基于 Word2vec 的句子语义相似度计算研究

李 晓[1] 解 辉[2] 李立杰[3]

(安阳师范学院计算机与信息工程学院 安阳 455002)[1] (清华大学计算机科学与技术系 北京 100084)[2]

(北京理工大学软件学院 北京 100081)[3]

摘 要 word2vec利用深度学习的思想,可以从大规模的文本数据中自动学习数据的本质信息。因此,借助哈尔滨工业大学的LTP平台,设计利用word2vec模型将对句子的处理简化为向量空间中的向量运算,采用向量空间上的相似度表示句子语义上的相似度。此外,将句子的结构信息添加到句子相似度计算中,并就特殊句式对算法进行了改进,同时考虑到了词汇之间的句法关系。实验结果表明,该方法更准确地揭示了句子之间的语义关系,句法结构的提取和算法的改进解决了复杂句式的相似度计算问题,提高了相似度计算的准确率。

关键词 句子相似度,word2vec,词向量,语义,句法结构

中图法分类号 TP391.1 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.09.048

Research on Sentence Semantic Similarity Calculation Based on Word2vec

LI Xiao[1] XIE Hui[2] LI Li-jie[3]

(School of Computer and Information Engineering, Anyang Normal University, Anyang 455002, China)[1]

(Department of Computer Sciences and Technology, Tsinghua University, Beijing 100084, China)[2]

(School of Software, Beijing Institute of Technology, Beijing 100081, China)[3]

Abstract Using the idea of deep learning, word2vec can automatically learn the essential information of data from large-scale text data. Therefore, with the help of LTP platform of Harbin Institute of Technology, based on the word2vec model, the processing of the sentence is simplified as a vector in the vector space algorithm, and the similarity of vector space represents the sentence semantic similarity. In addition, the sentence structure information is added to the sentence similarity calculation, the algorithm are improved on the special sentence pattern, and the syntax relationship between words is taken into account. The experimental results show that this method is more accurately to reveal the semantic relations between sentences, syntactic structure and improved extraction algorithm also solve the problem of computing the similarity of complex sentences, finally improve the accuracy of the similarity calculation.

Keywords Sentence similarity, Word2vec, Distributed representation, Semantic, Syntactic structure

1 引言

句子相似度计算是中文信息处理中一项很重要的基础研究工作,其技术已广泛应用于很多领域,如从问题数据库中选取出与用户问句最相似的问句的智能答疑系统,从实例集中检索出与待翻译语句最相似的源语言实例的机器翻译,以及文本自动分类和文本检测查重等[1-4]。这些应用系统的准确性很大程度上取决于句子相似度计算的准确性。因此,提高句子相似度计算的准确性是当前研究要解决的首要问题。

纵观历史,统计模型已成为自然语言处理研究领域的主流,但以往自然语言处理领域的统计学习方法大多属于浅层模型。使用浅层模型需要靠人工经验从数据中抽取特征。统计模型主要负责分类或预测,对数据的表示学习能力较弱,系统性能的好坏主要取决于人工提取特征的质量。因此,研究人员不得不在数据的标注、观察和特征提取上耗费大量的精力,随着研究的深入,研究人员还要对不同的任务提取不同的特征,这对于实现越来越多的大规模文本处理及文本语义处理是不现实的。

深度学习(Deep-Learning)在语音和图像领域上表现出的优异的表示学习能力,使得研究人员对深度学习在自然语言处理领域的应用产生了浓厚的兴趣,并逐渐将其应用于自然语言处理的各种任务中[5-7]。Deep-Learning通过构建具有很多隐层的机器学习模型和海量的训练数据来学习数据中更加有用的特征,而且学习到的特征能够更加准确地反映数据的本质。深度学习凭借对数据的表示学习能力,无须人工过多干预,能从大规模文本数据中自动学习数据的本质信息,从而提高文本表示的质量,在相关任务上取得了优异的性能。因此人们也期待着深度学习能够另辟蹊径,给文本相似度计

到稿日期:2016-08-12 返修日期:2016-12-24 本文受国家自然科学基金:面向甲骨学知识图谱的实体发现及语义关系挖掘研究(U1504612),河南省高等学校重点科研项目计划:基于语义向量空间模型的中文文本相似度计算研究(16A520037)资助。

