·人工智能及识别技术· 文章编号：1000-3428(2017)05-0143-06 文献标志码：A 中图分类号：TP391

微博文本的句向量表示及相似度计算方法研究

段旭磊，张仰森，孙祎卓

（北京信息科技大学 智能信息处理研究所，北京 100192）

摘 要： 在 Word2vec 框架内，针对微博文本的特点，提出采用词向量或高维词库映射计算句向量的方法。以 3 种算法构造句向量，即采用 Word2vec 对微博文本进行扩展后以 TF-IDF 方法表示句向量；将句子中每个词的词向量相加形成句向量；构建高维词库，将句子中的每个词映射到高维词库形成句向量。对比 3 种训练句向量的方法，选出最适合微博领域的模型。实验结果表明，采用高维词库映射的方法对微博的句向量计算的效果最佳。

关键词： 微博文本；相似度计算；词向量；高维词库；句向量

中文引用格式： 段旭磊，张仰森，孙祎卓. 微博文本的句向量表示及相似度计算方法研究[J]. 计算机工程，2017，43(5)：143-148.

英文引用格式： Duan Xulei，Zhang Yangsen，Sun Yizhuo. Research on Sentence Vector Representation and Similarity Calculation Method About Microblog Texts[J]. Computer Engineering，2017，43(5)：143-148.

Research on Sentence Vector Representation and Similarity Calculation Method About Microblog Texts

DUAN Xulei，ZHANG Yangsen，SUN Yizhuo

（Institute of Intelligence Information Processing，Beijing Information Science and Technology University，Beijing 100192，China）

【Abstract】 In Word2vec framework，aiming at the feature of Microblog text，this paper proposes the method using word vector or sentence vector of high dimension word database mapping calculation，and constructs sentence vector using three algorithms：Using Word2vec to expand text，and TF-IDF to obtain sentence vector. Adding the word vectors into sentence vector；Building a word bank to obtain high dimension vector space for the sentence. Through comparing the three methods，it selects the model fitting Microblog field. Experimental results show that the sentence vector calculation method using high dimension word database mapping is the best.

【Key words】 Microblog texts；similarity calculation；word vector；high dimension word database；sentence vector

DOI： 10.3969/j.issn.1000-3428.2017.05.023

0 概述

微博文本作为典型的短文本，其文本长度一般不超过 140 字，并且微博文本具有很强的随意性，大部分微博文本呈现碎片化表达，很多微博仅有几个字到十几个字，甚至有些微博只有几个表情或标点符号，比如“  ”、“…”等。另外，微博中存在大量新词和网络用语，在不了解发布者心情或者上下文的情况下，大部分微博是无法理解的，给微博处理带来很大难度。因此，微博文本与传统媒体文本的处理方法之间有着较大的区别。与此同时，作为新媒体短文本的代表，微博文本中蕴含着巨大的研究价值和商业价值，有必要对微博文本的处理方法进行深入研究。

Word2vec[1-2]是一个将词语转换成词向量的工具，不同于传统的向量空间模型（Vector Space Model，VSM），其使用的是 Distributed Representation 的词向量表示方式。它把对文本内容的处理简化为向量空间中的向量运算，并通过计算向量空间上的相似度来表示文本语义上的相似度，极大地提高了效率。在大量的短文本实验中，Word2vec 表现出了优秀的处理能力，可以为短文本数据寻求更加深层次的特征表示。Word2vec 训练后的词向量可以处理很多自然语言的工作，如分类、聚类、求同义词、词

基金项目： 国家自然科学基金(61370139)；北京市属高等学校创新团队建设与教师职业发展计划项目(IDHT20130519)。

作者简介： 段旭磊(1991—)，男，硕士研究生，主研方向为中文信息处理、数据挖掘、机器学习；张仰森(通信作者)，教授、博士后；孙祎卓，硕士研究生。

收稿日期： 2016-07-04 **修回日期：** 2016-08-15 **E-mail：** zys@bistu.edu.cn

性分析、语义分析、机器翻译[3]等。

在句向量的研究中,由于句子长度不一,现有方法都是截取句子的部分信息,将句子映射为定长向量。本文在 Word2vec 框架内,提出采用词向量或高维词库映射计算句向量的方法,从而将微博文本采用句向量表示,并与现有流行相似度计算方法进行比较,寻找更适合微博文本的处理方法。另外,根据微博文本稀疏的特点,利用词向量对短文本进行扩展,并验证了扩展后文本相似度计算的效果。

