity and distance[C]//Proceedings of the International Symposium on String Processing and Information Retrieval, 2005: 115-126.

[13] NIWATTANAKUL S, SINGTHONGCHAI J, NAENUDORN E, et al. Using of Jaccard coefficient for keywords similarity[C]//Proceedings of the International Multiconference of Engineers and Computer Scientists, 2013: 380-384.

[14] SALTON G, WONG A, YANG C S. A vector space model for automatic indexing[J]. Communications of the ACM, 1975, 18(11): 613-620.

[15] LI L, ZHU A H, SU T. Research and implementation of an improved text similarity algorithm based on vector space [J]. Computer Applications and Software, 2012, 29(2): 282-284.
李连, 朱爱红, 苏涛. 一种改进的基于向量空间文本相似度算法的研究与实现[J]. 计算机应用与软件, 2012, 29(2): 282-284.

[16] XU L H, SUN S T, WANG Q. Text similarity algorithm based on semantic vector space model[C]//Okayama: 2016 IEEE/ACIS 15th International Conference on Computer and Information Science, 2016.

[17] ALODADI M, JANEJA V P. Similarity in patient support forums using TF-IDF and cosine similarity metrics[C]//Proceedings of the 2015 International Conference on Healthcare Informatics, 2015: 521-522.

[18] LANDAUER T K, DUMAIS S T. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge[J]. Psychological Review, 1997, 104(2): 211-240.

[19] HOFMANN T. Probabilistic latent semantic analysis[C]//Proceedings of the Fifteenth Conference on Uncertainty in ArtificialIntelligence, 1999: 289-296.

[20] BLEI D M, NG A Y, JORDAN M I. Latent dirichlet allocation[J]. Journal of Machine Learning Research, 2003, 3(1): 993-1022.

[21] SHAO M, QIN L. Text similarity computing based on LDA topic model and word co-occurrence[C]//2014 2nd International Conference on Software Engineering, Knowledge Engineering and Information Engineering (SEKEIE 2014), 2014: 199-203.

[22] WANG Y, GE J, ZHOU Y, et al. Topic model based text similarity measure for chinese judgment document[C]//Data Science: Third International Conference of Pioneering Computer Scientists, Engineers and Educators, 2017: 42-54.

[23] LAU J H, BALDWIN T, COHN T. Topically driven neural language model [C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017: 355-365.

[24] WANG J, XU W, YAN W, et al. Text similarity calculation method based on hybrid model of LDA and TF-IDF[C]//2019 3rd International Conference on Computer Science and Artificial Intelligence, 2019: 1-8.

[25] RADA R, MILI H, BICKNELL E, et al. Development and application of a metric on semantic nets[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1989, 19(1): 17-30.

[26] WU Z, PALMER M. Verb semantics and lexical selection[C]//ACL Proceedings of Annual Meeting on Association for Computational Linguistics, 1994: 133-138.

[27] HIRST G, ST-ONGE D. Lexical chains as representations of context for the detection and correction of malapropisms[M]. Cambridge: MIT Press, 1998.

[28] LI Y, BANDAR Z A, MCLEAN D. An approach for measuring semantic similarity between words using multiple information sources [J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15 (4): 871-882.

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

[30] RESNIK P. Using information content to evaluate semantic similarity in a taxonomy[C]//Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1995: 448-453.

[31] LIN D. An information-theoretic definition of similarity[C]//Proceedings of the Fifteenth International Conference on Machine Learning, 1998: 296-304.

[32] BIAN Z X. Research on IC parameter model of conceptual semantic similarity in WordNet[J]. Computer Engineering and Applications, 2011, 47 (19): 128-131.
边振兴. WordNet 中概念语义相似度 IC 参数模型研究[J]. 计算机工程与应用, 2011, 47(19): 128-131.

[33] LESK M. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone[C]//Proceedings of the 5th Annual International Conference on Systems Documentation, 1986: 24-26.

[34] PEDERSEN T, PATWARDHAN S, MICHELIZZI J. WordNet: similarity-measuring the relatedness of concepts[C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2004: 38-41.

[35] LI Y, BANDAR Z A, MCLEAN D. An approach for measuring semantic similarity between words using multiple information sources [J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15 (4): 871-882.

[36] LI W Q, SUN X, ZHANG C Y, et al. A method for calculating semantic similarity of ontology concepts[J]. Acta Automatica Sinica, 2012, 38(2): 229-235.
李文清, 孙新, 张常有, 等. 一种本体概念的语义相似度计算方法[J]. 自动化学报, 2012, 38(2): 229-235.

