文章编号：2096-1618(2024)02-0163-07

基于关键词聚类的新闻文本相似度计算

祝 婷， 胡建成
(成都信息工程大学应用数学学院，四川 成都 610225)

摘要：针对新闻文本篇幅长、冗余信息多、文本相似度难以准确高效计算的问题，提出一种基于关键词聚类的新闻文本相似度计算方法。首先对文本数据进行预处理，挖掘出文本中的关键信息。使用以 TF-IDF 值为权重的加权采样方法抽取文本数据集中的关键词，基于聚类的方法光滑噪声数据。聚类形成簇后，在簇间词语相似度计算上，使用 word2vec 融合 TF-IDF 词语加权的计算方法，同时关注词语间的语义信息和词语频率。最后，基于各簇的相似度计算两篇文本的相似度。实验表明，所提新闻文本相似度计算方法在计算效果上优于传统计算方法。

关 键 词：新闻文本相似度；word2vec；TF-IDF；关键词聚类
中图分类号：TP391.1 **文献标志码**：A
doi：10.16836/j.cnki.jcuit.2024.02.006

0 引言

随着互联网技术的高速发展，各类网络平台充斥着指数级增长的文本数据。如何从海量文本中快速、有效地挖掘出所需信息，是当前文本研究领域亟待解决的问题。网络新闻作为信息时代的基础应用，是发布和获取信息的一个重要途径，在信息传播中发挥巨大的作用。

然而，现有网页中存在大量的重复性新闻，同一新闻的重复报道造成了巨大的网络资源浪费。如果在网页上传新闻时关注到新闻之间相似度的问题，通过对新闻相似度的监测，降低相似新闻的报道比率，更易于在网络上抓取需要的新闻信息，在某种程度上能提高各类新闻的检索、推荐的准确性，给用户更好的使用体验。

针对上述问题，本文以新闻文本为研究对象，基于新闻文本下的数据特征，提出一种基于文本关键词聚类后的相似度计算模型。

1 相关工作与理论

1.1 相关研究

文本相似度计算是自然语言处理领域中的一项基本研究，一系列的 NLP 任务，如问答[1-2]、信息检索[3-4]、抄袭监测[5-6]、情感分析、自然语言推理等，都建立在文本相似度计算基础上，识别相似的意义表达及其相似性得分一直是 NLP 领域的研究重点。文本相似性度量的一般方法是将文本表示为向量，通过计算向量间的距离得到文本间的相似度。文本的相似度计算方法一般可以分为基于文本表面信息的相似度计算和基于文本语义信息的相似度计算。

基于文本表面信息的相似度计算，是指直接提取文本的字面信息，作用于字符串序列或字符组合[7]。其中，基于字符的方法包括编辑距离（levenshtein distance，LD）[8]、最长公共子序列（longest common sequence，LCS）[9]、汉明距离（hamming distance）[10]、N 元模型（N-Gram）等。根据计算时的表示方式，基于术语的方法可以分为基于空间向量模型（vector space model，VSM）[11]、基于表面术语（surface term/word）和基于散列（hash）算法等方式。

考虑到不同字词在不同语句中表达不同含义，单纯基于表面文本无法准确计算文本间的相似度，提出了结合词语语义信息进行相似性度量的方法。其主要包括基于统计和基于规则的计算方法。基于统计的基本思想认为，如果词语的上下词相似，则词语应具有相似的语义。基于语料将文本表示成计算机可操作的向量形式，是利用统计方法计算文本相似度的主要思路。基于神经网络模型表示向量主要有两种方式，一种是直接将句子表示为句向量，另外一种是从词的角度出发，组合句子中的词向量得到句向量。2013 年 Mikolov 等[12]提出的词嵌入模型，在克服“维度灾难”的同时，还给词向量赋予了词语的语义信息。其后的 DSSM 模型[13]，不仅可以获得低维语义向量表示，同时还能预测两句话的相似度，之后的一部分相似度计算都将该模型作为基础模型。Mingyang Li 等[14]提出一

收稿日期：2023-02-08

种结合 word2vec 模型和 TF-IDF 的语义相似度计算方法,并将其应用于在线医学社区中患者咨询的中文文本数据的密度峰值聚类。K F Xylogiannopoulos 等[15]利用文本挖掘进行代码的相似度检测。Kim Y[16]在对句子进行分类时,使用卷积神经网络提取文本的特征,基于句子相似度对句子进行分类。Wang[17]利用聚类算法,对文本进行语义的扩展,再通过 CNN 进行分类处理。