1 相关研究

根据自然语言的粒度大小,可以将其分为字、词、短语、句子、段、节、章和文档等。在一般情况下,字、词是自然语言处理的最小单位,大部分的研究都需要从字、词出发,根据文本的不同领域特点,对文本进行相应处理。微博文本更加接近口语,将微博文本表示成字、词的过程比较困难,对微博文本的分词等预处理更加复杂,也增大了微博文本相似度计算的难度。

文本的相似度计算可以分为基于统计的相似度计算和基于语义的相似度计算。研究文本的相似度,首先是解决如何表示文本的问题。在这个过程中,用 VSM 表示文本的方法普遍受到人们的认可,其在知识表示方法上具有较大优势。在 VSM 中,通过将文本内容形式化为多维空间中的点,从而将对文本内容的处理转换为向量空间中向量的运算,大大降低了问题的复杂度。在将文本用向量表示的过程中,一般需要对文本特征项加权,加权是针对每个关键词对文本特征的体现效果、大小不同而设置的机制,权值计算一般参照 TF-IDF 公式,其中也有借助先验知识来衡量特征的权重[4]。但 VSM 是建立在假定关键词之间线性无关的基础上,其只考虑了词在上下文中的统计特性,没有考虑词本身的语义特性,因此具有很大的局限性。基于此,很多学者结合语义,对文本或者词语的相似度计算进行了大量研究,其中基于知网(HowNet)和基于 WordNet 的研究是语义相似度计算最常见的方法[5-8]。

随着深度学习研究的深入,VSM 模型词之间无关联的缺点也更加突出。在深度学习中,一般用 Distributed Representation 来描述一个词,常被称为“Word Representation”或“Word Embedding”,也就是俗称的“词向量”。词向量由 Hinton 在 1986 年提出[9],后来被 Bengio[10]发展,并由于 Word2vec 的开源而被人熟知。在国内外的研究中,采用 Word2vec 进行文本研究越来越普遍。

Word2vec 的基本思想是通过训练将每个词映射成高维实数向量,通过词之间的距离(如余弦相似度、欧氏距离等)来判断它们之间的语义相似度。Word2vec 核心技术是根据词频采用 Huffman 编码,使得所有词频相似的词隐藏层激活的内容基本一致,出现频率越高的词语,它们激活的隐藏层数目越少,有效地降低了计算的复杂度。Word2vec 的高效性,使其受到研究人员和技术人员的欢迎,正如文献[11]中提出,一个优化的 Word2vec 单机版本训练上千亿词的时间不会超过一天。

Word2vec 与潜在语义分析(Latent Semantic Index,LSI)、潜在狄立克雷分配(Latent Dirichlet Allocation,LDA)的经典过程相比,利用了词的上下文,语义信息更加丰富。图 1 为词向量的学习框架,其通过输入上下文 the,cat,sat 映射成的矩阵,预测第 4 个单词 on。

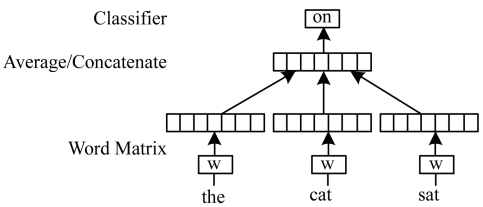

图 1 词向量学习框架

在近期的相关研究中,文献[12]介绍了 Sentence Vector 训练句向量的方法。Sentence Vector 是从未标记数据得到,因此可以应用在没有太多标签的数据上,并且在 Sentence Vector 中每篇文档映射在一个唯一的向量上。另外,Sentence Vector 解决了 Bag-Of-Words 没有 Word 顺序的问题,极大地降低了表示的维度。图 2 为句向量的学习框架。

