文章编号: 1000-5641(2020)05-0095-18

语义文本相似度计算方法

韩程程, 李 磊, 刘婷婷, 高 明

(华东师范大学 数据科学与工程学院, 上海 200062)

摘要: 综述了语义文本相似度计算的最新研究进展, 主要包括基于字符串、基于统计、基于知识库和基于深度学习的方法. 针对每一类方法, 不仅介绍了其中典型的模型和方法, 而且深入探讨了各类方法的优缺点; 并对该领域的常用公开数据集和评估指标进行了整理, 最后讨论并总结了该领域未来可能的研究方向.

关键词: 文本相似度; 语义相似度; 自然语言处理; 知识库; 深度学习

中图分类号: TP311 **文献标志码**: A **DOI**: 10.3969/j.issn.1000-5641.202091011

Approaches for semantic textual similarity

HAN Chengcheng, LI Lei, LIU Tingting, GAO Ming

(*School of Data Science and Engineering, East China Normal University, Shanghai* 200062, *China*)

Abstract: This paper summarizes the latest research progress on semantic textual similarity calculation methods, including string-based, statistics-based, knowledge-based, and deep-learning-based methods. For each method, the paper reviews not only typical models and approaches, but also discusses the respective advantages and disadvantages of each routine; the paper also explores public datasets and evaluation metrics commonly used. Finally, we put forward several possible directions for future research in the field of semantic textual similarity.

Keywords: textual similarity; semantic similarity; natural language processing; knowledge base; deep learning

0 引 言

随着互联网技术的迅速发展, 海量信息在互联网中不断涌现, 文本是信息最重要的载体. 研究文本信息的深度挖掘, 对于人们快速而准确地获取所需内容具有重要意义. 语义文本相似度计算 (Semantic Textual Similarity) 是联系文本表示和上层应用之间的纽带. 目前, 在信息检索 (Information Retrieval) 领域, 语义文本相似度计算在文本分类[1](Text Classification)、文本聚类[2](Text Clustering)、实体消歧[3](Entity Disambiguation) 等任务上有着极其重要的作用; 在人工智能领域, 问答系统[4](Q/A System) 和智能检索[5](Intelligent Retrieval) 等任务也都需要语义文本相似度算法作为支撑. 此外, 语义文本相似度计算也广泛应用在抄袭检测[6](Plagiarism Detection)、文本摘要[7](Text Summarization)、

收稿日期: 2020-08-09

基金项目: 国家重点研发计划(2016YFB1000905); 国家自然科学基金(U1911203, U1811264, 61877018, 61672234, 61672384); 中央高校基本科研业务费专项资金; 上海市科技兴农推广项目(T20170303); 上海市核心数学与实践重点实验室资助项目(18dz2271000)

通信作者: 高 明, 男, 教授, 博士生导师, 研究方向为教育计算、知识图谱、知识工程、用户画像、社会网络挖掘、不确定数据管理. E-mail: mgao@dase.ecnu.edu.cn

机器翻译[8](Machine Translation) 等自然语言处理 (Natural Language Processing) 任务中. 因此, 系统化地研究语义文本相似度算法具有非常重要的应用价值.

给定两段文本 A 和 B, 语义文本相似度计算旨在衡量两段文本在语义上的相近程度. 通常, 文本的语义相似度数值越小, 则说明两个文本之间的语义差异性越大, 即在语义层面上越不相似; 反之, 该数值越大, 则说明这两个文本所表达出的语义越相似. 由于人类语言表达十分复杂, 文本当中包含许多同义词、缩略词、否定词等, 还有多变的句法结构, 都加大了语义文本相似度计算的难度. 为了解决这些难题, 学术界和工业界都进行了大量的研究和实践, 提出了一系列针对语义文本相似度计算问题的模型和方法.

由于语义文本相似度计算一直以来都是自然语言处理领域的热点问题之一, 因此国内外已经有一些学者对已有的语义文本相似度计算方法进行了系统整理, 其中不乏一些优秀的综述论文[9-10], 但他们的工作都主要集中在传统的基于统计和基于知识库的方法上. 而近年来, 伴随深度学习技术的快速发展, 特别是 2013 年分布式词向量问世后, 语义文本相似度计算领域取得了突破性的进展, 基于深度学习的语义文本相似度计算方法已经逐渐成为该领域的主流方法. 与已有的综述文献不同, 本文在总结传统方法的基础上, 特别聚焦在语义文本相似度计算的最新进展上.

本文对现有的语义文本相似度计算方法进行了系统的综述, 总体分类框架如图 1 所示. 本文将分小节依次对每一类方法进行系统的介绍. 本文的第 1 章主要介绍基于字符串的语义文本相似度算法; 第 2 章主要介绍基于统计的语义文本相似度算法; 第 3 章主要介绍基于知识库的语义文本相似度算法; 第 4 章主要介绍基于深度学习的语义文本相似度算法; 第 5 章主要介绍语义文本相似度领域相关数据集及评价指标; 最后, 讨论语义文本相似度计算的未来研究方向, 并对全文进行总结.

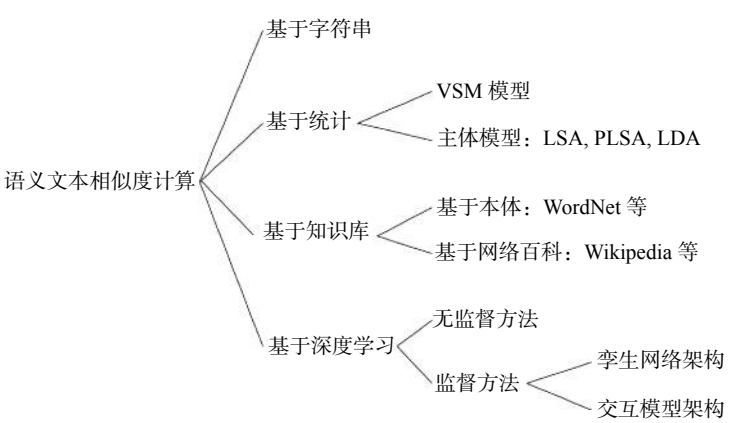

图 1 语义文本相似度计算研究分类

Fig. 1 Classification of semantic textual similarity

1 基于字符串的语义文本相似度计算

基于字符串的方法都是直接对原始文本进行比较, 主要包括编辑距离[11](Levenshtein Distance, LD)、最长公共子序列[12](Longest Common Sequence, LCS)、N-Gram[13] 和 Jaccard 相似度[14] 等.

编辑距离通常被用于句子的快速模糊匹配领域, 以表示两个文本之间, 由一个转换成另一个所需的最少编辑操作次数, 其中编辑操作包括增、删、改三种, 它既可以是字符级别, 也可以是单词级别. 车万翔等[15] 利用大规模知识库赋予不同编辑操作不同的权重, 提出了改进的编辑距离计算方法, 同时考虑了词序和部分语义信息, 提高了编辑距离的算法性能.

最长公共子序列 (LCS) 算法[12] 以两个文本的最长公共子文本的长度来表征文本间的相似度. 一

个序列 S 任意删除若干个字符得到的新序列 T, 则 T 叫作 S 的子序列. 两个序列 X 和 Y 的公共子序列中, 长度最长的被定义为 X 和 Y 的最长公共子序列 (LCS), 可以使用动态规划的方法计算两个文本的最长公共子序列以表征两个文本的相似度.

N-Gram 模型[13] 的基本思想是设置大小为 N 的滑动窗口, 将文本内容按照字符流或者单词流的形式进行窗口滑动操作, 形成多个长度为 N 的文本片段, 每个片段被称为一个 N 元组, 然后计算给定的两个文本中公共 N 元组的数量与总 N 元组数量的比值, 以此来表征两个文本的相似度.

Jaccard 系数[14] 是两个集合的交集与并集中包含的元素个数之比, 它仅关注两个集合中共有的元素个数, 而不关注集合元素之间的差异性, 我们可以将文本看成由其中的单词组成的集合, 此时单词即为集合元素, 也可以将文本所包含的 N 元组作为集合元素, 然后计算两个集合之间的 Jaccard 系数来表征两个文本间的相似度. Jaccard 系数还可以与局部敏感哈希 (Locality Sensitive Hashing, LSH)[16] 相结合进行快速的近似文本查找, 或对不相似的文本进行过滤.

Dice 系数与 Jaccard 系数[14] 类似, 都是基于集合的思想. Dice 系数是两个集合的交集中包含的元素个数与两个集合的长度之和的比值, 主要关注的是集合中相同的部分, 取值范围在 (0, 1) 之间, 该值越接近于 1, 则说明两个字符串越相似. 表 1 展示了基于字符串的文本相似度计算方法的总结.

