doi:10.16652/j.issn.1004-373x.2016.13.024

基于词向量的句子相似度计算及其应用研究

郭胜国，邢丹丹
（郑州财经学院 计算机系，河南 郑州 450044）

摘 要：目前计算机辅助翻译在相似度计算方面存在很大的局限性，精度较低。针对特定领域，收集领域相关的训练语料，采用Google的word2vec进行英语和汉语的词向量模型构建，设计并实现汉语句子的相似度计算方法，提出基于词向量Jaccard相似度与基于词向量依存句法相结合的相似度计算方法，实验结果显示其效果比传统方法有较大提升。最后，将相关英汉句子相似度算法以接口的形式封装，作为商品化软件华建IAT系统的相似度计算模块投入实际应用。

关键词：计算机辅助翻译；相似度计算；词向量；IAT系统
中图分类号：TN711-34；TM417　　**文献标识码**：A　　**文章编号**：1004-373X(2016)13-0099-04

Sentence similarity calculation based on word vector and its application research

GUO Shengguo，XING Dandan
（Department of Computer，Zhengzhou Institute of Finance and Economics，Zhengzhou 450044，China）

Abstract：Currently，the computer aided translation has great limitation on similarity calculation，and its accuracy is low. For the specific field，the training corpus about the field was collected，and the word vector model of English and Chinese was constructed by using word2vec developed by Google to design and implement the similarity calculation method of Chinese sentences. The similarity calculation method of combining word vector Jaccard similarity with word vector dependent syntax is proposed. The experiment results show that effect of the proposed method has more improvement than that of the traditional method. The similarity algorithm of relevant English and Chinese sentences is packaged by means of interface form. The similarity calculation module of Huajian IAT system was put into practical application as a commodity software.

Keywords：computer aided translation；similarity calculation；word vector；IAT system

近年来，机器翻译的需求量越来越大。从学术文献翻译到搜索引擎跨语言检索，机器翻译都呈现出巨大的需求空间。但是传统的机器翻译一直都存在较大的问题，就是准确率过低。通过前期调查，现有的机器翻译无法满足实际需求，同时随着“大数据”时代的到来，越来越多的学者将目光转向了计算机辅助翻译[1-2]。

计算机的翻译结果仅仅作为一个辅助性的参考，最终的结果由使用者根据候选结果的好坏进行修改取舍，实例语料库可以结合具体领域进行分门别类的应用，使得其翻译效果进一步提升，从而更加契合用户的需要。由于领域性的差异，尤其是专业术语较多的时候，这一优点在特定领域的翻译中显得更加突出[3]。

1 计算机辅助翻译及句子相似度计算

1.1 计算机辅助翻译技术

计算机辅助翻译系统涉及到的技术构成如图1所示。在翻译系统的整体构成上，相似度计算模块是计算机辅助系统的核心部分[4]。它主要完成输入句子与翻译记忆库的匹配并为用户提供参考功能。而翻译记忆库则是计算机辅助翻译的基础构成，它主要用来对已翻译的句子进行存储和检索。

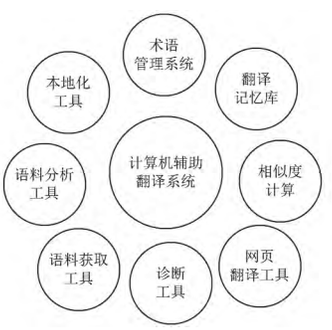

图1 计算机辅助翻译系统技术构成

1.2 句子相似度计算算法

1.2.1 基于共现词的相似度计算

相似度不仅受到共现词汇数量的影响，同时还要结

收稿日期：2015-11-24

合句子包含的词汇总数即句长进行度量，用公式可以更加直观地表达为：

$$\mathrm{SimScore}(I,R)=\frac{\mathrm{Inter}(I,R)}{\mathrm{Union}(I,R)} \tag{1}$$

式中：Inter(I,R)是输入句子I和检索到的句子R之间共现的词汇数目；Union(I,R)是指两个句子单词构成的合集的词汇数目。由此得到的相似度一般将其叫做Jaccard相似度。