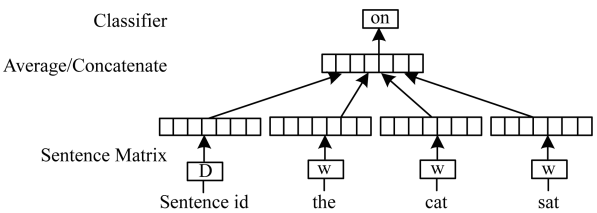

图 2 句向量学习框架

相比于 Word2vec,句向量(Sentence Vector)的区别点有以下 2 点:

1)训练过程中新增了 Sentence id,即训练语料中每个句子都有一个唯一的 id。Sentence id 和普通的 Word 一样,也是先映射成一个向量,即 Sentence Vector。Sentence Vector 与 Word Vector 的维数虽一样,但是来自于 2 个不同的向量空间。在之后的计算中,Sentence Vector 和 Word Vector 累加或者连接起来,作为输出层 Softmax 的输入。在一个句子或者文档的训练过程中,Sentence id 保持不变,共享着

同一个 Sentence Vector,相当于每次在预测单词的概率时,都利用了整个句子的语义。

2)在预测阶段,给待预测的句子新分配一个 Sentence id,词向量和输出层 Voftmax 的参数保持训练阶段得到的参数不变,重新利用梯度下降训练待预测的句子。待收敛后,即得到待预测句子的 Sentence Vector。

目前,利用 Word2vec 进行句向量研究的方法还比较少。本文将提出计算句向量的 3 种新方法,通过词向量或高维词库映射的方法等计算出句向量,从而取代 VSM 的表示方法,计算句子相似度。

此外,在针对微博等短文本的相似度研究中,由于文本较短,研究者一般采用知网等外部知识库进行扩展,如将知网和同义词词林结合进行扩展的研究,但这些方法适应性较差,大多数不能取得很好的效果。当然还有采用词向量模型进行特征空间优化的同义词扩展方法,如北京邮电大学的张为泰采用 Wordnet 和词向量结合的方法对短文本进行扩展,并利用扩展后的语料进行分类实验,取得了很好的效果[13]。然而,这些研究都停留在对短文本的表面扩展,没有具体分析如何将扩展后的短文本采用句向量表示,从而对整个句子进行分析。

2　算法设计

算法研究的基础是 Word2vec 训练的领域词向量,而文中的句向量是指每条微博文本处理后形成的高维向量。本文利用训练的模型,设计了 3 种句向量的计算方法,进行文本相似度的计算。

1)对文本进行扩展后,结合 TF-IDF 表示短文本。

传统的利用 TF-IDF 计算文本相似度的方法,一般都未考虑文本长短对文本相似度结果的影响。但对于微博文本而言,这种计算方法的效果很不理想。针对微博文本高维稀疏的特点,本文采用 Word2vec 对其进行扩展[13],然后利用 TF-IDF 对扩展后的文本进行向量表示,这样就避免了在对生微博文本处理时大部分特征为零的问题,从而提高 TF-IDF 对微博文本相似度计算的适用性。

上文提到的扩展是指将要计算的文本分词、去停用词等预处理后,对文本的词集合进行扩展。TF-IDF中 TF 指词频,即某个词在文本中出现的次数。词语出现的次数越高,表明其在文本中的作用越大。IDF 指逆文档频率,即如果这个词在所有文本集合中出现的次数越多,说明这个词的区分能力越低,越不能反映文本特性。在微博文本的处理过程中如果直接用 TF-IDF 进行表示,稀疏性太大,所以需要对文本进行扩展。算法步骤如下:

步骤 1　将文本预处理后的文本 S_n 和 S_m,采用词集合 S_n' 和 S_m' 表示。

步骤 2　采用 Word2vec 工具扩展文本中的每一个词,在扩展时选取和每个词语相似度最高的词用来扩展,形成新的词集 S_n'' 和 S_m''。

步骤 3　使用 TF-IDF 值分别计算词集 S_n'' 和 S_m'' 中的前 N 个关键词,合并 2 个文本的关键词,形成样本向量词集 $\boldsymbol{S}_v$。