表 1　基于字符串的语义文本相似度计算方法

Tab. 1　String-based method for semantic textual similarity

类型	计算方法		基本思想
基于字符串	编辑距离[11,16]		文本 S_A 转换到文本 S_B 所需的最少编辑操作次数. 编辑操作包括: 增、删、改.
	LCS[12]	$\frac{2K}{L_A+L_B}$	其中 K 表示 S_A 与 S_B 的最长公共子序列的长度(可用动态规划求解). L_A 与 L_B 分别表示 S_A 与 S_B 的长度.
	N-Gram[13]	$\frac{N_1}{N_2}$	其中 N_1 表示文本 S_A 与 S_B 共有的 N 元组数量, N_2 表示总的 N 元组数量.
	Jaccard[15]	$\frac{A \bigcap B}{A \bigcup B}$	其中 A 与 B 分别为表征 S_A 与 S_B 的集合, 集合中的元素可以是字符、单词、N 元组等
	Dice系数	$\frac{2\lvert A \bigcap B\rvert}{\lvert A\rvert+\lvert B\rvert}$	其中 A 与 B 分别表示文本 S_A 与 S_B 的子集合, 分子表示 A 与 B 交集个数的两倍, 分母为 A 与 B 集合中包含的元素个数之和

基于字符串的方法原理简单、实现方便, 并且直接对原始文本进行比较, 多用于文本的快速模糊匹配, 其不足主要在于没有考虑到单词的含义及单词和单词之间的相互关系, 并且同义词、多义词等问题都无法处理. 目前很少单独使用基于字符串的方法计算文本相似度, 而是将这些方法的计算结果作为表征文本的特征融入更加复杂的方法中.

2　基于统计的语义文本相似度计算方法

基于统计的方法源于一种分布假设, 该假设认为上下文相似的单词具有相似的语义, 这类计算方法先通过某种策略将文本转换成一个向量, 然后通过各种向量空间的转换, 最后计算表征文本的向量间距离, 通过向量空间中的度量来衡量文本间的相似度. 主流的基于统计的方法包括向量空间模型[17] (Vector Space Model, VSM) 和主题模型 (Topic Model), 而主题模型又可分为潜在语义分析模型[18](Latent Semantic Analysis, LSA)、概率潜在语义分析模型[19](Probabilistic Latent Semantic Analysis, PLSA) 和隐含狄利克雷分布模型[20](Latent Dirichlet Allocation, LDA) 等.

2.1　基于向量空间模型 (VSM) 的计算方法

Salton 等[17] 在 1975 年首次提出向量空间模型 (VSM), 由于该模型简单高效, 在之后很长的一段时间里, 它都是文本相似度计算领域的主流方法. VSM 的主要思想就是假设一个文本的语义只与该

文本中的单词有关, 而忽略其语序和单词之间的相互关系, 然后通过基于词频统计的方法, 将文本映射成向量, 最后通过向量间的距离计算以表征文本间的相似度.

在 VSM 中, 将单词作为文本向量的特征项, 其中特征项的权重可以用单词在该文本中出现的次数表示, 但这样做会导致一些没有实际含义的单词如“is”“are”等的权重变大, 进而严重影响文本相似度的计算. 因此, 目前 VSM 中最常用的是基于 TF-IDF 的权重计算法, 这种方法将特征项的权重表示为词频 (TF) 和逆文本频率 (IDF) 的乘积, 词频 (TF) 可以通过下式进行计算:

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}},$$

其中 i 表示单词索引, j 表示文本索引, $n_{i,j}$ 表示第 i 个单词在第 j 个文本中出现的次数, 分母表示第 j 个文本中的总单词数. 可以看到, TF 值就是某个单词在一个文本中出现的频次与该文本的单词总数的比值. 逆文本频率 (IDF) 可以通过下式进行计算:

$$IDF_i = \log\frac{M}{m_i + \alpha},$$

其中 M 表示总的文本个数, m_i 表示共有 m_i 个文本中包含了第 i 个单词, α 表示经验系数, 一般取 0.01, 目的是防止分母为 0. 这样可以通过下式计算出每个文本向量的特征项对应的 TF-IDF 权重:

$$TF\text{-}IDF_{i,j} = TF_{i,j} \cdot IDF_i.$$

从上式可以看出, 当一个单词在单一文本中出现的频次很高, 而很少出现在其他文本中时, 则这个单词的 TF-IDF 值就会很大, TF-IDF 的主要思想就是认为这类单词具有更好的类别区分能力, 因此给予这类单词更大的权重.

在利用 TF-IDF 权重计算法计算出各个特征项的权重之后, 就得到了可以表征文本的向量, 接下来只要计算向量之间的距离即可, 一般来说, 距离越近则两文本越相似. 文本相似度领域中常用的距离计算包括余弦相似度 (Cosine Similarity)、欧式距离 (Euclidean Distance)、曼哈顿距离 (Manhattan Distance)、切比雪夫距离 (Chebyshev Distance) 等, 我们还可以将文本向量看成不同的多维变量, 因此可以使用统计相关系数进行相似度计算, 如皮尔逊 (Pearson) 和斯皮尔曼 (Spearman) 相关系数等.

Qinglin 等[21] 在进行 TF-IDF 加权计算前, 首先通过计算特征项之间的信息增益、卡方检验和互信息等方法进行特征选择, 然后在此基础上利用 VSM 计算语义文本相似度, 提高了算法性能. Li 等[22] 指出传统的 VSM 方法没有对文本间相同特征词的个数进行统计, 这样会导致某些情况下计算结果会变差, 因此在 VSM 基础上增加了文本间相同特征词的统计. Tasi[23] 将最长公共子序列 (LCS) 和 VSM 相结合, 首先对文本进行基于 LCS 的最优匹配, 根据匹配结果赋予关键词不同的权重, 在此基础上在进行向量空间模型计算, 并且将余弦相似度替换为计算 Jaccard 系数, 进一步提升了算法性能.

尽管有很多研究对 VSM 方法进行了改进, 但是基于 VSM 的方法仍然有两点缺陷: ① 当文本量很大时, 生成的文本向量是非常稀疏的, 这就导致了空间和计算资源的浪费; ② VSM 为达到简化模型的效果忽略了词语间的关系, 但在很多情况下词语之间是存在联系的, 因此简单地认为词语间相互独立是不合理的. 随着近几年深度学习方法的迅速发展, VSM 模型的研究热度逐渐退去, 但在深度学习方法中依然有 VSM 的思想贯穿其中.

2.2 基于主题模型 (Topic Model) 的计算方法

主题模型是机器学习和自然语言处理领域的经典方法, 尽管目前在语义文本相似度计算领域中, 深度学习方法已经占据主导地位, 但主题模型的作用不容忽视. 主题模型的基本假设是每个文档包含多个主题, 而每个主题又包含多个单词. 换句话说, 文档的语义由一些隐变量表示, 这里的隐变量是指

主题, 而这些主题代表了文档的语义信息. 而主题模型的目的就是揭示这些隐变量和它们之间的相互关系.

主题模型主要包括: ① 潜在语义分析 (LSA) 模型; ② 概率潜在语义分析 (PLSA) 模型; ③ 潜在狄利克雷分布 (LDA) 模型.

LSA[18] 模型是基于 VSM 模型而提出的, 其基本思想是在得到文本的空间向量表示后, 通过奇异值分解 (SVD), 将高维且稀疏的空间向量映射到低维的潜在语义空间当中, 在得到低维的文本向量和单词向量之后, 再用余弦相似度等度量方式来计算文本间的语义相似度. LSA 的本质思想就是通过降维来去除原始矩阵中的噪音, 从而提高计算准确度, 但由于使用 SVD 分解, 使得计算复杂度增高, 并且可移植性较差.

Hofmann[19] 用概率方法代替 SVD 分解, 提出了概率潜在语义分析 (PLSA) 模型, 其核心思想是加入主题层, 采用期望最大化算法 (Expectation Maximization, EM) 训练主题, 找到一个潜在主题的概率模型, 该模型用于预测文本空间向量中观察到的数据. 在 PLSA 模型中, 多义词会被分到不同的主题下, 而同义词会被分到同一主题下, 从而避免了同义词和多义词对文本相似度计算的影响. 但是 PLSA 的模型参数会随着文档数量线性增长, 因而容易出现过拟合和泛化能力差等问题.

Blei 等[20] 在 PLSA 模型的基础上提出了潜在狄利克雷分布 (LDA) 模型, 可以将其视作 PLSA 的贝叶斯版本. 该模型是一个三层贝叶斯概率模型, 包含词、主题和文档三层结构, 并且使用狄利克雷分布来处理文档-主题分布和单词-主题分布. 在 PLSA 中, 在给定数据集的情况下文档概率是固定的, 如果某个文档没有出现过就无法获取该文档的概率值. 而在 LDA 中, 数据集作为训练数据用于训练文档-主题分布的狄利克雷分布. 即使没有看到某个文件, 仍然可以从狄利克雷分布中抽样得到, 从而提高模型的泛化能力.

王振振等[24] 对语料库进行 LDA 主题建模, 在建模过程中使用 Gibbs 采样对模型参数进行估计, 在得到文本的主题分布后, 通过使用 KL(Kullback-Leibler) 距离进行文本相似度计算, 这也是目前比较常见的基于 LDA 的语义文本相似度计算方法.