在新闻文本的处理上,现有研究大致都是基于新闻之间的关联性[18]和语义相关性。Goel N 等[19]在预训练的嵌入模型之上检测多语言新闻的相似性。廖运春等[20]提出一种基于加权 word2vec 和 TextCNN 的新闻文本分类方法。综合以上分析,本文从新闻文本的内容结构出发,考虑到新闻的文本内容较多,但无效的文本也多,一篇新闻中的较大篇幅描述的都是冗余信息,但这些冗余信息都是围绕着特定的主题展开描述。针对新闻文本这一特点,本文考虑先从原始新闻文本中挖掘信息关键词,通过对关键词的相似度计算、加权,得到新闻文本的相似情况。

本文的主要贡献:

(1)基于 TF-IDF 值提取新闻文本中的关键词语,使用 word2vec 和 TF-IDF 加权计算相似度。

(2)在新闻文本数据处理阶段引入聚类,光滑噪声数据,减少无效的计算。

1.2 TF-IDF 算法

TF-IDF 算法通过统计某词语在文档中出现的频率和词语区分文档的能力来判断该词语对文档的重要程度。该算法由两部分组成:TF 算法和 IDF 算法[21]。TF 算法通过统计词语在文档中出现的次数,进而评估该词在文档中的表达能力。一个词语在文档中出现的次数越多,其表达文档的能力就越强。计算方法如下:

$$\mathrm{TF}_{ij}=\frac{n_{ij}}{\sum_{k}^{n}k_j}$$

式中,分子 n_{ij} 表示词语 i 在文档 j 中出现的频率,$\sum_{k}^{n}k_j$ 表示文档 s 中每个词语出现的次数之和。

IDF 算法通过统计文档集中包含某词语的文档数,计算该词语的 IDF 值。其基本思想是如果文档集中包含该词语的文档数较少,该词区分文档的能力就会更强。计算方法如下:

$$\mathrm{IDF}_i=\lg\left(\frac{|D|}{1+|D_i|}\right)$$

式中,$|D|$ 是文档集合中的文档总数,$|D_i|$ 是文档中出现单词 i 的文档数。TF 算法和 IDF 算法分开使用时,这两种算法都有一些弱点。TF 只衡量词语的出现频率,不考虑词语对文档的区分能力;IDF 则相反,强调词语区分文档的能力,却忽略如果一个词语在一篇文档中出现次数较多,就意味着该词能够很好地表达文档的特征。TF-IDF 算法是 TF 算法和 IDF 算法的综合运用,该算法既考虑了词语在文档中出现的频率,同时考虑词语区分文档的能力,能更好地彰显词语对该篇文本的重要程度,计算方法如下:

$$\mathrm{TF}_{\mathrm{IDF}(ij)}=\mathrm{TF}_{ij}\times\mathrm{IDF}_i=\frac{n_{ij}}{\sum_{k}^{n}k_j}\times\lg\left(\frac{|D|}{1+|D_i|}\right)$$

式中,$\mathrm{TF}_{\mathrm{IDF}(ij)}$ 表示词语 i 在文档中的 TF-IDF 值。

1.3 聚类算法

聚类是数据挖掘任务中常用的一种分析手段,可以挖掘出数据库中分布的一些深层信息。聚类算法的核心思想是物以类聚,人以群分,即将相同属性的元素归为一类,不同属性的元素划分为不同的类别。聚类分析在数据挖掘、信息提取、噪声数据光滑等方面有着十分广泛的应用。聚类分析既可以作为一个单独的过程,用于寻找数据内在的分布结构,也可以作为其他任务的前驱过程,即对数据先进行聚类,然后对每个簇单独训练模型。

文本聚类分析是文本挖掘中的一种常用分析方法,以文本相似性为基础,在一个聚类中的模式之间比不在同一聚类中的模式之间具有更多的相似性。

1.4 word2vec 算法

文本中包含着大量的语义信息,能否全面、精确地将文本信息表示出来,这将直接影响整个自然语言处理系统的性能。传统方法使用向量空间模型作为文本语义特征的表示,忽略了特征词的顺序,没有考虑到上下文的语义理解。word2vec 是一种常用的无监督式神经网络词嵌入方法,该算法于 2013 年由谷歌的 Mikolov 提出[12]。word2vec 作为一种高效的自然语言处理算法,将单词表示为低维实值向量,以便测量和挖掘单词之间的关系[22]。该算法以一种无监督的方式,通过大量文本语料库的学习,获得词语的向量表示。也可以将该算法理解为将词语转换为可计算、结构化向量的一种工具,转换后的向量包含文本的语义信息,便于对词语之间的关系进行度量和挖掘。