步骤 4　分别计算 S_n'' 和 S_m'' 在样本向量词集 S_v 中的词语的频率,形成词频句向量 $\boldsymbol{S}_{v1}$ 和 $\boldsymbol{S}_{v2}$。

步骤 5　计算 $\boldsymbol{S}_{v1}$ 和 $\boldsymbol{S}_{v2}$ 的余弦相似度,即 $sim(\boldsymbol{S}_{v1},\boldsymbol{S}_{v2})=\cos(\boldsymbol{S}_{v1},\boldsymbol{S}_{v1})$。

2)文本中每个词语的词向量直接相加,得到句向量。

在进行句向量计算的过程中,将预处理后形成的词集合中的每个词利用 Word2vecde 得到词向量,然后将每个词向量相加,用以表示该文本的句向量。

设句子 $\boldsymbol{S}_i$ 有 K_i 个词,$\boldsymbol{S}_i=(V_{i1},V_{i2},\cdots,V_{ic},\cdots,V_{iN})$ 表示原句子经过 Word2vec 处理后得到的句向量分布,$\boldsymbol{W}_i(j)=(V_1(j),V_2(j),\cdots,V_C(j),\cdots,V_N(j))$ 表示第 i 个句子的第 j 个词的词向量,N 表示每个词向量的维度,c 表示 N 维向量中的第 c 个值,则词向量相加后得到的句向量如式(1)所示。

$$\boldsymbol{S}_i=\frac{1}{K_i}\sum_{1\leqslant j\leqslant K_i}\boldsymbol{W}_i(j) \tag{1}$$

得到归一化向量后,文本(句子)的相似度用式(2)表示:

$$sim(S_n,S_m)=\sum_{1\leqslant C\leqslant N}V_{nC}\times V_{mC} \tag{2}$$

算法步骤如下:

步骤 1　将文本预处理后的文本 S_n 和 S_m,采用词集合 S_n' 和 S_m' 表示。

步骤 2　采用 Word2vec 获得文本中每一个词的词向量。分别将每个文本中每个词的词向量相加,即采用式(1)获得句向量 $\boldsymbol{S}_{v1}$ 和 $\boldsymbol{S}_{v2}$。

步骤 3　利用式(2)计算 $\boldsymbol{S}_{v1}$ 和 $\boldsymbol{S}_{v2}$ 的相似度,即 $sim(\boldsymbol{S}_{v1},\boldsymbol{S}_{v2})$。

3)构建高维词库,利用词库映射句子语义,从而形成句向量。

据统计,在文章的写作中,几千甚至几百个词就足以描述大多数文章的内容,并且常用词语不会超过一万。基于这个思想,本文提出采用词库映射的方法,将微博文本的意思映射到常用的几千个词中。

首先构建一个具有 N 个词的常用词词库,并附加每个词的词向量,从而将每一个文本映射成一个 N 维的向量,向量的每一维是词库中与该维对应的词语和文本中每个词语的相似度的最大值。比如“我喜欢用苹果手机”这句话,对应的向量会在三星手机、诺基亚、小米手机、电脑等词上有比较高的得分。而如“我喜欢吃苹果”,对应的向量会在苹果树、

荔枝、香蕉等词上有较高得分。在词语相似度计算中,此思想相较于基于上下文的方法,将赋予每个词语更加丰富的语义分布。这种做法对于解决 Bag Of Words 的稀疏性问题效果较好。

使用 Word2vec 模型计算 2 个词语 $\boldsymbol{X}_z$ 和 $\boldsymbol{Y}_1$ 之间的相似度,$\boldsymbol{X}_z$ 表示 N 维词库中第 z 个词的词向量,$\boldsymbol{Y}_1$ 表示第 i 个文本中第 1 个词向量,设文本长度为 L(L 表示句子经过分词等预处理后,具有 L 个词)。

$$S_i = \{\max_{1\leqslant l\leqslant L}(X_1\times \boldsymbol{Y}_l^{\mathrm{T}}),\max_{1\leqslant l\leqslant L}(X_2\times \boldsymbol{Y}_l^{\mathrm{T}}),\cdots,\max_{1\leqslant l\leqslant L}(X_N\times \boldsymbol{Y}_l^{\mathrm{T}})\} \tag{3}$$