李 晓(1982-),女,硕士,讲师,CCF会员,主要研究方向为中文信息处理;解 辉(1981-),男,博士生,主要研究方向为计算网络、机器学习;李立杰(1977-),男,博士,讲师,主要研究方向为虚拟现实、机器学习,E-mail:joylx@163.com。

算研究带来全新的突破。

2　相关工作

现阶段，大多数的汉语句子相似度计算方法都局限于一定的模型。向量空间模型（VSM）提取句子中的动词、名词、形容词等实词，以向量的形式来表示句子；但向量空间法把句子中的词语看作相互独立的特征，忽略了词语之间的关联以及句子的序列关系、位置关系对句子语义的影响。编辑距离法将句子看作词串，通过计算从源词串 A 到目的词串 B 之间所需要的最少编辑次数得到句子的相似度，虽然考虑了句子中词语的顺序关系，但是这种关系明显机械化，无法真正体现词语在不同位置上的语义关联。后来人们通过对句子中的同义词或近义词进行扩展，一定程度上解决了召回率低的问题，但是扩展后难免引入噪声，尤其是句子中含有多义词时，例如“打酱油”和“打毛衣”的“打”字，董振东先生的知网（hownet）中对这类汉字给出了很好的语义关系解释，通过 hownet 中的义元树结构可以对词语粒度的形似度进行度量，很好地解决了噪声问题。同时为了更深入地理解句子的结构，人们利用依存句法分析对语言单位内成分之间的依存关系进行有效揭示，很好地解决了词语的位置关系对句子相似度计算的影响；但由于依赖有效搭配对的匹配，无法考虑所有的句法特征，势必造成误差，目前依存句法分析技术还有待进一步完善。

3　word2vec 模型原理

word2vec 主要采用 CBOW（Continuous Bag-of-Words Model）和 Skip-Gram（Continuous Skip-Gram Model）两种模型。无论是 CBOW 模型还是 Skip-Gram 模型，都是以 Huffman 树作为基础。Huffman 树中非叶节点存储的中间向量的初始化值是零向量，叶节点对应的单词的词向量是随机初始化的[8]。CBOW 的目标是根据上下文来预测当前词语的概率，而 Skip-Gram 恰好相反，它是根据当前词语来预测上下文的概率，如图 1 所示。这两种方法都利用人工神经网络作为它们的分类算法。起初，每个单词都是一个随机 N 维向量，经过训练之后，利用 CBOW 或者 Skip-Gram 方法获得每个单词的最优向量。

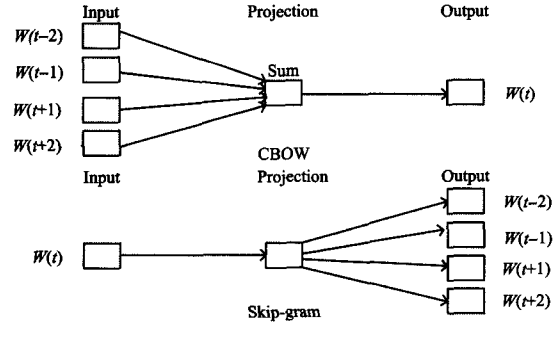

图 1　word2vec 的两种模型

3.1　CBOW 的模型结构

CBOW 的模型结构如图 2 所示，其中 CBOW 一共由 3 层构成，第一层称为输入层，输入的是若干个词的词向量，词的个数与随机抽取的窗口大小 c（c 表示当前词向前或向后覆盖词的个数）有关；中间称为映射层，输入的是若干词向量的累加和；第三层为输出层，输出的是方框里面的二叉树，它是以语料库中各词在语料中出现的次数为权值构造出来的 Huffman 树，Huffman 树的所有非叶节点与映射层的节点相关联（如图 2 所示）。所有叶节点代表语料库里的所有词（假设语料库词语个数为 v），每个叶节点对应一个词的词向量，每个非叶节点也是一个向量，但不代表某个词，而是代表某一类别的词。输入的若干词向量其实与 Huffman 树中的某几个叶节点是一样的，当然输入的词与它们最终输出的词未必是同一个词（基本不会是同一个词），只是这几个词与输出的词往往有语义上的关系[9]。CBOW 网络结构的功能是通过给定的上下文，计算目标词在当前网络结构下的概率。