word2vec 学习语义知识需要训练,训练后得到的词向量可以将文本转化为 k 维空间的向量。word2vec

有两种训练模式:连续词袋(CBOW)模型和 Skip-Gram 模型。其中,连续词袋模型根据上下文来预测当前值,通过词语前后的 m 个词语来获得当前词语的出现概率。即在知道词 w_t 上下文 w_{t-2},w_{t-1},w_{t+1},w_{t+2}的情况下预测当前词 w_t。相反,Skip-Gram 模型则用当前值来预测上下文,通过当前词语来预测其前后 m 个词语的出现概率,即在知道了词 w_t 的情况下,对词 w_{t-2},w_{t-1},w_{t+1},w_{t+2}进行预测。连续词袋模型的训练结果更侧重于对单词语法信息的描述,可以有更高的语法测试准确度;Skip-Gram 模型能更好地区分词的语义,更准确地描述词语的语义特征,得到的语义计算精度更高。相比之下,连续词袋模型计算速度比 Skip-Gram 模型快,但 Skip-Gram 模型在处理文本偏僻词较多的情况下更有优势。两种模型结构如图 1 所示。

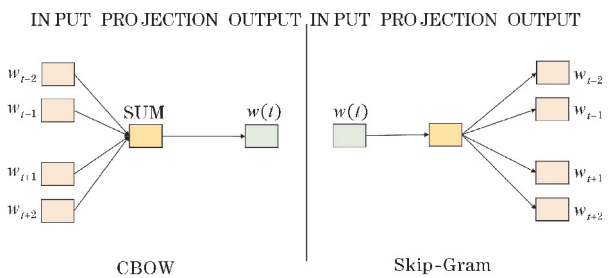

图 1　word2vec 计算模型图

2　文本相似度计算模型

2.1　模型架构

长文本的特点是文字篇幅较长,存在信息冗余的现象。如何挖掘文本的有用信息实现两篇文本的相似度计算是本文考虑的问题之一。数据挖掘是指从大量的、不完全的、有噪声的、随机的数据中,提取潜在有用的信息和知识的过程。为了快速有效地挖掘新闻文本中的有用信息,利用这些有效信息计算两篇新闻文本之间的相似度,本文提出了一种基于关键词聚类后的文本相似度计算方法。

首先,对文本进行分词和去停用词处理,根据词语的 TF-IDF 值筛选出包含信息较多的关键词,去除文本中的冗余信息。基于聚类光滑噪声数据,去除关键词中的孤立点,达到噪声清洗的目的。然后,根据聚类后得到的簇,基于簇心词语间的相似度对关键词簇进行匹配,综合 word2vec 和 TF-IDF 加权计算匹配的两个簇间的相似度。最后,根据簇的相似度计算文本间的相似度。具体模型结构如图 2 所示。

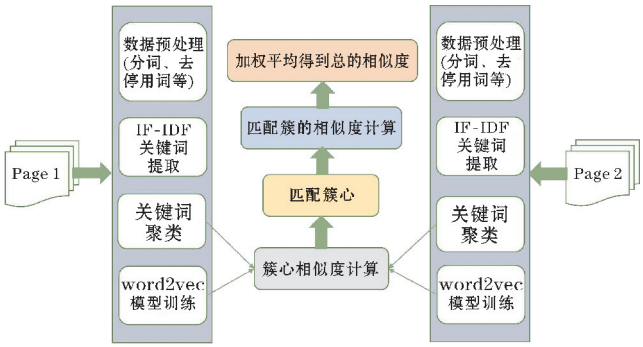

图 2　模型整体结构图

2.2　文本信息挖掘

2.2.1　基于 TF-IDF 值挖掘文本关键词

文本相似度的计算受信息完整度的影响。信息越完整,就能越准确地计算出词与词、文本与文本之间的潜在联系。同时,过多的无关信息也会干扰计算的准确性。一篇中文新闻文本可以看作是一组有序排列的词语,不能说某个词语完全代表该篇新闻,只能说该词语包含一定程度的新闻信息。考虑到新闻文本中的冗余信息较多,选择抽取文本中的关键词作为文本信息的提取。使用权重来表示词语中包含文本信息的多少,词语的权重越大,词语越能代表这个文本,反之亦然。当权重小于某个值(即给定阈值)时,可以认为该词语包含的信息较少,应将其删除。本文使用词语的 TF-IDF 值作为权重,实现关键词的抽取。抽取流程如图 3 所示。