最后采用余弦距离表示 2 个向量之间的相似度,如式(4)所示。

$$sim(\boldsymbol{S}_i,\boldsymbol{S}_l)=\cos(\boldsymbol{S}_i,\boldsymbol{S}_l) \tag{4}$$

算法步骤如下:

步骤 1 将文本预处理后的文本 S_n 和 S_m,采用词集合 S_n' 和 S_m' 表示。

步骤 2 采用式(3),由 Word2vec 分别计算词库中的每个词与文本中每个词的相似度,将词库中的每个词与文本中词语的最大相似度作为句向量的一维,形成句向量 $\boldsymbol{S}_{v1}$ 和 $\boldsymbol{S}_{v2}$。

步骤 3 计算和的余弦相似度,即 $sim(\boldsymbol{S}_{v1},\boldsymbol{S}_{v2})=\cos(\boldsymbol{S}_{v1},\boldsymbol{S}_{v1})$。

3 实验与结果分析

3.1 微博文本预处理

实验中的微博语料来自新浪微博平台,总计爬取约 1 亿条微博及 1 亿条微博评论。在前期的文本处理中,拣选长度在 5 个字符以上的微博,并将每条微博的评论合并后同样截取字符长度超过 5 的微博评论文本,共计得到微博及评论集合总条数:43 225 049条,最后采用 ICTCLAS 将两者相加后的文本进分词处理。由于微博文本的特殊性,采用与处理普通文本时相同的方法处理微博文本,通常难以奏效[14]。在对微博短文本进行分词时自己构建了针对微博分词的具有 80W 词的用户词典,其中包括了大量的新词、俚语、地名、机构名和微博常用词,例如“拉黑”“咸吃萝卜淡操心”“谷歌”“耐你”“呵呵”“笔记君”等。微博分词中,用户词典的使用大大提高了分词的准确性。分词前文本及分词后文本示例如下:

原句:“明明可以靠脸吃饭,偏偏要靠才华,笔记君,本宝宝爱死你了,哈哈哈哈!”

一般分词:“明明 可以 靠 脸 吃饭,偏偏 要靠 才华,笔记 君,本 宝宝 爱死 你 了 哈哈哈哈 !”

加用户词典后分词:“明明 可以 靠脸 吃饭,偏偏 要靠 才华,笔记君,本宝宝 爱死 你了,哈哈哈哈 !”

对微博文本进行分词及去停用词等预处理后,采用成熟的语料开始训练 Word2vec 模型。一个好的词向量模型对后续的研究至关重要[15],在大量的实验中发现,要想得到一个较高的 Word2vec 模型,训练语料的选取尤为重要。本文所训练的模型将用于微博文本处理,所以采用的原语料全部来自微博,并且使训练语料的规模尽可能得大,本文所采用的语料达到10 GB。另外 Word2vec 训练参数的选择也比较重要,特别是 VectorSize 和 WindowSize 的选择,本文分别选择 100 和 8。

训练 Word2vec 模型后,采用 analogy 方法和对微博常用语求取相似度词语集合的方法去验证本文模型的好坏,如表 1 所示,其中,analogy(“国王”“女王”“女人”)表示国王 - 女王 + 女人 = ?。

表 1 Word2vec 模型测试

方法	TOP1	TOP2	TOP3
analogy(“国王”“女王”“女人”)	男人:0.992	男人心:0.990	绅士的:0.983
similar(“毛泽东”)	毛主席:0.920	开国:0.875	邓小平:0.874

从表 1 的结果可以看出,在 Word2Vec 的经典示例中,本文的模型表现较好,能够很好地得到结果:“男人”,并且其结果词集总体效果也更好。

实验中 A1,B1,A2,B2,A3,B3 为选出的用来在计算文本相似度的示例文本。另外,针对 TF-IDF 方法,特别地选出一个长文本进行比较测试。长文本组合为 A4,B4。

A1:“今天天气真好,很适合出去运动”

B1:“今天天气可真好,万里无云,非常适合出去运动”

A2:“#巴黎恐怖袭击#【[话筒]扩散周知:爆炸瞬间的逃生法则】伴随巨响,爆炸导致的冲击波、破片及高温可能瞬间致命,愿灾难和恐怖袭击别再发生!但一旦遭遇危机,正确的逃生法则却能换来一线生机!戳图了解↓转给更多人!”