Daping 等[25] 将 LDA 与 VSM 以及基于知识库的方法相结合, 分别计算出主题相似度、统计相似度及语义相似度, 最终通过线性加权得到最终的相似度得分. Chao 等[26] 将词性标注与 LDA 相结合, 先对单词进行词性标注, 并按照单词词性将其分为名词、动词和其他词, 然后分别进行 LDA 主题建模, 最终通过不同权重综合 3 种模型进行文本相似度计算, 由于考虑到了词性对文本相似度的贡献差异, 因而提高了文本聚类的准确率.

Miao 等[27] 将分布式词向量和 LDA 相结合提出 lda2vec 模型, 在训练词向量的同时训练文档向量和主题向量, 在进行相似度计算时可以将 3 种向量相结合, 以提高算法性能. Lau 等[28] 将语言模型和 LDA 主题模型相结合, 使用分布式词向量而非 TF-IDF 权值将文本映射为向量, 并且对原始的文本向量组成的矩阵进行卷积, 最终通过 Attention 机制产生主题向量. 实验结果证明了其性能优于原始 LDA 模型.

3　基于知识库的语义文本相似度计算方法

基于知识库的语义文本相似度计算方法根据知识库的类型可以分为两大类, 一类是基于本体的方法, 这类方法运用结构化语义词典进行计算, 其基本思想就是运用这些语义词典中包含的概念信息和概念间的层次关系进行语义文本相似度计算. 另一类是基于网络知识的方法, 这类方法主要利用网络大型知识库资源, 如维基百科 (Wikipedia) 和百度百科等, 通过网页内容和网页间的超链接进行相似度计算.

3.1 基于本体的计算方法

在语义文本相似度计算中, 本体是指对特定领域的概念进行抽象化和结构化的描述, 通用本体主要包括一些概念语义词典, 常见的语义词典有 WordNet[29]、《同义词词林》[30]、《知网》(HowNet)[31] 等.

WordNet[29] 是一种英文的语义词典, 不仅包括单词的概念解释、词性信息等, 还包含了多种语义关系, 如同义关系 (Synonymy)、反义关系 (Antonymy)、上位/下位关系 (Hypernymy&Hyponymy)、部分整体关系 (Meronymy) 等, 并且通过概念树的结构展示各个概念之间的关系.《同义词词林》[30] 是一种结构上与 WordNet 十分相似的中文语义词典, 它将所有的概念分为大类、中类、小类、词群、原子词群, 并通过该分层建立概念树.

《知网》(HowNet)[31] 中包含中文和英文两种语言, 构建方式与上述两种语义词典有所不同. 它将词语分解为多个概念, 再将概念分解为多个义原. 义原是最小的不可分的语义单位, 所有概念的语义都可使用一个有限的义原集合去表示. 义原之间存在多种关系, 如上下位关系、同义关系、反义关系等, 通过这些关系可以构成一个树状的义原层次结构, 再根据这个结构进行语义文本相似度计算.

3.1.1 基于 WordNet 和《同义词词林》的计算方法

由于上述两种语义词典的构建方式较为相似, 很多方法可以互用, 因此放在一起进行介绍. 这类方法主要分为 4 类: ①基于距离的方法; ②基于信息量 IC(Information Content) 的方法; ③基于属性的方法; ④混合方法. 表 2 对相关方法进行分类并总结了各类方法的特点.

表 2 基于 WordNet 和同义词词林的计算方法

Tab. 2 Methods based on WordNet and the synonymy thesaurus

分类	相关方法	特点
基于距离	Rada等[32]、Richardon等[33]、Leacock等[34]、Wu等[35]、Hirst等[36]、Yang等[37]	利用概念结构树计算概念间距离, 并结合深度、层次信息计算语义文本相似度
基于信息量	Reshik[38]、Jiang等[39]、Lin等[40]	通过概念包含的信息量进行语义文本相似度计算, 信息量如何定义是该类方法改进的本质
基于属性	Lesk[41]、Banerjee等[42]、Pedersen等[43]	利用概念的释义信息及类型信息计算语义文本相似度
混合式	Li等[44]、Bin等[45]、郑志蕴等[46]	结合上述三类方法, 一般该类方法的算法复杂度较高

基于距离的方法是指利用语义词典中的概念结构树的层次关系来计算语义文本相似度, 两个概念节点在概念层次树中的路径越长, 相似度越低.

Rada 等[32] 假设所有边的权重相同, 通过求出两个概念间的最短路径 (Shortest Path) 以表征它们之间的语义文本相似度. Richardson 等[33] 在 Shortest Path 方法的基础上, 通过对概念的位置信息进行分析, 为概念结构树的边加上权重信息. Leacock 等[34] 除了计算最短距离之外, 还考虑了概念在语义词典层次结构中的深度. Wu 等[35] 则提出不直接计算概念间的距离, 而是计算两个概念与其公共父节点之间的距离以表征其语义文本相似度. Hirst 等[36] 提出在进行距离计算的同时加入转向因子, 该方法认为除了考虑距离问题之外, 路径中的方向转变越少, 则两概念的语义文本越相似. Yang 等[37] 充分利用 WordNet 中的同义关系和部分整体关系设计了两种更为复杂的搜索算法用来衡量单词对之间的语义文本相似度, 实验证明该算法的性能良好, 但该算法包含多个超参数, 增加了算法的不确定性.

基于信息量的方法是指利用概念所包含的信息量来衡量概念间的相似度, 概念对之间的共享信息量越大, 则说明它们越相似. 不同学者对于衡量信息量的方法的想法各有不同.

Resnik[38] 提出利用概念对的公共父节点在语义词典中出现的频率来衡量概念对的共享信息量; Jiang 等[39] 利用概念对本身信息量和公共父节点信息量的差值表示概念间的距离; Lin 等[40] 和

Jiang 等[39] 的想法类似, 两者定义的语义文本相似度计算公式稍有不同, 之后很多学者尝试改进基于信息量的方法, 改进的本质都是定义新的信息量计算方法和新的基于信息量的语义文本相似度的计算公式.

基于属性的方法是指利用概念的属性信息进行相似度计算, 属性一般指概念的释义信息和类型信息.

Lesk[41] 首先提出统计概念释义中的共现词语的数量以衡量语义文本相似度; Banerjee 等[42] 在 Lesk[41] 的基础上进行了改进, 在统计单词概念释义的同时考虑了其同义词的概念释义信息; Pedersen 等[43] 提出将概念的释义信息通过语义词典的层次结构转化为释义向量, 然后计算概念对的释义向量间的余弦值表征概念的语义相似度.

混合方法一般指结合上述多种方法, 综合考虑不同因素对于相似度计算的影响, 为各因素赋予不同的权重, 然后进行加权求和得到最终的语义文本相似度. Li 等[44] 将概念间的最短路径、公共父节点在结构树中的深度及概念的局部密度信息相结合; Bin 等[45] 主要基于信息量, 同时融合路径长度、概念深度等信息, 提出混合式相似度计算方法; 郑志蕴等[46] 结合基于距离、基于信息量及基于属性的方法, 采用主成分分析法, 提出一种自适应相似度加权计算方法以解决传统人工赋权的不足.

3.1.2　基于《知网》(HowNet) 的计算方法

《知网》用概念描述词汇语义, 再用义原描述概念语义. 义原是描述概念的最小单位, 同时《知网》中也标注了义原之间的各种关系, 其中最重要的是上下位关系, 通过上下位关系, 可以将义原组织为一个树状层次结构, 这是利用《知网》计算语义文本相似度的基础. 知网中的实词概念主要是由第一基本义原、其他基本义原、关系义原、关系符号 4 个部分进行描述. 刘群等[47] 提出将概念按照描述中所包含的义原类别分为几个部分, 每个部分分别通过义原结构树计算概念语义相似度, 最终加权求和得到最终结果; 李峰等[48] 在刘群的基础上引入信息量的概念, 认为越处于底层的义原节点包含的信息量越大, 并以此对算法进行了改进; 类似地, 江敏等[49] 在刘群的基础上加入了义原的深度信息, 并且对义原的反义关系的处理采用了全新的方式, 该方法应用于情感分析任务时算法效果明显提升.

基于结构化语义词典的方法确实在一定程度上考虑到了词语之间的语义信息, 但是有以下缺点: ① 人工成本高. 需要专家参与建立语义词典, 并且需要不断地更新新出现的词汇以及部分词汇之间的关系. ② 领域本体, 即特定领域的语义词典, 容易出现异构问题, 导致算法不通用. ③ 该类方法都是计算词语之间的相似度, 句子相似度的计算则是简单地对词语相似度进行加权求和, 并没有考虑到句法结构信息, 对长文本的相似度计算准确率较低.

3.2　基于网络知识的计算方法

随着互联网的快速发展, 网络知识愈加丰富, 如何充分利用网络中的资源进行语义文本相似度计算非常值得研究. 由于网页中知识颗粒度较粗, 加之部分网页的知识结构化程度较低, 如果直接对所有的网页链接进行分析, 会导致知识含量稀疏、计算困难等问题. 维基百科作为全球最大的多语种、开源的在线百科全书, 知识更新速度更快, 知识结构化程度更高, 因此, 挖掘与利用维基百科中的信息资源成为了研究热点. 我们可以将维基百科看成由网页内容及其包含的超链接组成的大型网络, 其中, 将超链接看成边, 网页内容看成节点, 则可以将其抽象为有向图模型. 表 3 对基于网络知识的方法进行了简要的分类并总结了各类方法的特点.