[37] ZHENG Z Y, RUAN C Y, LI L, et al. Research on adaptive comprehensive weighting algorithm for semantic similarity of ontology[J]. Computer Science, 2016, 43(10): 242-247.
郑志蕴, 阮春阳, 李伦, 等. 本体语义相似度自适应综合加权算法研究[J]. 计算机科学, 2016, 43(10): 242-247.

[38] STRUBE M, PONZETTO S P. WikiRelate! computing semantic relatedness using Wikipedia[C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2006: 1419-1424.

[39] GABRILOVICH E, MARKOVITCH S. Computing semantic relatedness using Wikipedia-based explicit semantic analysis[C]//Proceedings of the International Joint Conference on Artificial Intelligence, 2007: 1606-1611.

[40] MILNE D, WITTEN I H. An effective, low-cost measure of semantic relatedness obtained from Wikipedia Links[C]//Proceedings of the 23rd Association for the Advancement of Artificial Intelligence, 2008: 25-30.

[41] CAMACHO-COLLADOS J, PILEHVAR M T, NAVIGLI R. Nasari: a novel approach to a semantically-aware representation of items[C]//Proceedings of the Conference on the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2015: 567-577.

[42] QU R, FANG Y, BAI W, et al. Computing semantic similarity based on novel models of semantic representation using Wikipedia[J]. Information Processing & Management, 2018, 54(6): 1002-1021.

[43] WU Z, ZHU H, LI G, et al. An efficient Wikipedia semantic matching approach to text document classification [J]. Information Sciences, 2017, 393: 15-28.

[44] ZHAN Z J, LIANG L N, YANG X P. Word similarity calculation based on Baidu Encyclopedia[J]. Computer Science, 2013, 40(6): 199-202.
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.

[45] YIN K, YIN H F, YANG Y, et al. Semantic similarity calculation of Baidu encyclopedia entries based on SimRank[J]. Journal of Shandong Uni-