B2:“#巴黎恐怖袭击#【扩散!每个人都应知道的反恐防恐常识】13 日,巴黎遭遇严重恐怖袭击,数百人死伤。愿灾难不再,但我们都应了解防恐常识。怎样识别可疑人物和爆炸物?遇到刀斧砍杀袭击如何躲避?公共汽车上遇到纵火恐怖袭击怎么逃生?遇到枪击如何选掩蔽物?戳图了解↓为生命安全转!人民日报”

A3:”#背包去旅行#当我再次踏上黄山,带着朝圣的心情,祈求能在岁月流转间遇见她的另一张面容。还好得到了幸运之神的眷顾,让我看到了醉美的日出。从祈祷能有日出到凌晨摸黑去等待,从在山顶悬崖边焦急等待到太阳升起时的震撼瞬间,无不深深地印刻在我的脑海中。#一片叶子# @ 中国摄影师联盟 @ 新浪旅游”

B3:“#交大 120 周年校庆#美哉交通大学百载华章,壮哉西迁精神万世荣光,风云两甲子,弦歌三世纪,祝交大双甲子生日快乐! @ 西安交通大学”

A4:连巴黎都被恐怖袭击了,恐怖分子真是越来越猖狂。都是无辜百姓,真是太可恶。真的看完热门评论顿时觉得这世界不会再好了。还有好些人质呢!这些人太丧心病狂了。听说 那个区中国人多,希望平安。。Pray for Paris[祈祷]?[祈祷]?[祈祷]?。真庆幸能生在我大天朝。。旅行成了人类最奢侈的事能保命的地方太少了。KevinWangCHN 你要注意安全哦。。昨天才看完《茶花女》,今天就发生恐怖袭击,你叫我对巴黎情何以堪?。恐怖分子该死。\[恐怖分子#恐怖袭击#注意安全#\]

B4:强烈谴责暴行,恐怖分子是全世界人民的公敌,恐怖分子应从这个地球消失,美国你有 时间搅和南海,为什么没时间做做正事,你不是要当老大吗?那就带头行动起来呀?。Pray For Paris[祈祷]?。每次看到这种我都庆幸自己在中国,太可怕[泪][泪][泪]。惟愿平安[悲伤]。还是在中国安全一点。好心疼,好好的地方怎么就变这样了。想起去年还优哉游哉的在街道上走着呢~人的心一旦被恶魔侵蚀了后果真恐怖。在法留学[狗]已经吓傻恐怖分子处死了 3 个中国留学生 好难过[泪][泪]。比恐怖袭击更恐怖的罪行多了,穆斯林被大规模屠杀的时候怎么不来说说信仰问题?黑人被屠杀的时候怎么不说信仰问题?他们有错么?他们不也是无辜的生命?那么多恐怖罪行,怎么不提信仰问题?反对恐怖袭击,保护和平这是全世界 都该做的事,我也恨不得这些恐怖分子全部死光,但是为什么要上升到宗教信仰上。回复 江左萌萌主江左没狼 说的好像被东突分子杀害的都活着一样……\[全世界人民#强烈谴责#恐怖袭击#\]

3.2　实验数据

在下列 4 个实验中,实验 1 ~ 实验 3 是根据第 2 节算法生成的实验数据,实验 4 为根据文献[10]进行实验后得到的结果,用以对比本文的 3 种算法。

1)实验 1

采用第 2 节算法 1 对例句 A1 ~ A4 及对应的 B1 ~ B4 进行扩展,然后采用 TF-IDF 方法进行微博文本相似度计算,得到的结果如表 2 所示。

