基于 word2vec 与 LVQ 的作业查重及评分系统

张家麟[1]，孔繁宸[1]

(西南大学 计算机学院，重庆市 400715)[1]

摘　要：为了减少教师在批改作业上花费的大量时间，设计出了一个基于word2vec与LVQ神经网络的学生作业查重及评分系统。先运用LCS算法对文章进行查重，再让老师对20%的学生作业进行评分，从每份已评分作业中使用TF-IDF算法提取关键字，通过用word2vec获取关键字的词向量，再把每篇的多个词向量和评分传入LVQ神经网络进行训练得出模型，然后对其余80%的文章采用同样的方法提取关键字，把这些词向量传入LVQ神经网络得到分数。并在我院学生作业上进行了实验，在简答题、自我评价、总结分析类文本上得到了相当不错的实验结果，但在作文这样有句子结构的文本上效果一般。

关键词：LVQ神经网络；word2vec；查重评分系统；自然语言处理；

中图分类号：TP183　　**文献标志码：**A　　DOI：

Design and Implementation of the Checking and Grading System for Students' Homework

ZHANG Jia-lin[1] KONG Fan-chen[1]

(School of Computer and Information Science，Southwest University，Chongqing 400715，China)[1]

Abstract: In order to reduce the amount of time spent on correcting homework, we design a Checking and Grading System based on word2vec and LVQ neural network. The LCS algorithm is used to check students' homework at first, after that, we let the teacher to grade 20% of the students' homework. By using the TF-IDF algorithm to extract keywords from the scored homework, we apply the word2vec to get the key words' vectors, then the words' vectors and their corresponding scores are used to train the LVQ neural network. As for the 80% homework remained, we apply the trained LVQ network to evaluate their scores by using the same step. Experiments were carried out on the students' homework in our college, in the text of short answer, self-evaluation, summarize and analyze text, the results are very good, however, in the text which has sentence structure, such as composition, this model doesn't work well.

Key words: Learning Vector Quantization Neural Network; Word2vec; Checking and Grading System; Natural Language Processing

到稿日期：　　返修日期：

张家麟，男，本科，CCF 会员，主要研究方向为数据挖掘，自然语言处理，机器学习

E-mail:zjl970770194@email.swu.edu.cn。

1. 引言

学生作业是教学过程中必不可少的一部分，但是批改大量的作业会降低教学效率，也会给老师造成麻烦，浪费老师的时间。所以部分老师会选择用抽查的形式来批改作业，每次抽取一部分学生作业来批改，下次再抽查另一部分学生作业。或者是老师自己批改一部分，剩下作业就选择一个优秀的学生来批改。

随着人工神经网络的发展，计算机的日渐强大，同时学生作业也逐渐电子化，用机器学习的方法来取代传统人工批改作业的方法，不免是个不错的选择。在 2013 年 Tomas 等人[1,2]提出了 word2vec 词向量的思路，它可以很好的表达词的语义信息，使词语得到了量化，我们就能算出不同词语之间的相似度，例如郑文超等人[3]就利用 word2vec 对中文词进行了聚类的研究。一次作业或者一个问题的答案，不妨看作一个文本，但是我们不能直接用算余弦距离的方法[4]来对这些文本进行分类，因为每个学生对同一道题给出的答案都是大同小异的，最后分类结果都会是同一类。又由于每个学生答案中的介词代词并不是答题的关键，所以我们用 TF-IDF 算法[5]找出答题的关键词，并用 word2vec 获取这些关键词的词向量。将每篇文章关键词的词向量按照词频排序后连接在一起所得到的向量，就可以当作每篇文章的度量。于是，每篇文章的向量就可以当作 LVQ 神经网络的输入，使 LVQ 神经网络得到训练并且进行分级评分。LVQ 神经网络早已被用于各种模式识别领域，例如胡红等人[6]利用 LVQ 神经网络进行岩性识别，李敏等人[7]利用结合概率型神经网络(PNN)和学习矢量量化(LVQ)算法来进行文本分类。但是对同类文本进行分级评分的思路还是很少，本文的思路就是运用 LVQ 进行同类文本的分级评分。