Strube 等[50] 提出的 WikiRelate!方法是最早的探索用 Wikipedia 进行语义文本相似度计算的方法之一, 计算方法和基于本体中的方法类似, 包括基于距离和基于信息量的方法等, 在实验中证明了其结果可以优于部分基于 WordNet 的方法.

表 3 基于网络知识的计算方法

Tab. 3 Methods based on network knowledge

分类	相关方法	特点
基于本体	Strube等[50](WikiRelate!)	将基于本体的方法(如基于距离和基于信息量的方法)迁移至维基百科上进行语义文本相似度计算
基于向量空间	Gabrilovich等[51](ESA)、Witten等[52]、Yeh等[53]、Camacho-Collados等[54](NASARI)	将维基百科中的网页内容映射为高维向量,再通过基于向量空间的方法进行语义文本相似度计算

Gabrilovich 等[51] 提出显式语言分析 (Explicit Semantic Analysis, ESA) 方法, 该方法通过 Wikipedia 中的网页内容将文本映射为一个高维向量, 再通过基于向量空间的方法进行语义文本相似度计算; Witten 等[52] 首次提出利用 Wikipedia 中的超链接结合向量空间的方法进行语义文本相似度计算, 由于该方法仅使用链接信息, 几乎不用文本信息, 所以虽然方法简单, 但准确性较差; Yeh 等[53] 则是在 ESA 算法[51] 的基础上, 加入个性化的 PageRank 算法, 在提高性能的同时增加了计算量; Camacho-Collados 等[54] 在 ESA 的基础上提出了 NASARI 方法, 该方法额外加上了 WordNet 的知识信息, 得到了更有效的文本向量表示, 对比 ESA 算法效果明显提升.

基于网络知识的方法具有知识信息丰富、知识迭代速度快等优点, 但是相比于专家编制的语义词典, 维基百科的结构化程度较差, 其中也不免有一些错误的信息, 整体的分类也更加粗糙, 这导致目前基于网络知识的方法效果并不好, 如何更好地在半结构化数据中提取更多有用的信息还有待进一步研究.

4 基于深度学习的语义文本相似度计算方法

自 2013 年分布式词向量问世以来, 基于深度学习的方法在语义文本相似度领域涌现了许多杰出的工作, 目前效果最好的几种模型都是基于深度学习的方法实现的, 因此, 这类方法在相似度计算领域具有十分重要的意义.

目前, 基于深度学习的语义文本相似度计算方法可以分为两大类——无监督方法和监督方法. 无论是无监督还是监督方法, 都需要在分布式词向量的基础上展开. 因此, 4.1 节简要介绍词向量相关内容, 之后两节将分别介绍无监督方法和监督方法.

4.1 分布式词向量

简单地说, 词向量技术就是将单词映射成向量, 最早出现的 one-hot 编码和 TF-IDF 方法都可以将单词映射为向量. 但是, 这两种方法都面临维度灾难和语义鸿沟问题. 分布式词向量可以在保存更多语义信息的同时降低向量维度, 在一定程度上可以克服维度灾难和语义鸿沟问题.

Mikolov 等[55] 提出的 word2vec 是最早生成分布式词向量的方法, 它包含两个模型, 分别为 CBOW 和 Skip-gram, 基本思路是确定中心词和上下文窗口大小, CBOW 是通过上下文来预测中心词, Skip-gram 是通过中心词来预测上下文, 整体来说是通过自监督训练的模型生成词向量. word2vec 的主要问题在于它只能考虑局部信息, 局部信息的大小取决于上下文窗口的大小.

Pennington 等[56] 提出 Glove 模型, 该模型通过语料库构建单词的共现矩阵, 然后通过该共现矩阵用概率的思想得到最终的词向量, 综合了全局语料, 在一定程度上考虑了全局信息.

Joulin 等[57] 则是提出了一种快速文本分类方法 FastText, 其同样可以用于生成词向量, 模型架构与 CBOW 类似, 但赋予模型的任务不同.

以上 3 种词向量为静态词向量, 即当它们应用于下游任务时词向量始终是固定的, 无法解决一词多义问题, 同时无法满足不同语境下语义不同的场景需求.

动态词向量则是为了解决上述问题所提出的, 这类词向量首先在大型语料库上进行预训练, 然后

在面对具体下游任务时微调所有参数, 那么在上下文输入不同时, 生成的词向量也不同, 因此可以解决一词多义问题.

Peters 等[58] 提出 ELMO 模型, 其使用双向语言模型和两个分开的双层 LSTM 作为编码器, 通过在大型语料库上进行预训练得到动态词向量.

Radford 等[59] 提出 GPT 模型, 通过将单向语言模型与编码能力更强大的 Transformer[60] 架构的 decoder 部分相结合, 从而生成词向量.

Devlin 等[61] 提出的 BERT 模型, 则是应用 Transformer[60] 的 encoder 部分, 结合 mask 机制, 并且对模型增加了预测“next sentence prediction”任务, 从而生成了更加优质的动态词向量, 该模型也是目前最常用的词向量预训练方法之一.

4.2　基于无监督学习方法的语义文本相似度计算

无监督学习方法不需要带有标签的数据集就可以计算文本间的语义相似度, 这类方法更加通用, 在一些资源稀少的特定领域应用广泛. 这类方法的基本思路是, 通过数据集本身带有的信息进行自监督训练或通过数学分析的方法, 对句子中的词向量进行加权求和得到句向量, 并最终通过计算向量间距离以表征语义文本相似度. 表 4 对基于无监督学习的计算方法进行了简要的分类并总结了各类方法的特点.

表 4　基于无监督学习的方法

Tab. 4　Methods based on unsupervised learning

分类	相关方法	特点
基于自监督学习	Doc2vec[62]、Sent2vec[63]、Skip-Thought[64]、Quick-Thought[65]、SDAE[66]、FastSent[66]	该类方法利用自监督学习设计相关任务, 通过数据本身携带的信息训练模型, 然后通过训练好的模型得到句子的向量表示, 在此基础上计算语义文本相似度
基于数学分析	WMD[67]、SIF[68]、P-means[69]	该类方法无须训练, 直接通过PCA降维、线性规划等数学工具, 将词向量加权求和得到句向量表示后, 计算向量间距离以表征语义文本相似度

Le 等[62] 在 word2vec[55] 的启发下提出了 doc2vec. 该模型在训练词向量的同时, 加入表征段落的向量和词向量共同训练. doc2vec 与 word2vec 相同, 有两种训练方式, 一种是通过段落向量和上下文单词向量预测中心词; 另一种则是通过段落向量预测文本中包含的单词. 这样就可以通过计算训练好的段落向量之间的余弦值表征其语义文本相似度.

Pagliardini 等[63] 提出的 sent2vec 模型是 word2vec 模型中 CBOW 的扩展, 其不仅仅使用窗口中的词来预测目标词, 而且使用窗口中所有的 n-grams 表示来预测目标词. 为了得到句子向量, 将句子看成一个完整的窗口, 模型的输入为句子中的 n-grams 表示, 目标是预测句子中的 missing word(目标词), 而句子向量是所有 n-grams 向量表示的平均.

Kiros 等[64] 提出的 Skip-Thought 模型同样是从 word2vec 方法中获得灵感, 其基本思想是将 word2vec[55] 中的 skip-gram 模型通过 encoder-decoder 架构拓展到句子级别, 即用中心句预测其上下句, 更具体地说, 首先通过 encoder 对中心句进行编码, 然后通过 decoder 预测中心句的上下两个句子, 而模型的目标是, 预测的上下句与其对应的真实句越接近越好. Logeswaran 等[65] 在 Skip-Thought[64] 的基础上, 提出了一种更加高效的方法 Quick-Thought, 它主要是用分类任务代替了之前的预测任务, 对模型任务进行了简化, 实验结果证明, QT 在训练时间较少的情况下, 依然能够达到非常不错的效果.

Hill 等[66] 提出的模型称为序列去噪自编码器 (Sequential Denoising AutoEncoder, SDAE). 其包括基于 LSTM 的编码器和解码器两部分, 输入信息通过编码器产生编码信息, 再通过解码器得到输出信

息, 并且在原始输入句子上通过将词序互换、删除某些单词等方法加入噪音, 模型的目标是使输出信息和原始信息越接近越好. 该方法相较于 Skip-Thought[64] 方法的优势在于只需要输入单个句子, 而 Skip-Thought 需要输入 3 个有序的句子. Hill 等[66] 同时还提出了 FastSent 模型, 该模型和 Skip-Thought[64] 一样, 都是基于 word2vec 中的 skip-gram, 但是该方法忽略了词序信息, 只需要预测周围句子中包含的单词, 而无须按照单词在句子中的顺序进行逐一预测, 该方法计算速度快, 在无监督任务上效果也优于 Skip-Thought[64].