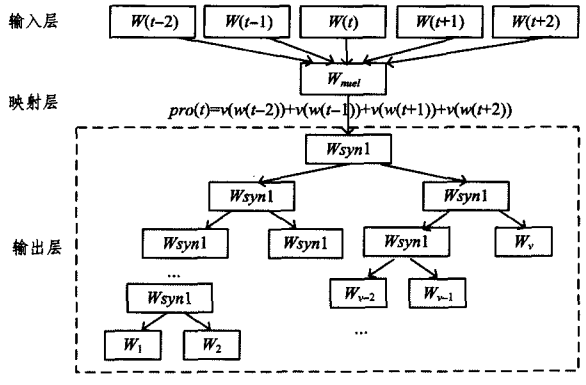

图 2　CBOW 的模型结构图

3.2　Skip-Gram 的模型结构

Skip-Gram 的模型结构如图 3 所示，其中 W_i 是之前计算得到的词，词 W_i 与 Huffman 树直接连接，此 CBOW 模型结构少了映射层。例如，判断“小明 喜欢 吃 甜甜 的 苹果”这句话是否为自然语言时，如果计算到“吃”这个词，同样随机抽到 $c=2$，对“吃”这个词需要计算的概率有 p（小明|吃）、p（喜欢|吃）、p（甜甜|吃）和 p（的|吃）共 4 个。在计算 p（小明|吃）这个概率时，要用到图 3 中的 Huffman 树，假设“小明”一词位于 Huffman 树根节点的右孩子的最左边的节点，即图 3 中的叶子节点 W_{v-2}；再假设从根节点到该叶子节点路径上的 3 个非叶节点分别为 a,b,c，“吃”这个词的词向量设为 d，那么 p（小明|吃）的概率可以用下式计算[10]：

$$p(\text{小明}|\text{吃})=(1-\sigma(a\cdot d))\cdot\sigma(b\cdot d)\cdot\sigma(c\cdot d)$$

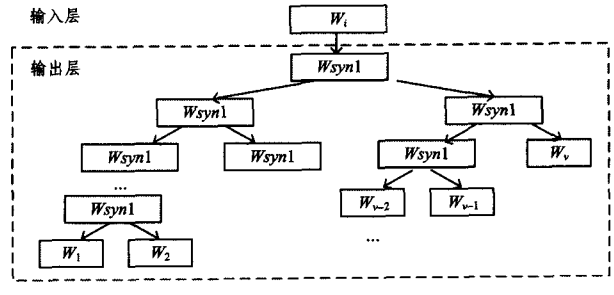

图 3　Skip-gram 的模型结构图

用同样的方法计算 p（喜欢|吃）、p（甜甜|吃）、p（的|吃），再将这 4 个概率连乘便得到“吃”这个词的上下文概率。将整句话所有词的概率都计算出来后连乘，即得到了这句话是自然语言的概率。这个概率如果大于某个阈值，则认为是正常的语言，否则认为不是自然语言，要将其排除掉。

3.3 word2vec 模型的使用

word2vec 模型在给定的语料库上训练 CBOW 和 Skip-Gram 两种模型，然后输出得到所有出现在语料库上的单词的词向量表示。基于得到的单词词向量，可以计算词与词之间的关系，如词语相似性、语义关联性等。

考虑到语料库的选择，在训练 word2vec 模型时使用中文维基百科的语料，最终得到各个词的词向量。

4 基于 word2vec 的句子语义相似度计算研究

4.1 基于 LTP 的句法结构分析

句子是由词或词组按照一定的语法结构组成的表达一个完整意思的语言单位[11]。句子作为一个整体，其相似性建立在部分相似性的基础上，因此从另一个角度讲，句子的相似性计算要归结到词语的相似性计算上。考虑到中文句子的句法结构特性，在利用 word2vec 模型得到词语相似度的基础上，将句子的结构信息添加到句子相似度计算中，以期在语义层面上获得更好的句子相似度计算结果。本文借助哈尔滨工业大学的语言技术平台(Language Technology Platform，LTP)作为句子结构分析的工具，帮助确定中心词和句子成分。LTP 平台是一个基于云计算技术研发的中文自然语言处理服务平台，提供了一套包含分词、词性标注、语法依存标记等自底向上的丰富、高效、高精度的中文自然语言处理模块，为本文计算句子相似度提供了良好的基础[12]。