表 2　实验 1 相似度结果

模型	A1/B1	A2/B2	A3/B3	A4/B4
(数据 1)TF-IDF	0	0	0	0.739
(数据 2)扩展 + TF-IDF	0	0	0	0.842

从实验结果可以看出,这个方法不能处理短文本,只能处理特别长的文本,并且效果也不是很好。在实验中根据 TF-IDF 的特点,采用文本扩展的方法,用来提高文本之间相似度计算的效果,但是扩展功能的增加,不能改变这种方法对短文本较差的结果,反而对于长文本,对其文本相似度计算的效果有所提高。

2)实验 2

采用第 2 节算法 2 对例句 A1 ~ A4 及对应的 B1 ~ B4 进行预处理,并求取词向量后将词向量相加形成句向量,计算微博文本之间的相似度后得到的结果如表 3 所示。

表 3　实验 2 相似度结果

模型	A1/B1	A2/B2	A3/B3	A4/B4
(数据 1)词向量相加的模型	0.289	0.205	0.083	0.960

从实验结果可以看出,此方法比实验 1 方法效果好,其中对长文本的效果明显较好,而且短文本相似度计算的效果比实验 1 好,但是效果并不理想。

3)实验 3

利用常用词词库计算 2 个微博文本之间的相似度。分别计算 N 维中每一维与分词后文本中每个词的相似度,取每一维与文本分词后词语相似度的最大值,形成新的 N 维向量,代表每一个文本,利用此向量的距离计算 2 个文本之间的相似度。其中,N = 3 000,5 000,其结果如表 4 所示。

表 4　实验 3 相似度结果

模型	A1/B1	A2/B2	A3/B3	A4/B4
(数据 1)N = 3 000	0.986 0	0.963	0.371	0.970
(数据 2)N = 5 000	0.985 5	0.882	0.423	0.971

从实验结果可以看出,该方法与其他方法相比,长文本相似度效果有所提高。而且这种方法计算出来的相似度,对相似的文本比较有效果,对差别大的文本,区分也较明显。通过实验可以发现,文本相似度的计算和词库息息相关。分别采用 3 000,5 000 词库进行计算对比,发现词库在 3 000 左右时的效果最好。而这 3 000 词库的选取,则是采用 Hadoop 分布式计算从整个实验语料中提取了词频最高的前 3 000 个词语。

4)实验 4

本文实验采用文献[10]方法与前 3 个实验进行对比,直接采用预处理后语料对微博文本训练句向量,形成句向量模型,采用句向量模型计算句子之间

相似度。实验采用的语料规模,即文本的数量分别为 2 000,10 000,100 000。实验结果如表 5 所示。

表 5 实验 4 相识度结果

模型	A1/B1	A2/B2	A3/B3	A4/B4
(数据 1) Sentence-2000	0.988	0.781	0.897	0.574
(数据 2) Sentence-10000	0.970	0.538	0.380	0.516
(数据 3) Sentence-100000	0.878	0.274	0.032	0.361

实验 4 的实验结果表明,该方法短文本效果比长文本效果好,但短文本的总体效果也不理想。对所有短文本的相似度值普遍较高,长文本的相似度值普遍较低。而且文本相似度的计算与整体语料的大小息息相关,并且存在不确定性,从实验结果中可以看出,随着实验语料的增大,文本之间相似度的效果有所波动,但是总体效果也不理想,存在较大不确定性。从实验数据来看,其效果不如实验 3。

最后将 4 组实验的数据进行汇总形成表 6,经过将每组实验示例文本进行反复对比,及志愿者对每组示例文本相似度值的评分,赋予每组示例文本一个标准相似度值。从表 6 的汇总结果可以看出,实验 3 的数据最接近标准值。特别说明的是由于本文所有实验的基础是建立在上亿条微博数据上,其算法计算量较大,特别是实验 3。也正因为此,本文采用了通用并行计算框架 SPARK 来存储分析数据。