以上方法都是利用自监督的思想, 在数据集中通过数据本身携带的信息进行模型训练, 以下方法则无须进行训练, 直接通过线性规划、PCA 降维等在词向量的基础上得到句向量.

Kusner 等[67] 将句子相似度问题转换为运输问题, 引入词移距离, 将文本间距离转化为约束条件下的最优化问题, 巧妙地将运输问题中的 EMD 算法应用到相似度计算当中, 得到了 WMD 算法.

Arora 等[68] 提出的 SIF 算法则是将主成分分析 (PCA) 用于句向量的生成, 首先用通过改进的 TFIDF 方法对词向量进行加权平均得到句向量, 然后减去所有句子向量组成的矩阵的第一个主成分上的投影, 得到最终的句子嵌入. 实验表明该方法具有不错的竞争力, 在大部分数据集上都比平均词向量或者使用 TFIDF 加权平均的效果要好.

Ruckle 等[69] 是对词向量求平均得到句向量的思想进行了改进, 通过引入超参数 p 将“词向量求平均得到句向量”的操作泛化为 p-means 的一类操作, 取不同的 p 值产生不同的特征, 当 p=1 时, 该方法等同于求平均的操作. 同时本文使用了多种词向量 (如 word2vec 和 Glove 等), 实验结果表明该方法在无监督任务上具有很好的性能.

无监督的方法不需要带有标签的训练集对模型进行训练, 减少了人工打标签的成本, 并且更加通用, 在一些资源稀少的特定领域应用广泛, 但同时由于无法将带有标签的信息和一些先验知识融入其中, 相对于监督方法而言, 其计算准确率较低.

4.3 基于监督学习方法的语义文本相似度计算

监督学习方法是一类需要带有标签的训练集对模型进行训练后才能进行语义文本相似度计算的方法, 这类方法由于有标签对模型进行指导, 在多数有训练集的任务上, 其性能要优于无监督学习方法. 监督学习方法从模型架构上可以分为两种, 一种是孪生网络 (Siamese Network) 架构; 另一种是交互 (interaction-based) 模型架构, 两种模型的典型架构如图 2 所示.

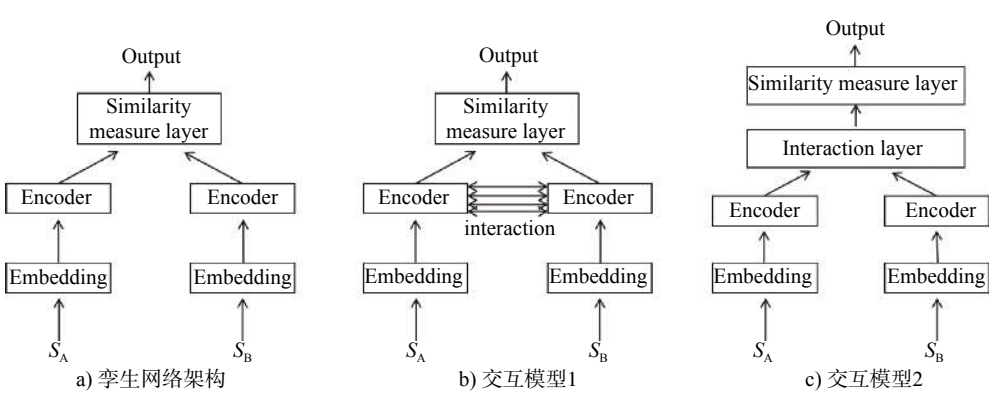

图 2 监督学习方法模型架构

Fig. 2 Model architecture for methods based on supervised learning

孪生网络架构一般分为 3 层, 分别为输入层、编码层和相似度测量层. 如图 2a) 所示, 输入层主要是句子分词后, 将单词映射为词向量后输入编码层; 编码层对句中的词向量进行编码得到句向量; 相

似度测量层则是对两个句向量进行相似度计算, 可以是简单地利用向量距离如欧式距离、余弦相似度等直接表征两个句子的语义相似性, 也可以是将两个句向量拼接后传入多层感知机或其他分类器得到最终的语义文本相似度度量. “孪生”主要体现在句子对中 S_A 和 S_B 同时输入左右两个网络当中, 这两个网络的输入层和编码层模型架构完全相同并共享权重.

孪生网络在语义文本相似度计算中效果很好, 但缺少了两个句子在编码过程中的交互, 两个句子编码时相互独立, 这样不利于后面的相似度度量, 因此交互模型应运而生.

交互模型如图 2b) 和图 2c) 所示, 整体与孪生网络类似, 但在编码层利用 Attention 机制或其他技术增加两个网络间的交互, 然后将交互结果传入相似度度量层. 表 5 是按照模型层次结构中所用的技术进行的简要总结, 接下来将分小节详细介绍基于孪生网络的方法和基于交互模型的方法.

表 5　监督方法模型简要总结

Tab. 5　Summary of models based on supervised learning

Type	Models	Sentence encoder layer	Interaction layer	Similarity measure layer
Siamese Network	DSSM[70]	MLP	-	Cosine Similarity
	CNN-DSSM[71]	CNN+MLP	-	Cosine Similarity
	LSTM-DSSM[72]	LSTM	-	Cosine Similarity
	Siamese-LSTM[73]	LSTM	-	Manhattan distance+ SVM
	Lin等[74]	BiLSTM+ self-attention	-	dot product+ MLP
	Pontes等[75]	CNN+LSTM	-	Manhattan distance
	InferSent[76]	GRU、LSTM、BiGRU、BiLSTM等	-	Maxpooling+ MLP
	Yang等[77]	DAN、Transformer	-	MLP
	USE[78]	Transformer	-	MLP
Interaction-based	ABCNN[79]	CNN	Attention	Pooling+ MLP
	PWIM[80]	BiLSTM	cosine similarity + Euclidean distance +dot product	CNN+ MLP
	BiMPM[81]	BiLSTM	matching	MLP
	DIIN[82]	self-attention	dot product	DenseNet
	DRCN[83]	BiLSTM+DenseNet[84]	vector addition+ vector subtraction+ vector modulo	MLP

4.3.1　基于孪生网络的方法

Huang 等[70] 提出的 DSSM 算法是最先将孪生网络架构用于语义文本相似度计算的算法之一, DSSM 架构主要分为输入层、表示层、匹配层, 这种三层架构也是后来基于孪生网络的算法最常用的架构. 输入层主要将原始文本映射为向量, 由于本文在 2013 年被提出, 当时的分布式词向量刚刚问世, 因此本文并没有使用词向量, 而是使用字符级的 trigram 方法将文本映射为高维向量, 更具体地说, trigram 是将 3 个字符为一组映射为 one-hot 向量, 最后将句子中的 trigram 向量相加得到高维句子向量表示. 表示层是将高维的句子向量映射为低维向量, DSSM 表示层就是简单的几个全连接层, 最终将句子映射为 128 维的低维向量. 匹配层是将两个低维句子向量表示之间求余弦相似度来表征两个句子的语义相似度. DSSM 模型在文本匹配任务上取得了突出的成绩, 但忽略了语序信息和上下文信息.

Shen 等[71] 将 CNN[72] 网络引入 DSSM 模型以保留更多的上下文信息. 该方法对 DSSM 的改进主

要发生在表示层, 该模型表示层中添加了卷积层和池化层, 使得上下文信息得到了有效保留, 但是由于卷积核的限制, 距离较远的上下文信息仍会丢失.

Palangi 等[73] 为了保留更多的上下文信息, 将 LSTM[74] 网络引入其中, 该网络考虑到了距离更远的上下文信息和一些语序信息, 使得算法效果有所提升. Mueller 等[76] 在预训练的词向量的基础上同样使用 Siamese-LSTM 对句子进行编码. 然后, 使用编码得到的句向量之间的曼哈顿距离来预测句子对的语义相似度. 实验证明将该方法和 SVM 结合用于情感分类 (Entailment Classification), 效果提升明显.

Lin 等[77] 将双向 LSTM(BiLSTM) 和 Self-Attention 技术相结合得到句子向量表示, 具体来说就是, 首先将句子通过 BiLSTM 模型, 将得到的每一时刻的两个方向的向量拼接成一个二维矩阵, 然后通过自注意力机制 (Self-Attention)[60] 得到句中每个词向量对应的权重, 最终通过词向量的加权求和得到句向量. 在训练网络时同样是使用 Siamese 架构, 在得到句向量后进行简单的特征提取, 如拼接、点积、对位相减等, 然后输入一个多层感知机, 得到最终的语义文本相似度.

Pontes 等[75] 将 CNN 模型和 LSTM 模型同时用于 Siamese 架构来计算语义文本相似度, 首先将句子分成局部片段, 然后将每个片段分别经过 CNN 网络得到片段向量, 再将原句中的词向量和它所对应的上下文向量拼接起来传入 LSTM 网络, 最终得到句向量后, 计算句向量间的曼哈顿距离以表征语义文本相似度.

下面 3 种方法基于迁移学习进行语义文本相似度计算, 基本思想是在其他 NLP 任务中进行训练然后将生成的句向量用于计算语义文本相似度当中.