改进的Jaccard方法不仅保留了计算句子共现词汇的影响因子，同时为每一个共现词汇加上了权重，一般采用词汇的TF-IDF值[5]。具体公式如下所示：

$$\mathrm{SimScore}(I,R)=\frac{\mathrm{Inter}(I,R)}{\mathrm{Union}(I,R)}\left(\sum_{w\in \mathrm{Inter}(I,R)}\log\frac{N}{df_w}\right) \tag{2}$$

式中：括号中的内容是共现词汇的TF-IDF权重加和；N是翻译记忆库中的实例句对总数；df_w是出现了词汇w的实例句对数目；$\log\frac{N}{df_w}$即为单词w的倒文档频率IDF值，它越大则表明该词汇的重要程度越大，对相似度计算的贡献越大。

1.2.2 基于语义词典的相似度计算

基于WordNet的句子相似度计算方法为：

$$\frac{\sum_{w\in I}\max \mathrm{SimValue}(w,R)+\sum_{w\in R}\max \mathrm{SimValue}(w,I)}{|I|+|R|} \tag{3}$$

式中：$\max \mathrm{SimValue}(w,I)$和$\max \mathrm{SimValue}(w,R)$是词汇$w$和另外一个句子中每个词的语义相似度中最大的值，作为该词汇与另一个句子的相似度。

1.2.3 编辑距离

将原来的布尔值改为词义的相似度来表示句子的编辑距离，将编辑距离除以两句中较长的一句的句子长度，最终得到句子的相似度：

$$\mathrm{SimScore}(I,R)=\frac{\mathrm{dis}(I,R)}{\max(I,R)} \tag{4}$$

2 基于词向量的句子相似度计算算法

2.1 英语句子相似度计算算法设计

在进行改进的华建IAT系统上，英语句子的原相似度计算算法是基于共现词汇相似度即Jaccard相似度和编辑距离相结合的方式设计的[6]。将词向量引入到相似度计算上，本文设计的英语相似度计算算法由两部分构成：基于词向量的Jaccard相似度计算算法；基于词向量和编辑距离的相似度计算算法。

2.1.1 基于词向量的Jaccard相似度计算算法

传统的Jaccard相似度度量算法是基于词汇的重叠度来衡量两句话或短语的相似度[7]。本文提出的基于词向量的Jaccard相似度计算算法则是将词汇的共现替换为词向量的共现程度，也即语义共现程度。公式化表达如下：

$$\mathrm{CosDis}(\boldsymbol{w}_1,\boldsymbol{w}_2)=\frac{\boldsymbol{w}_1\cdot\boldsymbol{w}_2}{\|\boldsymbol{w}_1\|\times\|\boldsymbol{w}_2\|}=\frac{\sum_{i=1}^{n}w_{1i}\times w_{2i}}{\sqrt{\sum_{i=1}^{n}(w_{1i})^2}\times\sqrt{\sum_{i=1}^{n}(w_{2i})^2}} \tag{5}$$

$$\mathrm{SimScore}_{\mathrm{Jaccard}}(I,R)=\frac{\sum_{w\in I}\min(\max(\alpha\times\mathrm{CosDis}(w,R)),1)}{|I|+|R|}+\frac{\sum_{w\in R}\min(\max(\alpha\times\mathrm{CosDis}(w,I)),1)}{|I|+|R|} \tag{6}$$

式(5)中$\mathrm{CosDis}(\boldsymbol{w}_1,\boldsymbol{w}_2)$用来计算两个词对应的词向量$\boldsymbol{w}_1$和$\boldsymbol{w}_2$的余弦相似度，$n$即为词向量的维度。式(6)中，$I$,$R$分别是用户当前输入的句子和系统从翻译记忆库中检索到的句子。$\max(\alpha\times\mathrm{CosDis}(w,R))$是计算句子$R$中所有词对应的词向量与$w$对应的词向量的余弦相似度中的最大值。参数$\alpha$用来调整两个词向量之间余弦相似度的放大系数，因为根据不同语料训练出来的词向量模型是不同的，其计算结果也就会有一定的浮动，可以根据得到的词向量的不同进行设定调整。为了防止放大系数的放大效果超出实际表示范围[-1，1]，所以对其最大值进行了门限设置。