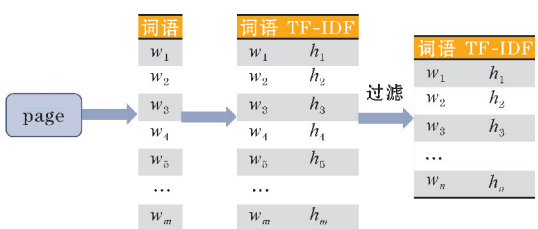

图 3　关键词抽取流程图

首先,对两篇文章进行文本数据清洗、分词、去停用词等预处理,得到文本 A 的词语集合 $A=\{a_1,a_2,a_3,\cdots,a_s\}$,文本 B 的词语集合 $B=\{b_1,b_2,b_3,\cdots,b_t\}$。计算集合中各词语的 TF-IDF 值,分别为 $W_A=\{w_{a_1},w_{a_2},w_{a_3},\cdots,w_{a_s}\}$ 及 $W_B=\{w_{b_1},w_{b_2},w_{b_3},\cdots,w_{b_t}\}$。

将词语的 TF-IDF 值作为词语的权重,TF-IDF 值越高,该词语的权重越高,说明该词语的重要性越高;TF-IDF 值越低,该词语的权重越低,说明该词语的重要性越小。设置权重阈值 α,提取 TF-IDF 值大于该阈值的词语作为文本的关键词。即文本 A 的关键词集合 $A'=\{a_1,a_2,a_3,\cdots,a_n\}$ 及对应权重 $H_A=\{h_{a_1},h_{a_2},$

$h_{a_3},\cdots,h_{a_n}\}$，文本的关键词集合 $B=\{b_1,b_2,b_3,\cdots,b_m\}$ 及对应权重 $H_B=\{h_{b_1},h_{b_2},h_{b_3},\cdots,h_{b_m}\}$。

本文基于 TF-IDF 算法实现文本的关键词抽取。词语的 TF-IDF 值衡量了该词语对该文档的重要程度，使用词语的 TF-IDF 值作为权重，抽取 TF-IDF 大于某阈值的词语作为该篇文档的表达，可以有效避免冗余信息的无效计算。

2.2.2 基于聚类光滑噪声数据

一篇新闻文本应该围绕一个或几个特定主题，从文本中提取到的有用信息也应该是围绕这些主题，偏离主题的词语可以理解为噪声数据。噪声数据会影响后续分析操作的正确性和效果。为了去除这些噪声数据，本文模型引入聚类分析进行文本信息的深度挖掘。

从统计学的观点看，聚类分析是通过数据建模简化数据的一种方法，也是光滑噪声数据的一种有效手段。文本聚类可以挖掘出文本中的潜藏信息，抽取数据中的规律和趋势。对关键词进行聚类，就是基于词语的相似度将词语划分成不同的类别，使同一类别中的词语相似度很高，而不同类别间词语的相似度很低。基于词语的相似度对关键词进行聚类，可以把文章中的相似词语聚成某个类别，而与大部分关键词都不相似的词语便成了孤立点，孤立点即为噪声数据。关键词的聚类算法模型如图 4 所示。

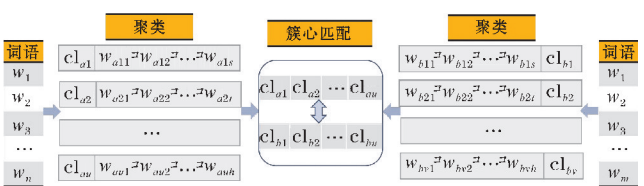

图 4 关键词聚类

在数据特征提取阶段，本文考虑基于相似度分别对两篇文本的关键词进行聚类。在计算词语相似度时，TF-IDF 方法只关注词语出现的频率，忽略了词语间的内部关系。而 word2vec 能很好地表示词语间的语义信息，但忽略了词语的权重信息。为更准确地计算关键词之间的相似度，对文本，首先使用 Word2vec 对关键词进行词向量化处理，计算词语之间的向量距离，融合词语 TF-IDF 值后得到词语和词语之间的相似度 Si_{ij}：