本文剩余部分结构如下，第二部分进行相关算法介绍，包括 LCS 算法、TF-IDF 算法、word2vec 算法、LVQ 神经网络以及 LVQ 神经网络的改进。第三部分包括基于我院 2016 级学生作业的实验以及实验检验。第四部分是全文进行总结并提出对未来工作的展望。

2. 相关工作

2.1. 整体算法流程图

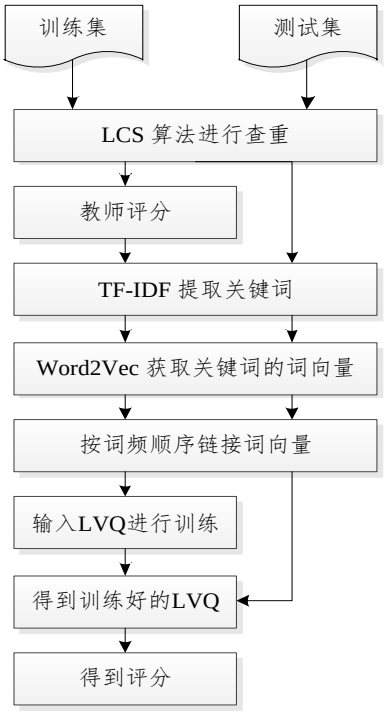

图 1 整体算法流程图

2.2. LCS 算法介绍

LCS（Longest Common Subsequence）算法早在 2007 年就被王映龙等人用于基因序列相似度的对比[8]，它是一种用于对比两串序列相似度的算法，如果把文本字符串看作序列，那么就可以用 LCS 算法来对比文本相似度了。不用 LDA 模型或是 SVM 模型来对比相似度，是因为对于同一个问题，每个学生给出的答案是大同小异的，而 LDA 或 SVM 用是用于文本对比分类的，LCS 正好可以用于文本相似度对比与查重。

2.2.1. 算法定义

把两个文本分别看成两个长字符串 S_1 和 S_2，遍历 S_1 的每个子字符串，检查它是不是 S_2 的子字符串，从而就能找到最长公共子字符串。设 S1 和 S2 长度分别为 L_1 和 L_2，最长公共子字符串长度为 L_{common}，则相似度计算公式如下：

$$\text{Similarity} = \frac{2 * L_{common}}{L_1 + L_2} \tag{1}$$

2.3. TF-IDF 算法介绍

关键词提取是文本评分的第一步，也是最重要的一步，关键词的提取的好坏直接影响着评分的效果。本文改进了传统的 TF-IDF 算法来达到预期效

果，TF-IDF 的改进也是经历了多年的研究，如赵小华[9]等就应用 TF-IDF-CHI 来修正每个特征词的权重，重新调整了每个特征词对类区分的贡献程度。

2.3.1. 传统 TF-IDF 算法

如果某个词或短语在一篇文章中出现的频率（TF）高，并且在其他文章中很少出现（IDF），则认为此词或者短语具有很好的类别区分能力，可用作关键词。TF-IDF 实际上是：TF*IDF，TF 词频(Term Frequency)，IDF 逆向文件频率(Inverse Document Frequency)。它们的计算公式如下：

$$TF_n = \frac{L_n}{SUM} \tag{2}$$

$$IDF_n = \log \frac{|D|}{1+|\{j : n \in d_j\}|} \tag{3}$$

SUM 是指文本中所有字词的出现次数之和，L_n 是指 n 这个关键词在文本中出现的次数，|D|是指语料库中的文件总数，$1+|\{j : n \in d_j\}|$ 是指包含该词语文件的数目+1，TF-IDF 值越高，则这个词越重要，由此可以选出关键词。

2.3.2. 改进的 TF-IDF 算法

由于本文是要对同类文章进行评分，所以本文对传统 TF-IDF 算法进行了相应地改进，首先让评分者给出此类文本的类别词，然后在传统 TF-IDF 算法的基础上求出每个词与类别词的余弦相似度 COS[10]，再用 COS*TF*IDF 得到改进 TF-IDF 值。它们的计算公式如下：

$$\cos\theta = \frac{\sum_1^n (A_i * B_i)}{\sqrt{\sum_1^n (A_i^2)} * \sqrt{\sum_1^n (B_i^2)}} \tag{4}$$

$$\text{TF-IDF} = TF * IDF * \cos\theta \tag{5}$$

由 2.4 的 Word2Vec 算法可得词向量 A = $(A_1,A_2,...,A_n)$，词向量 B = $(B_1,B_2,...,B_n)$，则可算出它们两词之间的相似度，由此能提取出与此次评分更相关的词语。