Conneau 等[78] 提出 InferSent 模型, 在自然语言推理 (NLI) 任务上, 通过孪生网络架构训练通用句向量, 文中使用了 7 种不同的 encoder, 其中包括 GRU、LSTM、双向 GRU、双向 LSTM 等, 并通过最终实验证明, 在 encoder 选取双向 LSTM 并使用 max-pooling 技术的情况下, 生成的句向量比 Skip-Thought[64] 和 FastSent[66] 等方法要更好.

Yang 等[85] 提出了“如果句子的回复具有相似的分布, 那么它们在语义上也是相似的.”的想法, 从这个想法出发, 通过使用对话数据来学习句子级向量表示, 并通过 SNLI 数据集对模型进行加强. 该模型基于孪生网络架构, 一共用了两种编码器, 一种为 DAN, DAN 是指将句子中的词向量进行加权平均之后送入前馈神经网络中得到句子隐层表示. 另一种为 Transformer[60] 架构的编码部分, 主要由多头注意力 (Multi-head Attention) 机制和前馈神经网络组成. 通过上述两种编码器对句子进行编码得到句向量, 再经过简单的特征提取后送入多层感知机模型得到最终的相似度得分. 实验表明, 该模型在语义文本相似度任务上表现突出.

Cer 等[86] 提出了 USE 模型, 在 Yang 等[85] 的模型基础上结合 Skip-Thought, 将模型拓展到了更多的任务上, 通过多任务学习得到通用句子表示.

纵览上述方法, 可以清晰地看到基于孪生网络的计算方法的发展进程, 对于模型的改进主要集中在使用不同的编码器, 如从最开始的多层感知机发展为 CNN、LSTM, 再到 Transformer 等. 从训练数据角度上说, 这类方法从单一训练数据集逐步发展到迁移学习, 再发展到多任务学习. 目前基于多任务学习计算语义文本相似度的探索才刚刚开始, 针对不同任务的训练数据具有不同的标签, 包含着句子不同角度的信息, 那么建立一个通用的模型架构, 通过在不同的任务数据集上进行训练, 从而提升模型性能, 是未来重要的研究方向之一.

4.3.2 基于交互模型的方法

基于孪生网络的方法在编码层对句子编码时是相互独立的, 句子对之间没有交互, 这样对于计算句子对的语义相似度会造成一定影响, 基于交互模型的方法就是为了解决这个问题而产生的, 它是在孪生网络的基础上增加两个平行网络之间的交互作用, 从而提取到句子对之间更加丰富的交互信息.

Yin 等[79] 提出了 ABCNN 模型, 是在词向量基础上通过 CNN 网络对句子进行处理, 在分别对句子对中的句子进行卷积和池化操作的同时, 使用 Attention 机制对两个中间步骤进行交互, 文中一共尝试了 3 种添加 Attention 的策略, 主要区别是作用于模型的不同阶段, 如第一种 Attention 是直接作用在词向量组成的矩阵上, 而第二种 Attention 是作用在经过卷积和池化操作后产生的输出矩阵上, 第三种则是将前两种方法相结合. 可以将经过卷积和池化操作后得到的结果看作短语向量, 而该短语向量的长度取决于卷积核的大小, 从这个角度理解, 第一种和第二种 Attention 方法的区别实质上是在不同粒度上对模型进行了处理.

He 等[80] 通过 BiLSTM 对句子进行建模提出 PWIM 算法, 该方法共分为 4 个部分. 第一部分将 BiLSTM 网络每个时刻的输出, 即该时刻的双向 hidden state 做拼接获得的结果, 作为对应时刻 word 的 representation. 第二部分通过计算两个句子的每个时刻的向量表示进行余弦相似度、欧式距离和点积计算, 然后根据计算结果来确定不同向量对的权重, 该方法认为句子内部不同的词的重要性是不一样的, 两个句子间重要的单词对, 对于句子相似度的计算贡献更大, 这些单词对应该得到更多的重视, 最后通过多层 CNN 网络得到最终结果.

Wang 等[81] 同样基于 BiLSTM 网络提出 BiMPM 模型, 其不同之处在于输入层使用词向量与字符向量的拼接, 值得注意的是, 该方法中设计了 4 种不同的匹配方法, 并将这些方法用于对两个句子在 LSTM 网络中产生的中间向量进行匹配, 再用匹配的结果进行拼接得到最终的句向量, 这也是该网络的交互思想的体现.

Gong 等[82] 提出 DIIN 模型, 输入层使用单词嵌入、字符特征和句法特征的串联, 编码层使用自注意力[60](self-attention) 机制, 交互层采用点积操作得到交互矩阵, 然后使用 DenseNet[84] 进行特征抽取, 之后将抽取的特征传入多层感知机模型得到最终的结果, 本方法简单有效, 在 NLI 任务上表现出很好的性能. Kim 等[83] 提出的 DRCN 模型和 DIIN 的结构十分相似, 不同之处在于 DRCN 在特征提取阶段结合了 DenseNet[84] 的连接策略与 Attention 机制, 在交互阶段也采取了更加多样化的交互策略, 提高了算法性能.

基于交互模型的方法实质就是在孪生网络架构的基础上, 通过某种策略对两个孪生网络的中间环节进行交互. 目前最普遍的策略是各种不同的 Attention 机制. 具有交互能力的模型结构普遍更为复杂, 包含更多的模型参数, 这就导致了这类模型的计算成本较高.

5　数据集及评测指标

高质量的公开数据集和统一的评测标准对一个领域的研究活动至关重要, 可以让完全不同的两种模型进行性能上的对比实验, 为进一步研究做铺垫. 语义文本相似度 (STS) 任务的相关研究在近几年急剧增加, 这主要是由国际语义评估研讨会 (SemEval) 推动的. SemEval 的赛事组织者中有许多非常有影响力的学者, 过去几年里, 该赛事为语义文本相似度任务提供了很多高质量的公开数据集, 并规范了统一的评测标准.

表 6 展示了目前语义文本相似度任务的常见公开数据集及其数据来源.

●**STS2012-2017:** 这一系列的公开数据集是 SemEval 从 2012 年至 2017 年间陆续在研讨会中提出的, 这个系列的公开数据集是目前语义文本相似度领域最为常用的公开数据集, 几乎所有新的语义文本相似度计算模型都会使用这些数据集对模型进行评估. 这些数据集由句子对组成, 每个句子对都由人工标注了两个句子之间的相似度得分, 相似度得分为 0—5, 分数越高则说明两个句子越相似. 其中, STS2017 提供了文本相似度的跨语言测试集, 其中包含了 7 种语言, 如英语、土耳其语、阿拉伯语和西班牙语等.

表 6 语义文本相似度任务常用公开数据集

Tab. 6 Common datasets for semantic textual similarity

数据集名称	句子对数量(训练、测试)	数据来源
STS2012	5150(3150, 2000)	现存释义集、新闻、视频描述、机器翻译评估集
STS2013	3750(2000, 1750)	新闻、机器翻译评估集
STS2014	14974(7592, 7382)	新闻、论坛、Twitter
STS2015	14342(11342, 3000)	新闻、论坛、图像描述
STS2016	1884(1000, 884)	新闻、机器翻译评估集、论坛
STS2017	3210(2210, 1000)	新闻、机器翻译评估集、论坛
TwitterPPDB	51524(42200, 9324)	Twitter
MSRVID	1500(750, 750)	视频描述
SICK	9927(5000, 4927)	视频描述、图像描述

●**TwitterPPDB:** 该数据集主要是在 Twitter 上收集的数据, 同样是以句子对的形式构建, 并且人工标注句子对的相似度得分, 得分在 1—5 之间, 得分越高则句子对越相似.

●**MSRVID:** 该数据集全称是微软视频释义语料库 (Microsoft Video Paraphrase Corpus). 它是在 2012 年 SemEval 竞赛中收集的, 包含 1500 对简短的视频描述, 然后进行了人工标注. 一半用于训练; 另一半用于测试. 每个句子对都有一个相关性分数$\in[0, 5]$, 分数越高, 表明两个句子之间的相似性越高.

●**SICK:** 该数据集是针对 2014 年 SemEval 竞赛收集的, 由 9 927 个句子对组成, 其中 4 500 个用于训练, 500 个作为验证集, 其余 4 927 个在测试集中. 这些句子来自图像和视频描述. 每个句子对都用一个相关性分数$\in[1, 5]$ 进行注释, 分数越高, 表明两个句子之间的关系越紧密.

以上是 STS 任务中最常用的公开数据集, 在语义文本相似度任务中, 由于预测值和标签值多为 0 到 5 之间的连续值, 因此准确率等指标无法直接使用. 该任务常用的评估指标包括皮尔逊相关系数、斯皮尔曼相关系数和均方误差.