基于词向量的Jaccard相似度计算和基于语义词典的Jaccard相似度计算有异曲同工之处，但前者表示的语义范围要远超过语义词典的表示范围，比如“worker”和“work”之间在WordNet中是找不到其语义关系的，但是在词向量中，其余弦相似度达到0.452之多，而这在语义词典中是很少有做到的。其他诸如语义词典的不可扩展性等弊病在词向量中是不存在的，这也是词向量的优势所在。

2.1.2 基于词向量和编辑距离的相似度算法

将词向量应用到编辑距离中，进而得到本文所使用的基于词向量和编辑距离的相似度计算算法。基于词向量的编辑距离，计算句子的相似度为：

$$\mathrm{SimScore}_{\mathrm{edit\_dis}}(I,R)=1-\frac{\mathrm{edit\_distance}}{\max(|I|,|R|)} \tag{7}$$

2.1.3 综合相似度计算

对于英语相似度计算主要由以下两部分构成[8-9]：基于词向量的Jaccard相似度主要考量了相关词汇的共现程度，其中既有表层的相似度又有词义的相似度；而基于词向量的编辑距离不仅考虑了词汇本身的语义、语境

相关性，还考量了句子结构的相似性[10]。所以，尽管算法构成上比较简单，但是包含的相似度计算因素是多样的，综合两种算法的优势，将其加权求和构成英语句子最终的相似度计算算法：

$$\mathrm{SimScore}(I,R)=\rho_1\mathrm{SimScore}_{\mathrm{Jaccard}}(I,R)+\rho_2\mathrm{SimScore}_{\mathrm{edit\_dis}}(I,R) \quad (8)$$

式中：$\rho_1+\rho_2=1$，并且满足$0<\rho_2\leqslant\rho_1<1$。

2.2　汉语句子相似度计算算法设计

（1）依存句法

首先对输入句子 I 和 R 进行句法分析，得到句法分析结果Ipar和Ppar。将两个句法分析结果中的依存对提取出来，保存为三元组的依存对deppair，其中deppair[0]和deppair[2]为依存词，deppair[1]为依存关系。在两个句子依存对集合中，依存关系相同的依存对进行余弦相似度计算，deppair$I$[0]和deppair$R$[0]表示词汇对应的词向量。SimilarityThreshold是设定的词汇之间语义相似度的阈值，可以根据模型的精度进行调整。当依存对中的两个词语义相似度都达到阈值要求时，按照下式进行句子的相似度计算。

$$\mathrm{SimScore}_{\mathrm{dep}}(I,R)=\frac{\sum_{\mathrm{deppair}I\in\mathrm{DepSet}I}\min(\alpha\times\max(\mathrm{CosDis}(\mathrm{deppair}I,\mathrm{DepSet}R)),1)}{2\times\max(|\mathrm{DepSet}I|,|\mathrm{DepSet}R|)}$$

将上述依存对中的词分别求余弦相似度，选取相似度最大的一组，对该组相似度求取算术平均值并用放大系数放大作为依存对的相似度。

（2）汉语句子的综合相似度计算

同英语句子的相似度计算一致，汉语句子的综合相似度计算采用两部分加权求和的方式进行。除了基于词向量的依存句法，相似度还有基于词向量的Jaccard相似度，二者共同构成本文提出的汉语句子相似度计算算法。公式如下：

$$\mathrm{SimScore}(I,R)=\mu_1\mathrm{SimScore}_{\mathrm{Jaccard}}(I,R)+\mu_2\mathrm{SimScore}_{\mathrm{dep}}(I,R)$$

式中：$\mu_1+\mu_2=1$，$0<\mu_2\leqslant\mu_1<1$。

3　算法实现和实验分析

3.1　词向量模型训练

使用的word2vec需要相应的语料进行词向量模型的训练。首先是word2vec的训练命令，对于得到的词向量模型，需要根据词向量的精度进行词汇之间相似度的阈值判定。这里根据常用同、近义词的余弦相似度测试结果进行设定。最终得到的阈值分别为：