versity (Engineering Science),2014,44(3): 29-35.
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:工学版,2014,44(3): 29-35.
[46] MIKOLOV T,CHEN K, CORRADO G,et al. Efficient estimation of word representations in vector space [DB/OL]. https://arxiv. org/abs/1301. 3781,2013.
[47] PENNINGTON J, SOCHER R, MANNING C D. Glove: global vectors for word representation[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing,2014:1532-1543.
[48] JOULIN A, GRAVE E, BOJANOWSKI P, et al. Bag of tricks for efficient text classification[C]//Proceedings of the Conference on the Europan Chapter of the Association for Computational Linguistics, 2017: 427-431.
[49] LE Q, MIKOLOV T. Distributed representations of sentences and documents [C]//Proceedings of the International Conference on Machine Learning,2014: 1188-1196.
[50] KUSNER M, SUN Y, KOLKIN N, et al. From word embeddings to document distances[C]//Proceedings of the International Conference on Machine Learning,2015: 957-966.
[51] ARORA S,LIANG Y,MA T. A simple but tough-to-beat baseline for sentence embeddings[C]//Toulon:International Conference on Learning Representations,2017.
[52] KIROS R,ZHU Y,SALAKHUTDINOV R R,et al. Skip-thought vectors [C]//Proceedings of the Advances in Neural Information Processing Systems,2015:3294-3302.
[53] HILL F,CHO K,KORHONEN A. Learning distributed representations of sentences from unlabelled data [DB/OL]. https://arxiv. org/abs/1602. 03483,2016.
[54] LOGESWARAN L, LEE H. An efficient framework for learning sentence representations[DB/OL]. https://arxiv. org/abs/1803. 02893,2018.
[55] HUANG P S, HE X, GAO J, et al. Learning deep structured semantic models for web search using clickthrough data[C]//Proceedings of the 22nd ACM International Conference on Information & Knowledge Management,2013: 2333-2338.
[56] SHEN Y, HE X, GAO J, et al. A latent semantic model with convolutional-pooling structure for information retrieval[C]//Proceedings of the 23rd ACM International Conference on Information and Knowledge Management,2014: 101-110.
[57] PALANGI H, DENG L, SHEN Y, et al. Semantic modelling with long-short-term memory for information retrieval[DB/OL]. https://arxiv. org/abs/1412. 6629v1,2014.
[58] BAO W,BAO W,DU J,et al. Attentive siamese LSTM network for semantic textual similarity measure[C]//Proceedings of the International Conference on Asian Language Processing,2018: 312-317.
[59] YUAN Y L, ZHANG G Y . High school math text similarity studies based on CNN and BiLSTM[C]//2020 5th International Conference on Mechanical, Control and Computer Engineering,2020:1982-1986.
[60] HU B, LU Z, LI H, et al. Convolutional neural network architectures for matching natural language sentences[J]. Advances in Neural Information Processing Systems,2014,27:2042-2050.
[61] ZHANG S, XU X, TAO Y, et al. Text similarity measurement method based on BiLSTM-SECapsNet model[C]//2021 6th International Conference on Image,Vision and Computing,2021: 414-419.
[62] PARIKH A P,TÄCKSTRÖM O,DAS D,et al. A decomposable attention model for natural language inference [DB/OL]. https://arxiv. org/abs/1606. 01933v1,2016.
[63] CHEN Q,ZHU X,LING Z,et al. Enhanced LSTM for natural language inference[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics,2017:1657-1668.
[64] ZHOU X,LI L,DONG D,et al. Multi-turn response selection for chatbots with deep attention matching network[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics,2018: 1118-1127.
[65] GRAVES A. Long short-term memory[M]. Berlin: Springer, 2012: 37-45.
[66] HUANG Z H, XU W, YU K. Bidirectional LSTM-CRF models for sequence tagging[DB/OL]. https://arxiv. org/abs/1508. 01991,2015.
[67] SABOUR S, FROSST N, HINTON G E. Dynamic routing between capsules[DB/OL]. https://arxiv. org/abs/1710. 09829,2017.
[68] HU J,SHEN L,SUN G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,2018:7132-7141.
[69] VASWANI A,SHAZEER N,PARMAR N, et al. Attention is all you need [C]//Proceedings of the 31st International Conference on Neural Information Processing Systems,2017: 5998-6008.
[70] PETERS M, NEUMANN M, IYYER M, et al. Deep contextualized word representations[C]//2018 Conference of the North American Chapter of the Association for Computational Linguistics,2018:2227-2237.
[71] DEVLIN J, CHANG M W, LEE K, et al. Bert: pre-training of deep bidirectional transformers for language understanding[DB/OL]. https://arxiv. org/abs/1810. 04805v2,2018.
[72] LIU Y,OTT M,GOYAL N,et al. Roberta: a robustly optimized bert pretraining approach[DB/OL]. https://arxiv. org/abs/1907. 11692,2019.
[73] REIMERS N, GUREVYCH I. Sentence-bert: sentence embeddings using siamese bert-networks[DB/OL]. https://arxiv. org/abs/1908. 10084, 2019.
[74] LI B, ZHOU H, HE J, et al. On the sentence embeddings from pre-trained language models [DB/OL]. https://arxiv. org/abs/2011. 05864, 2020.
[75] SU J, CAO J, LIU W, et al. Whitening sentence representations for better semantics and faster retrieval [DB/OL]. https://arxiv. org/abs/2103. 15316,2021.
[76] GAO T, YAO X, CHEN D. Simcse: simple contrastive learning of sentence embeddings[DB/OL]. https://arxiv. org/abs/2104. 08821,2021.
[77] WU X,GAO C,ZANG L,et al. Esimcse: enhanced sample building method for contrastive learning of unsupervised sentence embedding[C]//Proceedings of the 29th International Conference on Computational Linguistics,2022:3898-3907.
[78] YAN Y,LI R,WANG S,et al. Consert: a contrastive framework for self-supervised sentence representation transfer[DB/OL]. https://arxiv. org/abs/2105. 11741,2021.
[79] CHUANG Y S, DANGOVSKI R, LUO H, et al. DiffCSE: difference-based contrastive learning for sentence embeddings[DB/OL]. https://arxiv. org/abs/2204. 10298,2022.
[80] ZHOU K,ZHANG B,ZHAO W X,et al. Debiased contrastive learning of unsupervised sentence representations [DB/OL]. https://arxiv. org/abs/2205. 00656v1,2022.
[81] LEE S,LEE D,JANG S,et al. Toward interpretable semantic textual similarity via optimal transport-based contrastive sentence learning[DB/OL]. https://arxiv. org/abs/2202. 13196,2022.
[82] JIANG T, JIAO J, HUANG S, et al. Promptbert: improving bert sentence embeddings with prompts [DB/OL]. https://arxiv. org/abs/2201. 04337,2022.
[83] ZENG J, YIN Y, JIANG Y, et al. Contrastive learning with prompt-derived virtual semantic prototypes for unsupervised sentence embedding [DB/OL]. https://arxiv. org/abs/2211. 03348,2022.

(责任编辑:孙　娟)