●**皮尔逊相关系数 (Pearson correlation coefficient)**: Pearson 系数是用于度量两个变量 X 和 Y 之间的相关性, 将模型在测试集上预测的所有值看成一个多维变量 X, 将测试集对应的标签值看成多维变量 Y, 之后可用下列公式计算 X 和 Y 的 Pearson 系数 r:

$$r=\frac{\sum\limits_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum\limits_{i=1}^{n}(X_i-\bar{X})^2}\sqrt{\sum\limits_{i=1}^{n}(Y_i-\bar{Y})^2}},$$

r 值就是皮尔逊相关系数的大小, 代表了两个变量的相关程度, 取值范围为 (–1, 1), 该数值越接近于 1, 则预测值与真实值越相关, 模型的效果也就越好.

●**斯皮尔曼相关系数 (Spearman's rank correlation coefficient)**: 通常也叫斯皮尔曼秩相关系数. “秩”, 可以理解成就是一种顺序或者排序, 该系数就是根据原始数据的排序位置进行求解. Spearman 系数同时也叫作等级之间的皮尔逊相关系数, 等级是指某数值在其所在的列中从小到大排序后所在的位置. 其计算公式如下:

$$\rho=1-\frac{6\sum d_i^2}{n(n^2-1)},$$

n为测试集中句子对的个数, d_i指的是X_i和Y_i之间的等级差. 取值范围同样在 (–1, 1) 之间, 在评估模型时, ρ越接近于 1, 则模型的性能越好.

●**均方误差 (mean-square error, MSE)**: 均方误差是线性回归任务常用的评估指标, 是预测值与真实值之差的平方和的平均值, 同样适用于语义文本相似度计算任务, 计算公式如下:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\widehat{y}_i - y_i)^2,$$

其中 n 为样本个数, $\widehat{y}_i$表示预测值, y_i表示真实值, 均方误差越小, 则说明预测值与真实值越接近, 则模型预测结果越好.

6　总结与展望

本文首先对语义文本相似度计算方法进行了系统的分类, 并对每一类中的典型方法进行了详细的介绍. 本文在详细介绍了基于传统的语义文本相似度方法的同时, 对该领域近几年出现的基于深度学习的语义文本相似度计算方法进行了全面的调研与总结. 除了对各类方法的介绍外, 本文针对每一类方法存在的问题都进行了分析与评述. 纵览语义文本相似度计算方法的发展历程, 本文总结了一些未来可能的研究方向, 具体如下:

●近几年, 基于深度学习的方法发展迅速, 很多学者忽略了一些传统方法, 导致深度学习方法和传统方法的割裂, 特别是基于知识库的方法, 我们将结构化语义词典中包含的信息作为先验知识, 可以更好地用于深度学习模型的训练. 换句话说, 将传统方法与深度学习方法进行深度融合是未来可能的一个研究方向.

●基于深度学习的方法发展至今, 逐步从单一任务模型发展到了多任务模型, 目前对于利用多任务学习生成通用句向量进行语义文本相似度计算的研究才刚刚起步. 如何建立通用神经网络架构在多任务中进行训练, 同时提取不同任务中包含的文本信息是一个非常有价值的研究方向.

●目前, 关于深度强化学习的研究在多个领域都取得了突出的成果. Chen 等[87] 首次将深度强化学习方法与孪生网络架构相结合, 应用于语义文本相似度计算, 这是目前为数不多的将深度强化学习应用于语义文本相似度计算的文章, 但是文中仅仅使用深度强化学习作为一种类似特征抽取器的功能, 并没有发挥出深度强化学习的优势, 如何更好地将深度强化学习方法应用于语义文本相似度计算是值得研究的方向之一.

[参　考　文　献]

[1] BLOEHDORN S, BASILI R, CAMMISA M, et al. Semantic kernels for text classification based on topological measures of feature similarity [C]//Proceeding of the Sixth International Conference on Data Mining (ICDM'06). 2006: 808-812.

[2] TONG Y, GU L. A news text clustering method based on similarity of text labels [C]//International Conference on Advanced Hybrid Information Processing. 2018: 496-503.

[3] ATTARDI G, SIMI M, DEI R S. TANL-1: Coreference resolution by parse analysis and similarity clustering [C]//Proceedings of the 5th International Workshop on Semantic Evaluation. 2010: 108-111.

[4] DAS A, MANDAL J, DANIAL Z, et al. A novel approach for automatic bengali question answering system using semantic similarity analysis[EB/OL]. (2019-10-23)[2020-07-01]. https://arxiv.org/ftp/arxiv/papers/1910/1910.10758.pdf.

[5] AMIR S, TANASESCU A, ZIGHED D A. Sentence similarity based on semantic kernels for intelligent text retrieval [J]. Journal of Intelligent Information Systems, 2017, 48(3): 675-689.

[6] SOORI H, PRILEPOK M, PLATOS J, et al. Semantic and similarity measure methods for plagiarism detection of students' assignments [C]//Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015. 2016: 117-125.

[7] VADAPALLI R, KURISINKEL L J, GUPTA M, et al. SSAS: Semantic similarity for abstractive summarization [C]//Proceedings of

the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers). 2017: 198-203.
[8] QIAN M, LIU J, LI C, et al. A comparative study of English-Chinese translations of court texts by machine and human translators and the Word2Vec based similarity measure's ability to gauge human evaluation biases [C]//Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks. 2019: 95-100.
[9] MAJUMDER G, PAKRAY P, GELBUKH A, et al. Semantic textual similarity methods, tools, and applications: A survey [J]. Computación y Sistemas, 2016, 20(4): 647-665.
[10] 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.
[11] RISTAD, ERIC S, YIANILOS, et al. Learning string-edit distance[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(5): 522-532.
[12] XU X, CHEN L, HE P. Fast sequence similarity computing with LCS on LARPBS [C]//International Symposium on Parallel and Distributed Processing and Applications. 2005: 168-175.
[13] KONDRAK G. *N*-gram similarity and distance [C]// String Processing and Information Retrieval. 2005: 115-126.
[14] NIWATTANAKUL S, SINGTHONGCHAI J, NAENUDORN E, et al. Using of Jaccard Coefficient for Keywords Similarity [J]. Lecture Notes in Engineering and Computer Science, 2013, 1(3): 13-15.
[15] 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.
[16] SLANEY M, CASEY M. Locality-sensitive hashing for finding nearest neighbors [J]. IEEE Signal processing magazine, 2008, 25(2): 128-131.
[17] SALTON G, WONG A, YANG C S, et al. A vector space model for automatic indexing [J]. Communications of The ACM, 1975, 18(11): 613-620.
[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 [J]. Uncertainty in Artificial Intelligence, 1999, 15(6): 289-296.
[20] BLEI D M, NG A Y, JORDAN M I, et al. Latent dirichlet allocation [J]. Journal of Machine Learning Research, 2012(3): 993-1022.
[21] GUO Q L, LI Y M, TANG Q. Similarity computing of documents based on VSM [J]. Application Research of Computers, 2008, 25(11): 3256-3258.
[22] LI L. Research and implementation of an improved VSM-based text similarity algorithm [J]. Computer Applications and Software, 2012, 29(2): 282-284.
[23] TASI C, HUANG Y, LIU C, et al. Applying VSM and LCS to develop an integrated text retrieval mechanism [J]. Expert Systems With Applications, 2012, 39(4): 3974-3982.
[24] 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.
[25] XIONG D P, WANG J, LIN H F. An LDA-based approach to finding similar questions for community question answer [J]. Journal of Chinese Information Processing, 2012, 26(5): 40-45.
[26] ZHANG C, CHEN L, LI X, et al. Chinese text similarity algorithm based on PST_LDA [J]. Application Research of Computers, 2016, 33(2): 375-377.
[27] MIAO Y, YU L, BLUNSOM P, et al. Neural variational inference for text processing [EB/OL]. (2016-01-04)[2020-07-01]. https://arxiv.org/pdf/1511.06038.pdf.
[28] LAU J H, BALDWIN T, COHN T, et al. Topically Driven Neural Language Model [C]// Meeting of the Association for Computational Linguistics. 2017: 355-365.
[29] MILLER, GEORGE A. WordNet: A lexical database for English [J]. Communications of the Acm, 1995, 38(11): 39-41.
[30] 梅家驹, 竺一鸣, 高蕴琦, 等. 同义词词林 [M]. 上海: 上海辞书出版社, 1983.
[31] 董振东. 语义关系的表达和知识系统的建造 [J]. 语言文字应用, 1998(3): 76-82.
[32] RADA R, MILI H, BICKNELL E J, et al. Development and application of a metric on semantic nets [J]. IEEE Transaction on System Man &Cybernetics, 1989, 19(1):17-30.
[33] RICHARDSON R, SMEATON A F. Using WordNet in a knowledge-based approach to information retrieval [EB/OL]. (1995-02-01)[2020-07-01]. http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=0DDA60E11D37A7DA2777BF162C86760F?doi=10.1.1.48.9324&rep=rep1&type=pdf.
[34] LEACOCK C, CHODOROW M. Combining local context and WordNet similarity for word sense identification [M]// FELLBAUM C. WordNet: An Electronic Lexical Database. Massachusetts: MIT Press, 1998.
[35] WU Z B. Verb semantics and lexical selection[C]// Acl Proceedings of Annual Meeting on Association for Computational Linguistics. 1994: 133-138.
[36] HIRST G, STONGE D. Lexical chains as representations of context for the detection and correction of malapropisms[M]// FELLBAUM C. WordNet: An Electronic Lexical Database. Massachusetts: MIT Press, 1998, 305: 305-332.
[37] YANG D, POWERS D M W. Measuring semantic similarity in the taxonomy of WordNet [C]// ACSC'05: Proceedings of the Twenty-eighth Australasian conference on Computer Science. 2005, 38: 315-322.
[38] RESNIK P. Using information content to evaluate semantic similarity in a taxonomy [C]// IJCAI'95: Proceedings of the 14th International Joint Conference on Artificial Intelligence. 1995(1): 448-453.
[39] JIANG J J, CONRATH D W. Semantic similarity based on corpus statistics and lexical taxonomy [EB/OL]. (1997-10-01)[2020-07-01]. https://arxiv.org/pdf/cmp-lg/9709008.pdf.
[40] LIN D. An information-theoretic definition of similarity [C]//ICML'98: Proceedings of the Fifteenth International Conference on