$$S_{ij}=s_{ij}\times h_{a_i}\times h_{a_j}$$

基于相似度 S_{ij} 的聚类标准对关键词集合 A' 进行聚类处理，设置簇内关键词数阈值 α，当簇内关键词数小于 α 时，认为该簇内的词语为孤立点，应当舍弃。提取簇内关键词数大于 α 的簇，即关键词集合 A' 聚类后的簇为 $C_A=\{c_{a1},c_{a2},c_{a3},\cdots,c_{au}\}$，提取各簇的簇心为 $\mathrm{cl}_A=\{\mathrm{cl}_{a1},\mathrm{cl}_{a2},\mathrm{cl}_{a3},\cdots,\mathrm{cl}_{au}\}$。关键词集合 B' 聚类后的簇为 $C_B=\{c_{b1},c_{b2},c_{b3},\cdots,c_{bv}\}$，簇心为 $\mathrm{cl}_B=\{\mathrm{cl}_{b1},\mathrm{cl}_{b2},\mathrm{cl}_{b3},\cdots,\mathrm{cl}_{av}\}$。

根据提取到的簇心，基于 word2vec 计算簇心集合 cl_A 和 cl_B 中各簇心词语的相似度，得到簇心相似度矩阵：

$$\boldsymbol{S}_{\mathrm{cl}}=\begin{bmatrix} S(\mathrm{cl}_{a1},\mathrm{cl}_{b1}) & \cdots & S(\mathrm{cl}_{a1},\mathrm{cl}_{bv}) \\ \vdots & \vdots & \vdots \\ S(\mathrm{cl}_{av},\mathrm{cl}_{b1}) & \cdots & S(\mathrm{cl}_{au},\mathrm{cl}_{bv}) \end{bmatrix}$$

找出簇心相似度矩阵 S_{cl} 中最大的元素 $S(\mathrm{cl}\,ai,\mathrm{cl}_{bj})$，该值对应簇心所属簇即为匹配簇。删除 c_{ai},c_{bj} 所在的矩阵行和列。继续寻找剩余矩阵元素中的最大值，匹配最大值所属的两个簇心，直至所有的簇都匹配。当最大相似度小于 0 时，直接删除该组簇心。最终得到匹配的簇集合 $P=\{c_{a1}\leftrightarrow c_{b1},c_{a2}\leftrightarrow c_{b2},\cdots,c_{au}\leftrightarrow c_{bv}\}$。

2.3 相似度计算

本文将挖掘到的文本信息聚集在关键词簇中，通过计算两篇文本关键词簇间的相似度，进而得到两篇文本间的相似度。而簇间的相似度又取决于关键词的相似度，对于关键词间的相似度计算，本文首先使用 word2vec 将关键词向量化，在计算相似度时，融入 TF-IDF 值作为词语权重，同时关注到词语中的语义信息和词语的频率。

2.3.1 融合词语权重的簇间相似度计算

基于簇的匹配计算两个匹配簇 $c_{ai}\leftrightarrow c_{bj}$ 之间的相似度。组成簇的基本单位是关键词，计算簇的相似度本质是计算关键词的相似度。本文综合考虑词语的语义信息和词语出现频率，使用融入语义信息的 word2vec 将词语向量化，基于 TF-IDF 值加权计算，得到两个关键词的最终相似度。

对于簇 $c_{ai}=\{w_{a1},w_{a2},w_{a3},\cdots,w_{af}\}$，计算出其各关键词的 TF-IDF 值分别为 $H_{wai}=\{h_{wa1},h_{wa2},h_{wa3},\cdots,h_{waf}\}$。基于 TF-IDF 值计算权重，各关键词的权重值 $Q_{wai}=\{q_{wa1},q_{wa2},q_{wa3},\cdots,q_{waf}\}$，其中 $q_{wai}=\dfrac{h_{wai}}{\sum_{e=1}^{f}h_{wae}}$。同样地，对于簇 c_{bj}，其关键词的 TF-IDF 值 $H_{wbi}=\{h_{wb1},h_{wb2},h_{wb3},\cdots,h_{wbf}\}$，各关键词的权重值 $Q_{wbj}=\{q_{wb1},q_{wb2},q_{wb3},\cdots,q_{wbg}\}$。使用预训练后的 word2vec 将关键词向量化，计算簇 c_{ai} 中的每个词语与簇 c_{bj} 中的每个词语的相似度，得到簇 c_{ai} 与簇 c_{bj} 相似度矩阵：

$$\boldsymbol{S}_{ca}=\begin{bmatrix} S(w_{a1},w_{b1}) & \cdots & S(w_{a1},w_{bg}) \\ \vdots & \vdots & \vdots \\ S(w_{af},w_{b1}) & \cdots & S(w_{af},w_{bg}) \end{bmatrix}$$