句子的组成部分叫做句子成分，词与词组构成句子成分。句法结构分析就是要分析出句子成分及其结构关系，句子结构在经过句子成分分析之后一目了然，有利于我们对句子相似度进行基于部分的匹配计算。本文结合前期在句法结构上对句子相似度计算研究的成果[13]，利用 word2vec 模型得到词语语义相似度，最终计算句子的相似度，公式如下：

$$Sim(S_1,S_2)=\beta_1\cdot\beta_2$$

$$\beta_1=k\cdot\lambda\cdot\gamma$$

$$\beta_2=\alpha_1 Sim_1(B_1,B_2)+\alpha_2|Sim_2(B_1,B_2)|+\alpha_3 Sim_3(B_1,B_2) \tag{1}$$

$Sim(S_1,S_2)$的值由两部分决定，β_1 表示相似度调节系数，β_2 表示语义相似度值。β_1 包含 3 个具体参数：k,λ,γ。

(1)k 为句型调节系数，将疑问句对祈使句、陈述句、感叹句的调节系数设置为 0.1，其他句型间的调节系数设置为 0.5，句型相同时 k 为 1。

(2)λ 为句子成分系数，即句子划分的成分不等时的调节系数，其值设为 $2*i/(m+n)$，m 和 n 分别表示 S_1 和 S_2 所含的成分个数，i 为 S_1 和 S_2 中相对应成分的个数。

(3)γ 为否定系数，即两个句子中明显出现 S_1 与 S_2 的谓语中心词是反义词或对义词，或者 S_1 相对 S_2 的谓语中心词前有“不”的情况，则将 γ 的值设置为 -1，因此句子相似度为 -1 时说明两个句子的意思相反。

(4)由于一个句子经过句法分析被划分为 3 个部分，因此 β_2 的值由 $Sim_1(B_1,B_2)$，$Sim_2(B_1,B_2)$，$Sim_3(B_1,B_2)$ 3 部分构成，分别表示主语成分相似度、谓语成分相似度、宾语成分相似度；$\alpha_1,\alpha_2,\alpha_3$ 分别表示 3 个部分的权重系数，本文根据句子结构成分的贡献度和实践经验将其设置为 0.3，0.5，0.2。

4.2 融合 word2vec 和句法分析的句子语义相似度计算研究

本文开发句子相似度计算程序的环境为：Python3.5 和 anaconda3，安装了哈工大分词工具的 python 版本 pyltp[1)]，主程序中主要调用了 pyltp 中的 Parser，Postagger，Segmentor，以及 gensim 工具包的 word2vec 模型[12]。

本文研究的句子语义相似度计算的算法思路如下：

(1)首先利用 pyltp 工具对维基百科中文语料进行分词，并将分词结果处理成 word2vec 模型的训练格式；

(2)使用 word2vec 模型训练分好词的语料，得到训练好的词向量文件；

(3)由于 pyltp 分别返回了分词结果、标注结果和语义依存结果，为了程序的方便，我们利用 pyltp 返回的结果自建了一个以 python 数据结构为基础的 list 树(见图 4)，以便对两个句子的成分进行对比。

图 4 list 树结构

list 树的代码如下所示。

```
树的节点结构：data，children(list 类)
class node:
    def_init_(self,data):
        self._data = data
        self._children = []
#返回节点数据
    def getdata(self):
        return self._data
#返回节点孩子 list
    def getchildren(self):
        return self._children
#将 node 加入到该 node 的节点
    def add(self,node):
        ##if full
          self._children.append(node)
#找到有传入 data 值的子节点
    def go(self,data):
        for child in self._children:
```

1) http://ltp.readthedocs.io/zh_CN/latest/otherlanguage.html

```
    if child.getdata() == data:
        return child
    return None
```

(4)综合句法结构信息和 word2vec 训练的词语语义相似度，使用式(1)计算句子的相似度。

5　实验与结果分析

本文用 python 语言开发了一个基于 word2vec 模型的计算句子语义相似度的程序，并随机选取了 20 组句子，每组两个句子进行相似度计算，运行结果如图 5 所示。选取了北京大学计算语言研究所提出的基于关键词的方法[14]和何维老师提出的基于表层相似度和词序相似度的计算方法[15]对这 20 组句子进行相似度计算，着重选取了实验中具有代表性的部分数据与本文方法进行对比分析，实验结果如表 1 所列。同时，请专业人士对 20 组句子进行相似度判断，并给出每组句子的相似度范围。对 3 种方法的句子相似度计算的准确率进行了统计，其结果如表 2 所列。