Machine Learning. 1998(7): 296-304.
[41] 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.
[42] BANERJEE S, PEDERSEN T. An adapted lesk algorithm for word sense disambiguation using WordNet [C]//International Conference on Intelligent Text Processing and Computational Linguistics. 2002: 136-145.
[43] PEDERSEN T, PATWARDHAN S, MICHELIZZI J. WordNet : Similarity-Measuring the relatedness of concepts [C]//Demonstrations'04: Demonstration Papers at HLT-NAACL 2004. 2004(5): 38-41.
[44] 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.
[45] SHI B, FANG L Y, YAN J Z, et al. Ontology-based measure of semantic similarity between concepts [C]//WCSE '09: Proceedings of the 2009 WRI World Congress on Software Engineering. 2009(2): 109-112.
[46] 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: 242-247.
[47] 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.
[48] 李峰, 李芳. 中文词语语义相似度计算——基于《知网》2000 [J]. 中文信息学报, 2007, 21(3): 99-105.
[49] 江敏, 肖诗斌, 王弘蔚, 等. 一种改进的基于《知网》的词语语义相似度计算 [J]. 中文信息学报, 2008, 22(5): 84-89.
[50] STRUBE M, PONZETTO S P. WikiRelate! Computing semantic relatedness using Wikipedia [C]//AAAI'06: Proceedings of the 21st National Conference on Artificial Intelligence. 2006(2): 1419-1424.
[51] GABRILOVICH E, MARKOVITCH S. Computing semantic relatedness using wikipedia-based explicit semantic analysis [C]//IJCAI' 07: Proceedings of the 20th International Joint Conference on Artifical Intelligence. 2007(1): 1606-1611.
[52] WITTEN I, MILNE D N. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links [C]//Proceedings of AAAI'2008. 2008: 25-30.
[53] YEH E, RAMAGE D, MANNING C D, et al. WikiWalk: Random walks on Wikipedia for semantic relatedness [C]//Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing. 2009: 41-49.
[54] CAMACHO-COLLADOS J, PILEHVAR M T, NAVIGLI R. Nasari: A novel approach to a semantically-aware representation of items [C]//Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2015: 567-577.
[55] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space [EB/OL]. (2013-09-07)[2020-07-01]. https://arxiv.org/pdf/1301.3781.pdf.
[56] PENNINGTON J, SOCHER R, MANNING C D. Glove: Global vectors for word representation [C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014: 1532-1543.
[57] JOULIN A, GRAVE E, BOJANOWSKI P, et al. Bag of tricks for efficient text classification [EB/OL]. (2016-08-09)[2020-07-01]. https://arxiv.org/pdf/1607.01759.pdf.
[58] PETERS M E, NEUMANN M, IYYER M, et al. Deep contextualized word representations [EB/OL]. (2018-03-22)[2020-07-01]. https://arxiv.org/pdf/1802.05365.pdf.
[59] RADFORD A, NARASIMHAN K, SALIMANS T, et al. Improving language understanding by generative pre-training [EB/OL]. (2018-11-05)[2020-07-01]. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/languageunsupervised/language understanding paper.pdf.
[60] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C]//NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 5998-6008.
[61] DEVLIN J, CHANG M W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding [EB/OL]. (2019-05-24)[2020-07-01]. https://arxiv.org/pdf/1810.04805.pdf.
[62] LE Q, MIKOLOV T. Distributed representations of sentences and documents [C]//ICML'14: Proceedings of the 31st International Conference on International Conference on Machine Learning. 2014, 32: 1188-1196.
[63] PAGLIARDINI M, GUPTA P, JAGGI M. Unsupervised learning of sentence embeddings using compositional n-gram features [EB/OL]. (2018-12-28)[2020-07-01]. https://arxiv.org/pdf/1703.02507.pdf.
[64] KIROS R, ZHU Y, SALAKHUTDINOV R R, et al. Skip-thought vectors [C]//Advances in neural information processing systems. 2015: 3294-3302.
[65] LOGESWARAN L, LEE H. An efficient framework for learning sentence representations [EB/OL]. (2018-03-07)[2020-07-01]. https://arxiv.org/pdf/1803.02893.pdf.
[66] HILL F, CHO K, KORHONEN A. Learning distributed representations of sentences from unlabelled data [EB/OL]. (2016-02-10)[2020-07-01]. https://arxiv.org/pdf/1602.03483.pdf.
[67] KUSNER M, SUN Y, KOLKIN N, et al. From word embeddings to document distances [C]//International Conference on Machine Learning. 2015: 957-966.
[68] ARORA S, LIANG Y, MA T. A simple but tough-to-beat baseline for sentence embeddings [EB/OL]. (2017-02-04)[2020-07-01]. https://openreview.net/pdf?id=SyK00v5xx.
[69] RÜCKLÉ A, EGER S, PEYRARD M, et al. Concatenated power mean word embeddings as universal cross-lingual sentence representations [EB/OL]. (2018-09-12)[2020-07-01]. https://arxiv.org/pdf/1803.01400.pdf.
[70] 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.

[71] 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 conference on information and knowledge management. 2014: 101-110.
[72] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks [C]//Advances in nNeural Information Processing Systems. 2012: 1097-1105.
[73] PALANGI H, DENG L, SHEN Y, et al. Semantic modelling with long-short-term memory for information retrieval [EB/OL]. (2015-02-27)[2020-07-01]. https://arxiv.org/pdf/1412.6629.pdf.
[74] GERS F. Long short-term memory in recurrent neural networks [D]. Lausanne: EPFL, 2001.
[75] PONTES E L, HUET S, LINHARES A C, et al. Predicting the semantic textual similarity with siamese CNN and LSTM [EB/OL]. (2018-10-24)[2020-07-01]. https://arxiv.org/pdf/1810.10641.pdf.
[76] MUELLER J, THYAGARAJAN A. Siamese recurrent architectures for learning sentence similarity [C]//AAAI'16: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. 2016(2): 2786-2792 .
[77] LIN Z, FENG M, SANTOS C N, et al. A structured self-attentive sentence embedding [EB/OL]. (2017-03-09)[2020-07-01]. https://arxiv.org/pdf/1703.03130.pdf.
[78] CONNEAU A, KIELA D, SCHWENK H, et al. Supervised learning of universal sentence representations from natural language inference data [EB/OL]. (2017-07-21)[2020-07-01]. https://arxiv.org/pdf/1705.02364v4.pdf.
[79] YIN W, SCHÜTZE H, XIANG B, et al. Abcnn: Attention-based convolutional neural network for modeling sentence pairs [J]. Transactions of the Association for Computational Linguistics, 2016(4): 259-272.
[80] HE H, LIN J. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement [C]//Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016: 937-948.
[81] WANG Z, HAMZA W, FLORIAN R. Bilateral multi-perspective matching for natural language sentences [C]//Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence Main track. 2017: 4144-4150.
[82] GONG Y, LUO H, ZHANG J. Natural language inference over interaction space [EB/OL]. (2018-05-26)[2020-07-01]. https://arxiv.org/pdf/1709.04348.pdf.
[83] KIM S, KANG I, KWAK N. Semantic sentence matching with densely-connected recurrent and co-attentive information [C]//Proceedings of the AAAI conference on artificial intelligence. 2019, 33: 6586-6593.
[84] HUANG G, LIU Z, VAN DER MAATEN L, et al. Densely connected convolutional networks [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.
[85] YANG Y, YUAN S, CER D, et al. Learning semantic textual similarity from conversations [EB/OL]. (2018-04-20)[2020-07-01]. https://arxiv.org/pdf/1804.07754.pdf.
[86] CER D, YANG Y, KONG S, et al. Universal sentence encoder [EB/OL]. (2018-04-12)[2020-07-01]. https://arxiv.org/pdf/1803.11175.pdf.
[87] CHEN G, SHI X, CHEN M, et al. Text similarity semantic calculation based on deep reinforcement learning [J]. International Journal of Security and Networks, 2020, 15(1): 59-66.

(责任编辑: 张 晶)