英语：相似度应当分成几档对待，[0，0.2）为不相似，[0.2，0.3）为相似，[0.3，0.4）为较强相似，[0.4，1]基本相同；

汉语：[0，0.25）为不相似，[0.25，0.4）为相似，[0.4，0.5）为非常相似，[0.5，1]基本等同。

之所以汉语的各档相似度阈值设定得比英语高一点，是因为汉语的训练语料更多，相应地，词汇之间的相似度比英文更加贴近实际情况。另一方面又可以看出英语和汉语的最低相似度阈值都很低，因为词向量的训练过程是结合具体语料进行的，语料的质量和方法本身的统计思想决定了方法相比人工判定是比较粗糙的。

3.2　系统组成结构

系统的组成结构如图2所示。翻译开始后，首先选定语言种类，因为要根据语言选择相应的词向量模型，也便于系统更有针对性地进行实例检索。选定语言后，针对语言种类分别进行词向量模型的读取和加载。对于英语，下一步要对输入句子和检索出的Top$N$的实例句子进行预处理，包括删除标点、大写还原等。特别要说明的是，由于华建IAT系统的商用保密性原因，实验不能直接在该系统的实例库上进行。故本实验另外单独搭建系统进行效果测试，针对结果进行改进并最后将算法运用到华建IAT系统上。在实验系统中，使用基于词共现的Jaccard相似度（没有使用词向量）进行检索，返回Top50的实例。汉语的检索过程与此相同，只是在输入句子和检索的Top50实例的预处理上多了分词。为了使实验结果和实际使用效果尽量保持一致，分词使用了华建内部提供的分词工具。

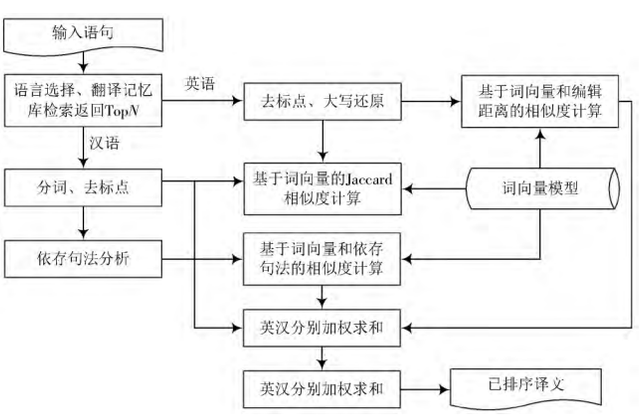

图2　系统结构图

3.3　相似度计算实验

3.3.1　实验数据

为了尽量让实验结果客观、可信，本文英语的相似度计算实验数据来自LDC2的收费语料LDC2013T03。该语料来自NIST 2012年的机器翻译评测。本文将这些同义句全部按组抽取排列。为了在长短句中有所均衡，随机从中抽取了100组，得到最后的测试语料，形式

如下所示：

1. White dragon fruit has a reddish purple peel and white flesh; it has tiny black seeds distributed through the middle and is of so-so quality when eaten fresh.

2. White pitayas have purple-red skin and white flesh, with tiny black seeds distributed throughout and the quality is okay if eaten uncooked.

3. White dragon fruit with purple reddish skin and white meat, with tiny black seeds distributed, average quality for eating fresh.

4. A white pitaya is mauve-skinned with white flesh containing fine black seeds, and of average quality when eaten fresh.

1. At least they don't go out of their way to fleece Chinese people.

2. At least they don't specifically rip off Chinese people.

3. At least they don't exclusively rip off Chinese people.

4. At least they do not rip off Chinese only.

选取每组中的第一句作为基准句，其他三句作为第一句的相似句，用于测试。所有这400句先与8 336句同样是NIST 2012的评测句子进行混合作为一组英文测试语料，然后又与9 600句华建语料库中的句子混合，共计10 000句，作为第二组英文测试语料。

汉语的测试语料由于没找到和英语类似的同义或近义句，所以采取了常用的相似度研究方法。从华建公司的语料库中随机选取100个句子，并自行构建100句对应的相似句子，即100组相似句对。将这200句相似句和另外来自华建语料库的其他9 800句混合，共计10 000句作为汉语的测试语料。