表 6 实验结果汇总

实验	A1/B1	A2/B2	A3/B3	A4/B4
实验 1(数据 1)	0.000 0	0.000	0.000	0.739
实验 1(数据 2)	0.000 0	0.000	0.000	0.842
实验 2(数据 1)	0.289 0	0.205	0.083	0.960
实验 3(数据 1)	0.986 0	0.963	0.371	0.970
实验 3(数据 2)	0.985 5	0.882	0.423	0.971
实验 4(数据 1)	0.988 0	0.781	0.897	0.574
实验 4(数据 2)	0.970 0	0.538	0.380	0.516
实验 4(数据 3)	0.878 0	0.274	0.032	0.361
标准值	0.999 0	0.980	0.300	0.980

4 结束语

本文提出了计算微博文本相似度的 3 种方法,从实验结果可以看出,在这 3 种方法中,使用词库映射作为文本向量的方法效果较好,且此方法与直接计算句向量进行文本相似度计算的方法相比,效果也较佳。在计算词语相似度时,由于一个词的义项较少,因此用 Word2vec 这种基于共现的方法能取得较好的效果。但本文方法也存在一定的问题,对于文本由于其精细度较高,即使绝大部分词语相同,整句话的意思也很可能不相同。所以文本级别的训练仍然是一个公认的难题。下一步将继续改进高维词库映射方法,当词库中的词与微博文本中多个词相似度最大且相等时,应选择哪个词的义项作为合适的词语语义。

参考文献

[1] Mikolov T,Chen K,Corrado G,et al. Efficient Estimation of Word Representations in Vector Space[C]//Proceedings of ICLR'13. New York,USA:ACM Press,2013:1-12.

[2] Mikolov T,Yih W T,Zweig G. Linguistic Regularities in Continuous Space Word Representations[C]//Proceedings of HLT-NAACL'13. New York,USA:ACM Press,2013:236-248.

[3] Feng S,Liu R,Wang Q,et al. Word Distributed Representation Based Text Clustering[C]//Proceedings of International Conference on Cloud Computing and Intelligence Systems. Washington D. C.,USA:IEEE Press,2014:213-225.

[4] 刘 铭,吴 冲,刘远超,等. 基于特征权重量化的相似度计算方法[J]. 计算机学报,2015,38(7):1420-1433.

[5] 朱嫣岚,闵 锦,周雅倩,等. 基于 HowNet 的词汇语义倾向计算[J]. 中文信息学报,2006,20(1):14-20.

[6] 黄承慧,印 鉴,侯 昉. 一种结合词项语义信息和 TF-IDF 方法的文本相似度量方法[J]. 计算机学报,2011,34(5):856-864.

[7] 徐琳宏,林鸿飞,杨志豪. 基于语义理解的文本倾向性识别机制[J]. 中文信息学报,2007,21(1):96-100.

[8] 王小林,王 东,杨思春,等. 基于《知网》的词语语义相似度算法[J]. 计算机工程,2014,40(12):177-181.

[9] Hinton G E. Learning Distributed Representations of Concepts[C]//Proceedings of the 8th Annual Conference on Cognitive Science Society. Amherst,USA:[s.n.],1986:1-12.

[10] Bengio Y,Schwenk H,Senécal J S,et al. A Neural Probabilistic Language Model[J]. Journal of Machine Learning Research,2003,3(6):1137-1155.

[11] Mikolov T,Sutskever I,Chen K,et al. Distributed Representations of Words and Phrases and Their Compositionality[J]. Advances in Neural Information Processing Systems,2013,26(1):3111-3119.

[12] Le Q V,Mikolov T. Distributed Representations of Sentences and Documents[C]//Proceedings of the 31st International Conference on Machine Learning. Washington D. C.,USA:IEEE Press. 2014:1188-1196.

[13] Zhang W,Xu W,Chen G,et al. A Feature Extraction Method Based on Word Embedding for Word Similarity Computing[J]. Communications in Computer & Information Science,2014,496(1):160-167.

[14] 张剑峰,夏云庆,姚建民. 微博文本处理研究综述[J]. 中文信息学报,2012,26(4):21-27,42.

[15] Lai S,Liu K,Xu L,et al. How to Generate a Good Word Embedding?[J]. IEEE Intelligent Systems,2015,31(6):5-14.

编辑 索书志