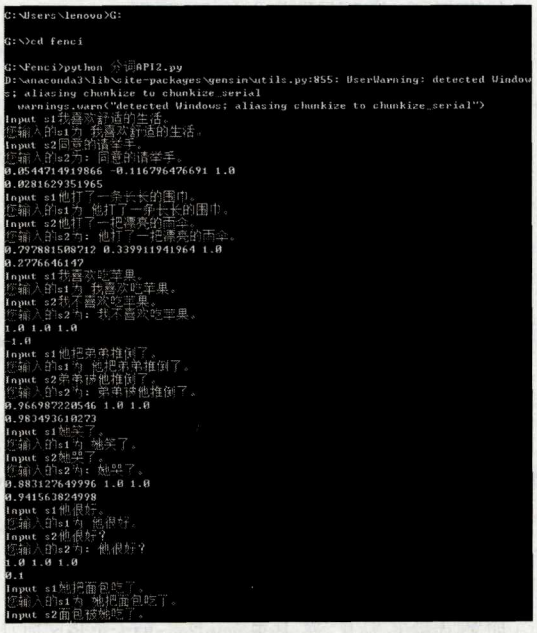

图 5　实验结果数据示例

表 1　部分实验结果数据对比

	测试句子	文献[14]	文献[15]	本文方法
1	S1：我喜欢舒适的生活。 S2：同意的请举手。	0.0000	0.0000	0.0282
2	S3：他打了一条长长的围巾。 S4：他打了一把漂亮的雨伞。	0.5000	0.6625	0.2777
3	S5：我喜欢吃苹果。 S6：我不喜欢吃苹果。	0.8889	0.7368	−1.0000
4	S7：他把弟弟推倒了。 S8：弟弟被他推倒了。	0.7500	0.5400	0.9835
5	S9：她笑了。 S10：她哭了。	1.0000	0.7750	0.9416
6	S11：他很好。 S12：他很好?	1.0000	1.0000	0.1000

表 2　句子相似度计算结果的准确率

方法	测试组数	正确组数	准确率/%
文献[14]	20	9	40
文献[15]	20	11	55
本文方法	20	16	80

从实验过程的细节可以看出，借助 word2vec 得到的词向量已经由原始的稀疏表示形式映射到密集表示的低维词向量形式上，而且携带了一定的语义信息(见图 6)，不再是向量分量只有一个“1”的 One-hot Representation 形式，这给计算两个词语之间的相似度提供了丰富的上下文关联，因此可以更好地计算两个句子之间的语义相似度。从计算结果可以看出，“喜欢”和“请”两个词的相似度为 0.16865，与以往计算出来的相似度结果“0”相比，增添了与其他词语的相关性成分。

```
Input s1我喜欢舒适的生活。
您输入的s1为 我喜欢舒适的生活。

Input s2同意的请举手。
您输入的s2为：同意的请举手。
0.0281629351965

In [2]: mod['请']
Out[2]:
array([ 1.96348739, -3.22663999, -1.07050395, -3.30137491, -1.00502408,
       -0.49977663,  1.08756518,  3.00858927, -3.73986673,  0.44330093,
        2.5480659 ,  2.64186025, -3.4921608 ,  2.34468699, -1.50870657,
        1.86926556, -0.60056823, -2.03246546,  0.10938193, -0.36421147,
        1.30957472,  0.6225121 , -0.73322278, -4.36998224,  2.88574815,
        6.48187494,  0.93793148, -5.35592175,  1.7771734 , -0.40862322,
        5.58561373, -3.42259574,  1.09159851, -1.97557902,  0.74295574,
       -0.69552642,  5.43908501,  0.28663486, -2.68257546,  0.81436849,
        2.45999813, -2.66704154,  1.0897485 ,  1.61564362,  3.24834991,
       -0.99515742, -2.97384238,  1.8423636 , -1.78658378, -0.34079382,
        1.22549903, -0.73558378, -2.45614815, -1.37319493,  3.93133903,
        3.74987578, -1.16028941,  3.06066108,  3.59362125, -0.8168456 ],
dtype=float32)

In [3]: mod['喜欢']
Out[3]:
array([ 1.75676191, -5.05543327, -1.93116081, -2.42097402,  0.6137445 ,
       -3.81058645,  3.38029456, -1.65538025, -1.14463854,  1.10332835,
       -0.65637976, -0.60611355,  6.51059198,  0.96624947,  1.35560465,
       -3.3946383 ,  0.66912746,  0.22531117, -0.40936691,  1.06993103,
        5.40843582,  1.08652234, -0.21245304,  0.17234871, -0.8217242 ,
        1.85918331, -0.56214321, -5.42062283,  0.13162704, -1.58285797,
        2.95521545, -2.93522763, -0.83332342,  0.22630067, -3.33455324,
        2.53835082, -1.02196467,  0.40125895,  0.37763473,  2.2121942 ,
        0.93580669,  2.41509867, -8.03343105,  3.63520765, -3.9358778 ,
        5.42568684,  1.61265182,  0.36791575, -0.87318093, -0.7588492 ,
        0.95082986,  2.07892203, -0.21925688, -1.57792616,  1.63127053,
        3.45935464,  4.45526123,  2.77578497,  1.80010414, -1.76833582],
dtype=float32)

In [4]: mod.similarity('请','喜欢')
Out[4]: 0.16865013777153781
```