2.4. Word2Vec 算法介绍

基于 Bengio 提出的 NNLM(Neural Network Language Model)[11]，Hinton 的 Log_Linear 模型[12]，以及 Mikolov 等人所提出的 word2vec 语言模型[13]，word2vec 可以快速有效的训练词向量。

word2vec 模型有两种，分别是 CBOW 模型(见图 2)以及 Skip-gram 模型(见图 3)。其中 CBOW 模型利用 w(t)前后各 c(图中 c=2)个词去预测当前词；而 Skip-gram 模型恰好相反，它利用词 w(t)去预测它前后各 c(图中 c=2)个词。

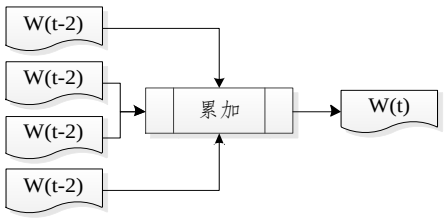

图 2 CBOW 模型

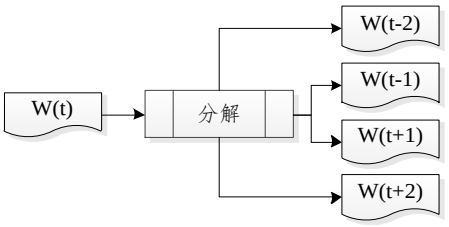

图 3 Skip-gram 模型

CBOW 模型和 Skip-gram 模型的训练方式基本相似，本文选用后者在下文中详细讲解。

2.4.1. 分词

Word2vec 的第一步当然是对文本进行划分，中文和英文的分词各有各的难点，中文的难点在于将句子分解成一个单词数组。而英文虽然不需要分词，但是要处理各种各样的时态，所以要进行词干提取和词形还原。

对于中文文本，选用的结巴分词来处理，虽然结巴分词会有误差，但是基本能满足需求。对于英文文本，就直接划分后，再根据语料库来进行词形还原。

2.4.2. 构造辞典和 Huffman 树

这一步需要遍历一遍所有文本，找出所有出现过的词，并统计各词的出现概率。再根据词的出现概率来构建 Huffman 树，同时对每个节点进行二进制编码再初始化词向量。需要注意的是所有词都只出现在 Huffman 树的叶节点。

2.4.3. Skip-gram 模型训练

在 Huffman 树构建好之后，将当前词的词向量作为输入层，将其周围词的词向量作为输出层。借助之前构造的 Huffman 树，在每次训练中，映射层的值需要沿着 Huffman 树不断的进行 logistic 分类，并且不断的修正各中间向量和词向量。

当整个神经网络训练完后，即得到所有词的词向量，有趣的是，当这些词得到量化后，可以发现类似这样的规律：“king”-“man”+“woman”=“queen”，可以看出词向量非常有利于表达词的语义特征。

2.5. LVQ 神经网络

人工神经网络能模拟生物神经系统对真实事物作出交互反应，在生物神经系统中，竞争的思维尤为特别，人类辨别事物的好坏，是根据事物更趋近好的一端，或是更趋近坏的一端来判别的，正所谓“近朱者赤近墨者黑”。

在竞争网络结构的基础上，学习向量化（learning vector quantization，LVQ）网络被提出来，它融合竞争学习思想和有监督学习算法的特点，通过教师信号对输入样本的分配类别进行规定，从而克服自组织网络采用无监督学习算法带来的缺乏分类信息的弱点。

2.5.1. 传统 LVQ 神经网络

传统 LVQ 神经网络早已经广泛运用和改进，例如由胡帅等人[14]的基于 PCA-LVQ 神经网络的教学质量评价模型研究。它由输入层、竞争层、输出层组成，竞争层通过计算输入数据之间的欧式距离来完成分类，并将分类结果输出到输出层，每个输出层神经元代表一个分类结果。