其中 $S(w_{ai},w_{bj})=\mathrm{sim}(w_{ai},w_{bj})\times q_{wai}$。

簇 c_{bj}与簇 c_{ai}相似度矩阵:

$$\boldsymbol{S}_{cb}=\begin{bmatrix} S(w_{b1},w_{a1}) & \cdots & S(w_{b1},w_{ag}) \\ \vdots & \vdots & \vdots \\ S(w_{bf},w_{a1}) & \cdots & S(w_{bf},w_{ag}) \end{bmatrix}$$

其中 $S(w_{ai},w_{bj})=\mathrm{sim}(w_{bj},w_{ai})\times q_{wbj}$。

分别找出簇 c_{ai}中每个词语在簇 c_{bj}中相似度最高的关键词和簇 c_{bj}中每个词语在 c_{ai}中相似度最高的关键词。即 $\boldsymbol{S}_{ca}$矩阵中每一行的最大值 $u_h=\max\boldsymbol{S}_{ca}(w_{ah},w_{bk})$,$k=1,2,3,\cdots,g$。以及 $\boldsymbol{S}_{cb}$矩阵中每一行的最大值 $u_k=\max\boldsymbol{S}_{cb}(w_{bk},w_{ah})$,$h=1,2,3,\cdots,f$。两个匹配簇之间的相似度为

$$S_{c_{ai}\leftrightarrow c_{bj}}=\frac{1}{2}\left(\sum_{h=1}^{s}u_i+\sum_{k=1}^{t}u_j\right)$$

式中,$h_{w_{ah}}$、$h_{w_{bk}}$表示词语 w_{ah}、w_{bk}的对应权重,f、g 为簇 c_{ai}、簇 c_{bj}内关键词的个数。

2.3.2　基于簇的文本相似度计算。

对文本进行深度挖掘处理后,提取到的文本信息蕴含于关键词簇中,计算两篇文本的相似度,只需计算两篇文本对应簇之间的相似度。基于簇的相似度计算得到两个匹配簇的相似度 $S_{c_{ai}\leftrightarrow c_{bj}}$后,以簇内词语数量占文本关键词总数的比例为权重,得到文本 A 与文本 B 之间的相似度 Sim 为

$$\mathrm{Sim}=\sum_{(i,j)=(1,1)}^{(u,v)}\frac{f+g}{s+t}s_{c_{ai}\leftrightarrow c_{bj}}$$

式中,s、t 为文本 A、文本 B 中关键词的个数。

模型以簇为单位计算文本相似度,在一定程度上能减少无效的计算。在簇间相似度计算上,以 TF-IDF 值为权重,同时考虑词语的语义信息,能更好捕捉词语间的相似度关系,模型的可解释性更强。

3　实验及其分析

为验证模型的有效性,本文选取搜狐新闻发布的公开数据集中的 B 类数据作为实验数据集,并进行了从数据处理到结果分析的一系列实验。

3.1　word2vec 预训练

实验使用 word2vec 进行词向量化。针对其两种计算模型:Skip-Gram 计算模型学习的时间长,适合语料库中有大量的低频词语,CBOW 计算模型学习效率高,适用于陌生词语较少的语料库。考虑本实验的数据集为新闻文本数据集,新闻文本中的词语基本为日常用语,罕见词语较少因此选择使用 word2vec 模型中的 CBOW 进行模型构建和训练。为提高 word2vec 训练的泛化能力,实验首先使用维基百科数据集训练词向量模型,之后再使用搜狐数据集对模型进行增量训练。