图 6　词向量表示及词语相似度计算

借助 LTP 平台得到的句子成分分析为本文句子相似度计算提供了较好的成分对应依据，这比只从关键词角度计算句子相似度的方法更具科学性和合理性。数据示例如图 7 所示。

```
Input s1我喜欢舒适的生活。
您输入的s1为 我喜欢舒适的生活。

Input s2同意的请举手。
您输入的s2为：同意的请举手。
0.0281629351965

In [2]: t1
Out[2]:
[[0, ['我', 'r', 2, 'SBV']],
 [1, ['喜欢', 'v', 0, 'HED']],
 [2, ['舒适', 'a', 5, 'ATT']],
 [3, ['的', 'u', 3, 'RAD']],
 [4, ['生活', 'v', 2, 'VOB']],
 [5, ['。', 'wp', 2, 'WP']]]

In [3]: t2
Out[3]:
[[0, ['同意', 'v', 3, 'ATT']],
 [1, ['的', 'u', 1, 'RAD']],
 [2, ['请', 'v', 0, 'HED']],
 [3, ['举手', 'v', 3, 'VOB']],
 [4, ['。', 'wp', 3, 'WP']]]
```

图 7　句子成分对比

其中 t1 和 t2 分别表示由 LTP 分析结果转化而来的两个句子的 list 树结构，这种树结构便于更好地比较两个句子的语义对应关系。

从实验结果可以看出，与文献[14-15]的方法相比，本文方法在计算准确度上有了明显提高，尤其在一些特殊句型上，具体分析如下：

(1)文献[14-15]的方法不能很好地反映句子之间的语义关系，本文算法通过 word2vec 模型训练出的词向量更准确地刻画了词语之间的语义关系，从语言本身的层面揭示了句子之间的相似度。比如对于 S3 和 S4 两个句子中的“打”这个动词，文献[14-15]提出的方法显然将其作为同一个词对待，而本文方法则从语义角度进行了区分，实验结果更符合人们的理解。

(2)针对否定句型和其他特殊句型，本文从句法结构上对算法进行了有效设计，如 S5 和 S6 这组句子，较前两种方法，本文方法从语义解释上能更准确地反映两个句子的相反意思；另外，对于“把”字句和“被”字句句型(如 S7 和 S8)，本文相似度计算结果为 0.9835，非常接近 1，从实验中的其他句子可以看出，“把”字句和“被”字句句型的相似度计算结果基本上都是 1；不同句型之间的相似度(如 S11 和 S12)也明显体现出情感语义的不同，相似度计算更为准确。

(3)从示例中也发现了一些问题，比如实验结果显示 S9 和 S10 的相似度为 0.9416，说明 word2vec 训练出的“哭”和“笑”两个词的词向量是非常接近的，因此相似度很高，但实际上“哭”和“笑”两个词在情感上意思是相反的。我们也验证了类似其他的词语，发现相似度都很高。以上说明 word2vec 对词语的情感分析存在一定的局限性。