3.3.2 测试标准

针对英语和汉语的测试语料情况，给出了相应的测试标准。对于英语的每一组相似句存在三句相似句，所以采用了两种测试方法分别进行。

第一种是测试Top5中的Top3包含该组三句相似句中的数量所占百分比，即：

$$\text{Accuracy} = \text{每组Top3中命中个数累加}/300 \times 100\%$$

第二种是测试Top5中的Top1在该组三句相似句中的数量所占百分比，即：

$$\text{Accuracy} = \text{每组Top1中命中个数累加}/100 \times 100\%$$

汉语的测试标准同英语第二种测试标准相同。

3.3.3 实验结果及数据分析

在相似度算法中的参数包括六个：SimilarityThreshold，α，ρ_1，ρ_2，μ_1，μ_2。SimilarityThreshold表示相似度阈值，通过对英汉的词向量模型进行分析比较，英语的相似度阈值设定为0.2，汉语的为0.25。α为词向量之间余弦相似度的放大系数，主要是帮助进行误差修正。通过对词向量的相似度精度进行测试，英语设定为2.5，汉语为2。其他两组参数为英汉句子相似度计算中Jaccard相似度和另一部分各占的比重。

通过分析实验可以发现，词向量对算法效果的提升非常明显。基于词的相似度在不同语料库之间的表现差异（命中率之差）要比基于词向量的方法大一些，这表明基于词向量的方法不仅在性能上更好，同时其在不同领域语料之间的稳定性和适应性也更强。

3.4 方法实现及应用

本文提出的方法是对华建IAT系统进行针对性的设计和实现。为了便于后续的维护，最终以接口的形式将相关方法整合给辅助翻译系统进行使用。考虑到系统在实际运行中对时效性的要求，采用IronPython完成.NET平台上接口的实现，以保留Python对文本处理的优势。主要函数构成如下：

W2V_Main（InputSentence，RetrivalSentencePairs，LanguageType，ReturnNum）；

EnSimilarityCompute（InputSentence，CompareSentence）；

ChSimilarityCompute（InputSentence，CompareSentence）。

其中，W2V_Main()是.NET平台进行调用的入口函数，参数含义依次为：InputSentence表示用户输入句子；RetrivalSentencePairs为翻译记忆库检索出的Top*N*个待排序实例句对；LanguageType为输入句子的语言种类；ReturnNum表示需要返回的已排序句子的数目。通过调用英汉对应的相似度计算模块，其返回结果为ReturnNum个已排序译文构成的数组。

EnSimilarityCompute()是英文句子相似度计算模块的入口，其作用是返回用户输入句子InputSentence和系统检索到的句子CompareSentence（句对中的源语言句子）之间的相似度。ChSimilarityCompute()则是汉语句子相似度计算模块的入口函数。这两个句子相似度计算函数会调用英汉对应的预处理函数、Jaccard相似度函数（公用）、编辑距离相似度函数（英语）、依存句法相似度函数（汉语，通过os.system()进行调用并接收返回的分析结果），最后返回两部分加权求和得到的相似度计算结果。

4 结 论

本文采用基于词向量的相似度计算算法进行英汉双语的句子相似度计算，并根据两种语言在使用习惯、句式变化和句子长度等方面的不同进行有针对性的算法设计。词向量的语境相似度在很大程度上增加了句子相似度计算的深度和广度，并且其优秀的语义相关性也可以将同、近义词的使用纳入到相似度考虑的范畴。

（下转第107页）

的聚类。因此改进的算法性能较好。

4 结　论

微分进化算法利用自然界优胜劣汰的思想和简单的差分操作使得其在一定程度上具有自组织、自适应、自学习的特征。但微分进化算法跟其他进化算法一样存在早熟收敛等问题，因此对微分进化算法进行改进，提高其全局搜索能力和收敛速度，使之能够适应各种复杂的工程问题，这是一个值得深入研究的领域。