3.2　基于 k-means 的聚类分析

k 均值聚类算法作为使用最为普遍的一种聚类分析算法,该算法以某种距离 l 为依据,将数据集 $D=\{d_1,d_2,d_3,\cdots,d_n\}$ 划分为 k 个簇。划分后的簇为 $C=\{c_1,c_2,c_3,\cdots,c_k\}$,在该划分下,平方误差和(SSE)最小,其中:

$$\mathrm{SSE}=\sum_{i=1}^{k}\sum_{s\in c_i}\|s-\mathrm{cl}_i\|^2$$

式中,cl_i 表示各个簇的簇心,即簇内的各数据点到簇心距离和越小,簇内越紧密。

本文选用 k 均值聚类对文本关键词进行聚类分析,实现数据噪声的清洗和数据模型的特征融合。通过计算平均轮廓系数得到聚类的最佳 k 值,具体方法流程如图 5 所示。其中,S 为初始的平均轮廓系数,bestk 为最佳的类别值,k 为初始的类别值。

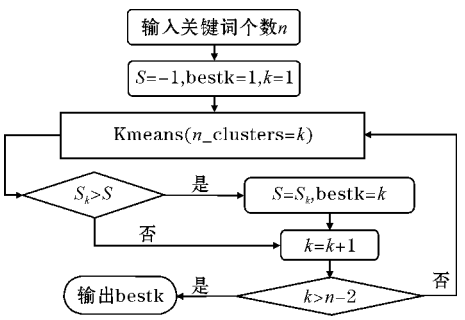

图 5　寻找最佳 K 值流程图

3.3　关键词抽取实验

词语的 TF-IDF 一方面是筛选关键词的标准,另一方面是词语相似度计算的权重。设置的 TF-IDF 阈值过高,可能导致提取的文本关键词不能很好地代表该文本;设置过低,又可能会导致提取太多与文本不相干的关键词,加重计算的复杂度和计算时间。因此,选择合适的 TF-IDF 阈值对准确计算文本相似度尤为重要。关键词的数量对模型的效果也有很大的影响,在给定阈值的前提下抽取关键词,如果大于该阈值的词语数量较少,而过少的关键词很难包含文本的大部分信息,可能导致信息提取不充分,影响计算效果。

设置 TF-IDF 阈值分别为 0.1、0.15、0.2、0.25、0.3,选取能够获得关键词数量大于 5、10、15、20 的数据集进行实验,图 6 展示了不同情况下的实验效果。

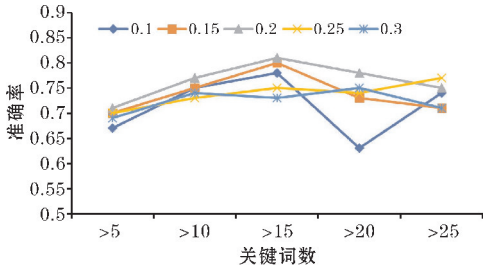

图 6 关键词效果图

从图 6 可以看出,当阈值取0.2,关键词数>15 时,该数据集下模型的性能最好。

3.4 算法对比实验及结果分析

为验证本模型的效果,实验选择准确率、召回率、*F*1 值和模型计算时间作为评价指标验证模型性能的优劣。实验选择本文介绍的算法模型与传统的 word2vec 算法、TF-IDF 算法相关模型在数据集上进行对比实验,各模型实验效果如表 2 所示。

表 2 不同模型性能比较 单位:%

相似度计算模型	准确率	召回率	*F*1
Kmeans+word2vec+tf-idf	80	77	79
word2vec+tf-idf	75	76	69
word2vec	72	77	70

从表 2 可以看出,相比以往模型,在文本数据信息的挖掘上加入聚类后,计算方法整体效果提升较好。模型中加入聚类算法,一方面能有效清除噪声数据,提高相似度计算的准确性,另一方面,大大减少了无效的计算。相比于仅仅使用 word2vec 计算关键词语相似度,加入以 TF-IDF 为权重的相似度计算方法能同时结合文本的语义信息和词语出现频率,在计算准确率上要优于仅使用一种方法度量的模型。

4 结束语

本文针对新闻文本数据的特点,首先挖掘新闻文本的关键信息,使用 TF-IDF 值作为词语的权重抽取关键词,基于关键词聚类光滑噪声数据。然后通过对这些关键信息的相似度计算衡量两篇新闻之间的相似度,在词语相似度计算上,从多个方面提取文本信息,使用 word2vec 将词向量化,关注到词语间的语义信息,基于 TF-IDF 值计算加权,关注词语频率。在文本相似度计算上,以关键词簇为单位计算文本的相似度,减少无效的计算。实验结果表明,本文提出的方法在计算效果上优于以往传统的长文本相似度计算方法。本文模型还存在一些不足之处,如相似度计算的效果取决于文本信息挖掘的效果,即关键词的抽取效果,但在关键词抽取时时仅考虑了词语的 TF-IDF 值,后续工作可以考虑在关键词抽取中加入文本的主题词,在相似度计算上融入深度学习的计算方法。

参考文献:

[1] Berant J,Chou A,Frostig R,et al. Semantic Parsing on Freebase from Question-Answer Pairs[C]. Empirical Methods in Natural Language Processing. Association for Computational Linguistics,2013.