结束语 本文采用基于 word2vec 词向量的词语相似度来完成句子相似度的计算，能更准确地反映词语本身的语义关系，这给计算句子语义相似度提供了良好的基础。另外，基于 LTP 平台得到的句法分析结果很好地刻画了句子成分之间的对应关系，为句子相似度计算有效规避了汉语的复杂句式和句法结构对句子相似度计算的影响。实验结果表明本文算法是有效的，在一定程度上为句子语义相似度计算研究开辟了一条新的思路。目前基于 gensim 工具包，很多学者也推出了 sentence2vec 和 doc2vec 的向量表示，接下来将考虑研究句子或短文本的向量表示，以期为句子相似度计算研究更广阔的应用提供可能。

参 考 文 献

[1] ZHANG D, XU H, SU Z, et al. Chinese comments sentiment classification based on word2vec and SVM perf[J]. Expert Systems with Applications, 2015, 42(4): 1857-1863.

[2] ENRÍQUEZ F, TROYANO J A, LÓPEZ-SOLAZ T. An approach to the use of word embeddings in an opinion classification task[J]. Expert Systems with Applications, 2016, 66: 1-6.

[3] YUAN Y, HE L, PENG L, et al. A new study based on Word2vec and cluster for document categorization[J]. Journal of Computational Information Systems, 2014, 10(21): 9301-9308.

[4] SHAREF N M, MARTIN T, KASMIRAN K A, et al. A comparative study of evolving fuzzy grammar and machine learning techniques for text categorization[J]. Soft Computing, 2015, 19(6): 1701-1714.

[5] SONG M, HEO G E, DING Y. SemPathFinder: Semantic path analysis for discovering publicly unknown knowledge[J]. Journal of Informetrics, 2015, 9(4): 686-703.

[6] O'SHEA K. Natural language scripting within conversational agent design[J]. Applied Intelligence, 2014, 40(1): 189-197.

[7] CHONG C C, LIM T Y, SOON L K, et al. Meaning preservation in Example-based Machine Translation with structural semantics[J]. Expert Systems with Applications, 2017, 78: 242-258.

[8] WEI L, LI D M, LIU C C, et al. Study on the construction of heterogeneous resource ontology based on FCA and Word2vec[J]. Information Science, 2017(3): 69-75. (in Chinese)
韦炼，李端明，刘超超，等. 基于 FCA 和 Word2vec 的异构资源本体构建研究[J]. 情报科学，2017(3)：69-75.

[9] YAN L, MA R, LI D, et al. RDF approximate queries based on semantic similarity[J]. Computing, 2017, 99(5): 481-491.

[10] WEI X C, LIN H F. Transfer learning oriented text feature alignment algorithm[J]. Computer Engineering, 2017, 43(2): 215-219. (in Chinese)
魏晓聪，林鸿飞. 面向迁移学习的文本特征对齐算法[J]. 计算机工程，2017，43(2)：215-219.

[11] HUANG R, ZHANG W. Study on Sentiment Analyzing of Internet Commodities Review Based on Word2vec[J]. Computer Science, 2016, 43(s1): 387-389. (in Chinese)
黄仁，张卫. 基于 word2vec 的互联网商品评论情感倾向研究[J]. 计算机科学，2016，43(s1)：387-389.

[12] WANG M W, XU X F, XU F, et al. Word2vec Based Word Alignment Corpus for the Greater China Region[J]. Journal of Chinese Information Processing, 2015, 29(5): 76-83. (in Chinese)
王明文，徐雄飞，徐凡，等. 基于 word2vec 的大中华区词对齐库的构建[J]. 中文信息学报，2015，29(5)：76-83.

[13] ZHANG L, YAN Q, LV X Q. Short Text-oriented Sentiment Refraction Model[J]. Journal of the China Society for Scientific and Technical Information, 2017, 36(2): 180-189. (in Chinese)
张乐，闫强，吕学强. 面向短文本的情感折射模型[J]. 情报学报，2017，36(2)：180-189.

[14] CHENG C P, WU Z G. A method of sentence similarity computing based on HowNet[J]. Computer Engineering & Science, 2012, 32(2): 172-175. (in Chinese)
程传鹏，吴志刚. 一种基于知网的句子相似度计算方法[J]. 计算机工程与科学，2012，32(2)：172-175.

[15] HE W, WANG Y. Text representation based on sentence and Chinese text categorization[J]. Journal of the China Society for Scientific and Technical Information, 2009, 28(6): 839-843. (in Chinese)
何维，王宇. 基于句子的文本表示和中文文本分类研究[J]. 情报学报，2009，28(6)：839-843.