LVQ 神经网络的具体流程如下：

1) 初始化输入层的第 j 个神经元与竞争层的第 i 个神经元之间的权值w_{ij}及学习速率η。

2) 将输入样本向量$X=(X_1,X_2,\ldots,X_n)$输入到输入层，计算竞争层神经元与输入向量之间的距离 d_i，具体计算方法如式(6)所示。

$$d_i=\sqrt{\sum_{j=1}^{n}(X_j-w_{ij})^2} \tag{6}$$

3) 选取与最小 d_i 相对应的竞争层神经元，将其作为获胜神经元权值进行调整。如果分类正确，则按照式(7)调整权值，否则按照式(8)进行权值更新。

$$w'_{ij}=w_{ij}+\eta(X_j-w_{ij}); \tag{7}$$

$$w'_{ij}=w_{ij}-\eta(X_j-w_{ij}). \tag{8}$$

2.5.2. 改进的 LVQ 神经网络

对于传统 LVQ 神经网络，竞争的量化标准是欧氏距离，从数学角度上讲，欧式距离就是实际两点的直线距离，如下 Voronoi 图所示，就是传统 LVQ 神经网络的一个小的聚类方式。

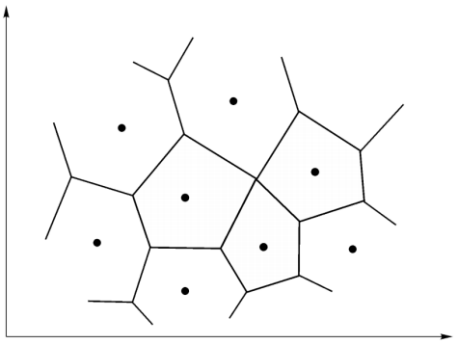

图 4 欧式距离聚类

但实际上欧式距离并不适用于词向量的相似度对比[15],而应该采用余弦距离来进行聚类，在二维空间的聚类方式可见图 5。

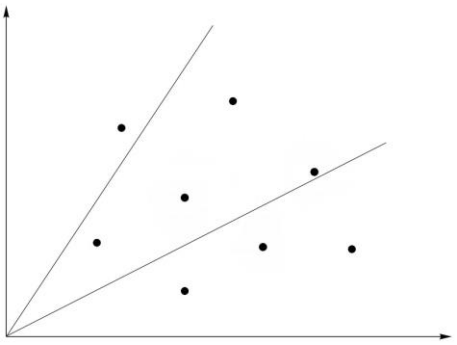

图 5 余弦距离改进的聚类

其余的计算过程与传统 LVQ 神经网络相同，只是距离比较这一点做了相应的改进。

3. 实验与检验

3.1. 数据来源

本文基于 Python 2.7 和 MATLAB 2014a 平台编程建立了学生作业查重及评分系统。实验数据来源于本院 2016 级本科生的课堂作业，作业格式均为实验过程加心得体会的 word 文档，文档名仅为学号。共有 6 次作业，每次作业应有 54 份，存在不交作业的现象。对于每份作业需要有个 100 分制的评分，前 3 次作业，老师已经对全部作业进行了评分。

3.2. 实验过程

对于每次作业的 word 文档，首先使用 Python 的 win32com 模块，提取出 word 文档中的文字，结合 Python 自带语料库，将这些所有文字进行 word2vec 训练，其中 word2vec 的计算采用 gensim 开源软件实现[16]。

对于每份作业，使用本文改进的 TF-IDF 进行关键词提取，例如以下这样的文本：

“通过本次实验，我明白了，字符串的基本使

用方法，知道如何操作字符串相加，逐一对比时，若字符串长度不等，要先分开长度不等的情况，再对比。我还学会了合理使用搜索引擎来自我学习。”