[2] WT Yih,X He,C Meek. Semantic Parsing for Single-Relation Question Answering[C]. Meeting of the Association for Computational Linguistics. 2014.

[3] Shen Y,He X,Gao J,et al. Learning semantic representations using convolutional neural networks for web search[C]. Proceedings of the 23rd international conference on world wide web. 2014.

[4] Heidari M,Zad S,Rafatirad S. Ensemble of Supervised and Unsupervised Learning Models to Predict a Profitable Business Decision[C]. 2021 IEEE International IOT,Electronics and Mechatronics Conference(IEMTRONICS). IEEE,2021.

[5] Vani K,Gupta D. Detection of idea plagiarism using syntax-Semantic concept extractions with genetic algorithm[J]. Expert Systems with Applications,2017,73:11-26.

[6] El Mostafa H,Benabbou F. A deep learning based technique for plagiarism detection: a comparative study[J]. IAES International Journal of Artificial Intelligence,2020,9(1):81.

[7] 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.

[8] Levenshtein V I. Binary codes capable of correcting deletions,insertions,and reversals[J]. Soviet physics doklady,1966,10(8):707-710.

[9] Melamed I D. Automatic evaluation and uniform filter cascades for inducing n-best translation lexicons[J]. arXiv preprint cmp-lg/9505044,1995.

[10] 张焕炯,王国胜,钟义信. 基于汉明距离的文本相似度计算[J]. 计算机工程与应用,2001(19)

21-22.

[11] Salton G. A vector space model for automatic indexing[J]. Communications of the ACM, 1975, 18(11):613-620.

[12] 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, 26:311-319.

[13] 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.

[14] Li M, Bi X, Wang L, et al. Text Similarity Measurement Method and Application of Online Medical Community Based on Density Peak Clustering [J]. Journal of Organization and End User Computing(JOEUC), 2022, 34(2):1-25.

[15] Xylogiannopoulos K F, Karampelas P. Identifying Social Networks of Programmers using Text Mining for Code Similarity Detection[C]. 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), The Hague, Netherlands, 2020:643-650.

[16] Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv: 1408.5882, 2014.

[17] Peng W. Semantic Clustering and Convolutional Neural Network for Short Text Categorization[J]. Neurocomputing, 2016, 174:806-814.

[18] Ahn D. The stages of event extraction[C]. Proceedings of the Workshop on Annotating and Reasoning about Time and Events. 2006:1-8.

[19] Goel N, Reddy R. SemEval-2022 Task 8: Multilingual News Article Similarity[J]. arXiv preprint arXiv:2208.09715, 2022.

[20] 廖运春,舒坚. 基于加权 Word2Vec 和 TextCNN 的新闻文本分类[J]. 长江信息通信, 2022, 35(9):32-35.

[21] Aizawa A. An information-theoretic perspective of tf-idf measures[J]. Information Processing & Management, 2003, 39(1):45-65.

[22] Khatua A, Cambria E. A tale of two epidemics: Contextual Word2Vec for classifying twitter streams during outbreaks[J]. Information Processing & Management, 2019, 56(1):247-257.

News Text Similarity Calculation based on Keyword Clustering

ZHU Ting,　HU Jiancheng

(College of Applied Mathematics, Chengdu University of Information Techology, Chengdu 610225, China)

Abstract: Aiming at the problems of long news text, too much redundant information, and difficulty in accurately and efficiently calculating text similarity, a news text similarity calculation method based on keyword clustering is proposed. First, the text data is preprocessed to extract the key information in the text. The weighted sampling method weighted by TF-IDF values was used to extract keywords in the text dataset, and the clustering-based method was used to smooth noise data. After getting clusters from clustering, word2vec is used to calculate the word similarity between clusters, and the TF-IDF word weighting calculation method is used, and the semantic information and word frequency between words are considered. Finally, the similarity of the two texts is calculated based on the similarity of each cluster. Experiments show that the proposed news text similarity calculation method performs better than the traditional calculation method.

Keywords: news text similarity; word2vec; TF-IDF; keyword clustering