本文通过研究国内外学者对微分进化算法的改进方法，提出了一种基于主成分的微分进化算法，该算法将种群空间映射到由主成分构成的种群空间，主成分空间中前 m 个主成分构成的个体可以直接进入下一代的进化，而其余的个体则从主成分空间和原始种群空间中挑选出适应度值高的进入下一代，这样的改进方法使得种群能够更多地获得父代的信息，增加了种群的多样性。通过数值实验证明该算法在收敛精度和收敛速度上都有很大的提高。该算法对聚类中心进行编码构成进化种群，利用PCADE算法更新种群，最终得到较好的聚类中心，然后根据聚类中心对样本进行聚类，利用IRIS数据集进行试验，结果表明基于PCADE的K-均值聚类算法在聚类问题中得到了较好的结果。

从上述两个方面来看，该改进的微分进化算法PCADE在性能方面有了很大的提升，值得进一步推广。

参　考　文　献

[1] 李士勇，陈永强，李研.蚁群算法及其应用[M].哈尔滨：哈尔滨工业大学出版社，2004.

[2] 胡中波，熊盛武，胡付高.改进的差分演化算法及其在函数优化中的应用[J].武汉理工大学学报，2007，29(4)：125-128.

[3] 苏海军，杨煜普，王宇嘉.微分进化算法的研究综述[J].系统工程与电子技术，2008，30(9)：1793-1797.

[4] 欧陈委.K-均值聚类算法的研究与改进[D].长沙：长沙理工大学，2011.

[5] 孟凡军，李天伟，徐冠雷，等.基于K-均值聚类算法的雾天识别方法研究[J].现代电子技术，2015，38(22)：56-58.

[6] VOSS M S. Principal component particle swarm optimization [C]// Proceedings of 2005 IEEE Swarm Intelligence Symposium. [S.l.]: IEEE, 2005: 401-404.

[7] EBERHART R, SHI Y. Comparing inertia weights and constriction factors in particle swarm optimization [C]// Proceedings of 2000 IEEE Congress on Evolutional Computation. La Jolla: IEEE, 2000: 84-88.

[8] TASGETIREN M F, SUGANTHAN P N. A multi-populated differential evolution algorithm for solving constrained optimization problem [C]// Proceedings of 2006 IEEE Congress on Evolutionary Computation. Vancouver: IEEE, 2006: 33-40.

作者简介：梁聪刚(1981—)，男，河南南阳人，硕士，讲师。研究方向为偏微分方程。

王鸿章(1979—)，男，河南汝州人，硕士，讲师。研究方向为偏微分方程。

(上接第102页)

同时，汉语句子的依存句法很好地规避了汉语句式多变和句子长度对相似度的影响，在句子的整句范围内进行更加深入的相似性度量。

参　考　文　献

[1] 侯宏旭，刘群.基于实例的汉蒙机器翻译[J].中文信息学报，2011，21(4)：65-72.

[2] LI Y N, LI H S, CAI Q, et al. A novel semantic similarity measure within sentences [C]// Proceedings of 2012 2nd International Conference on Computer Science and Network Technology. Changchun, China: IEEE, 2012: 1176-1179.

[3] 李丹，许霄羽，杨悦.基于语义网技术的网络机器翻译研究[J].现代电子技术，2011，34(4)：107-109.

[4] 王荣波，池哲儒.基于词类串的汉语句子结构相似度计算方法[J].中文信息学报，2005，19(1)：21-29.

[5] 黄河燕，陈肇雄.基于多策略的交互式智能辅助翻译平台总体设计[J].计算机研究与发展，2012，41(7)：1266-1272.

[6] 李素建.基于语义计算的语句相关度研究[J].计算机工程与应用，2012，38(7)：75-76.

[7] 张民，李生，赵铁军，等.一种汉语句子间相似度的度量算法和实现[J].计算机语言学进展与应用，1995(7)：152-158.

[8] 梁晗，陈群秀，吴平博.基于事件框架的信息抽取系统[J].中文信息学报，2006，20(2)：40-46.

[9] 秦兵，刘挺，王洋，等.基于常问问题集的中文问答系统研究[J].哈尔滨工业大学学报，2013，35(10)：1179-1182.

[10] 梁红玉，张平.LEX在ATLAS-C翻译器设计中的应用[J].现代电子技术，2004，27(24)：102-104.

作者简介：郭胜国(1982—)，男，硕士研究生，讲师。研究方向为智能控制、计算机应用。

邢丹丹(1982—)，女，硕士研究生，讲师。研究方向为计算机应用。