提取出来的关键词与 TF-IDF 值如表 1：

表 1 关键词与其 TF-IDF 值

关键词	TF-IDF 值
字符串	0.9574
长度	0.5323
对比	0.5108
搜索引擎	0.3602
相加	0.3375

然后获取每个关键词的词向量，每个词向量有 20 维，链接词向量得到文章的量化标准，见表 2。

表 2 链接后的词向量表

学号	X_1	X_2	…	X_{99}	X_{100}
001	0.0059	0.0234	…	0.0214	0.0182
002	0.0194	0.0117	…	-0.010	0.0102
⋮	⋮	⋮	⋮	⋮	⋮
053	0.0101	0.0244	…	0.0242	0.0176
054	0.0044	-0.112	…	0.0039	0.0054

再将这些词向量传入，改进的 LVQ 神经网络就可以进行训练以及评分了。

在样本容量如此少的情况下，LVQ 神经网络不能进行 100 个等级的评分，所以先进行 5 个等级的评分，再对每个等级进行相应的加分，本文使用的是根据字数多少进行加分，因为字数多少反应了学生的认真程度，而且平时老师评分的时候也会根据字数多少来评判一个学生的态度。

3.3. 检验分析

对于前 3 次老师已经全部评分过的作业，采用五分交叉验证，即把已评分的作业随机划分成 5 份，每次用第 1 份进行训练 LVQ 神经网络， 其余 4 份进行测试检验，5 次检验后平均误差为 8.72%。

对于后三次作业，让老师对其中 20%的作业进行评分，再用其进行 LVQ 神经网络的训练，最后用训练好的 LVQ 神经网络对余下 80%的作业进行评分，将所有评分成绩给学生们公布，让学生反应满意度，最终平均每次作业有 5 个学生提出了质疑。

结束语 本文提出的学生作业评分系统，不仅能用于学生作业的评分，还能广泛地用于同类文章的分级评分，具有鲁棒性与通用性。但分级不易过多，不然评分误差会增大，不易直接给出 100 分制的分数。而且不能对作文这样有句子结构的文章进行评分，将来会对此进行进一步研究。

参考文献

[1] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space [J]. Computer Science, 2013.

[2] 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, 29(26):3111-3119.

[3] 郑文超, 徐鹏. 利用 word2vec 对中文词进行聚类的研究[J]. 软件, 2013, 34(12):160-162.

[4] 彭凯, 汪伟, 杨煜普. 基于余弦距离度量学习的伪 K 近邻文本分类算法[J]. 计算机工程与设计, 2013, 34(6):2200-2203.

[5] 张瑾. 基于改进 TF-IDF 算法的情报关键词提取方法[J]. 情报杂志, 2014, 33(4):153-155.

[6] 胡红, 曾恒英, 梁海波,等. 基于主成分分析和学习矢量化的神经网络岩性识别方法[J]. 测井技术, 2015, 39(5):586-590.

[7] 李敏, 余正涛. 结合概率型神经网络(PNN)和学习矢量量化(LVQ)算法的文本分类方法[J]. 计算机系统应用, 2012, 21(10):81-85.

[8] 王映龙, 杨炳儒, 宋泽锋等. 基因序列相似程度的 LCS 算法研究[J]. 计算机工程与应用, 2007, 43(31):45-47.

[9] 赵小华，马建芬.文本分类算法中词语权重计算方法的改进[J].电脑知识与技术，2009，5(36):10626-10628.

[10] 药珍妮. 基于主题和特征的文本相似度算法研究[J]. 软件, 2016, 37(10):122-126.

[11] Bengio Y, Schwenk H, Senécal J, et al. Neural Probabilistic Language Models [J]. Journal of Machine Learning Research, 2001, 3(6):1137-1155.

[12] Mnih A, Hinton G. Three new graphical models for statistical language modelling[C]// Machine Learning, Proceedings of the Twenty-Fourth International Conference. DBLP, 2007:641-648.

[13] Le Q V, Mikolov T. Distributed Representations of Sentences and Documents [J]. Computer Science, 2014, 4(34):1188-1196.

[14] 胡帅, 顾艳, 曲巍巍. 基于 PCA-LVQ 神经网

络的教学质量评价模型研究[J]. 河南科学, 2015,7(33):1247-1252.
[15] 余弦距离、欧氏距离和杰卡德相似性度量的对比分析. http://www.cnblogs.com/chaosimple/archive/2013/06/28/3160839.html
[16] Gensim. Topic Modelling for Humans [OL]. http://radimrehurek.com/gensim