说　明　书

一种基于正则变分嵌入式的软件需求聚类方法及系统

技术领域

本发明涉及数据挖掘领域，特别是涉及一种基于正则变分嵌入式的软件需求聚类方法及系统。

背景技术

软件开发流程包括需求分析、系统设计、详细设计、测试和评估，想要设计出一款好的软件，首要工作便是需求分析，但实际开发过程中，人们忽略了需求分析的重要性，将关注点放在了设计阶段。需求分析引起的错误在软件开发过程中是看不到的，只要在测试阶段才会被发现，但这时修改错误将付出双倍的代价。软件需求描述不仅是用户和开发者沟通的桥梁，更是功能设计和性能指标的依据，它贯穿于整个软件开发过程，需求会随着时间推移发生变更，这会给后期开发带来巨大的风险。

需求分析阶段往往存在以下问题：（1）用户和开发人员所在研究领域的差异，两者熟悉自己的领域但是却对对方的领域很陌生，导致用户和开发人员需求沟通存在障碍；（2）软件需求描述中潜在需求未能挖掘出来，由于用户理解的计算机领域的专业知识有限，所要求的功能和性能设计表达不完整，导致在需求分析时可能遗漏；（3）软件需求文本描述模糊，存在稀疏性、歧义性、不可验证性等问题，同种意思的文本表达在不同领域的理解也有所区别，文本描述简短的缺点具体体现在功能描述冗余，开发流程繁琐，模块耦合性较高等方面，导致用户体验感和可操作性较差，软件成本和开发效率无法得到保证，从而造成项目失败。

需求分析阶段的问题给软件开发造成了不便，如果通过聚类方法将相似性描述聚集起来，只需要关注同一个类内的描述，那就可以指导这个类代表的是软件哪方面的功能以及应用领域。

传统的文本聚类方法首先要进行文本预处理，然后将对预处理过后的单词序列进行特征提取，再由词向量工具映射到向量空间上，基于度量函数完成样本划分。传统的特征提取技术是基于统计分析的，利用评估函数为已有的特征

参数进行权值分配，特征权重代表了单词在整句话中的重要程度，但是这种方式往往忽略了序列的位置关系和语义信息，文本中的单词被当作孤立的个体，仅依靠频率无法准确表达语义，忽略了上下文关系。

常用的聚类方法大部分基于划分法、密度法和层次法，例如 K-means、DBSCAN、Agglomerative Clustering 等。已知的聚类方法在文本领域和图像领域都达到了不错的效果，但这未必能说明聚类方法适用于所有的领域知识划分，传统聚类还是极其依赖于所提取特征的质量和词向量模型的训练。深度学习的引入开始将工作集中在学习表征上，以深度模型学习文本的特征表达，相比于传统的特征提取非线性和拟合能力更强，所提取的特征在传统聚类上有大幅度提升，通过网络参数的调整，可以取得较高的聚类准确率。但是大部分深度聚类是基于两段式聚类，首先对文本进行特征提取，然后将压缩特征由传统聚类划分。两段式聚类结构清晰，特征提取器将高维特征压缩成低维特征，降低了数据稀疏性，包含了更多的文本语义，对于聚类具有更好的连续性和解释性，但是两段式聚类的缺点也很明显，就是表征学习和聚类过程分两个步骤进行，聚类中心和提取的特征不能根据聚类结果改善，且还是存在易陷入局部最优的缺点，前向聚类过程是一次性的，只能小幅度改善聚类输出。

传统的文本聚类方法易受初始聚类中心的影响，且大部分基于距离度量，方法依旧存在易陷入局部最优的缺点，深度聚类结合了深度学习技术和传统聚类方法，模型结构容易了解，聚类结果相较于传统聚类有小幅提升，但是大部分深度聚类方法基于两段式聚类，不能反向传播优化聚类中心和样本分布。

目前有关软件需求文本的聚类方法很稀少，大部分聚类方法重点关注方法的改进，而忽略了特征提取方式的优化，实际情况是并不是所有的聚类方法都具有普适性，往往都是根据数据分布而采取使用某种特征提取方式和聚类方法，而特征提取器往往也是难以抉择的，从某种程度来说间接决定样本划分。由于属于无监督训练，词向量模型的训练也至关重要，但大规模的软件需求语料库很少，没有好的词向量，聚类方法将变得毫无意义。

软件需求文本是一堆杂乱无章的非结构化数据，存在诸多冗余信息，因此无法输入到机器中学习，我们需要对单词缩略词更改、拼写校正、词干提取和

词形还原，传统的特征提取器是线性映射，结果缺乏合理的解释和描述，文本映射成计算机可以识别的向量形式，才可以进行聚类划分。软件需求数据经过分析后，根据软件需求数据的样本分布，判断采用哪种聚类方法以及特征提取器，以及反向传播的神经网络聚类模型如何设计，同时我们还要保证特征向量不会受聚类损失导致嵌入空间损坏，保留样本的局部结构。

因此，亟需一种软件需求聚类方法，针对软件需求文本存在离散度高、噪声大和数据稀疏等特点，提高软件需求文本聚类的准确性。

发明内容

本发明的目的是提供一种基于正则变分嵌入式的软件需求聚类方法及系统，提高软件需求文本聚类的准确性。

为实现上述目的，本发明提供了如下方案:

一种基于正则变分嵌入式的软件需求聚类方法，包括:

获取不同类别软件的软件需求数据;

对所述软件需求数据进行文本预处理，确定软件需求文本;

利用 BERT 预训练的句向量模型将所述软件需求文件映射到向量空间，确定句向量;

利用正则变分嵌入式聚类模型对所述句向量进行聚类，确定聚类结果;

利用正则变分嵌入式聚类模型对所述句向量进行聚类的步骤为:

所述正则变分嵌入式聚类模型对所述句向量进行 Dropout 正则化处理，确定正则化向量;

利用全连接层对所述正则化向量进行特征压缩; 并根据所述压缩后的正则化向量，采用编码器确定嵌入特征;

利用解码器对所述嵌入特征进行解码，确定原始向量;

根据所述嵌入特征，采用 K-means 算法确定聚类划分结果;

根据所述嵌入特征、所述原始向量以及所述聚类划分结果确定相应的损失函数，并对所述嵌入特征、所述原始向量以及所述聚类划分结果相应的损失函

数进行反向传播，确定所述聚类结果。

可选地，所述获取不同类别软件的软件需求数据，具体包括:

利用 Scrapy 技术获取 Softpedia 网站下下 Windows 平台的 11 类软件的软件需求数据;

将每类软件需求数据以 csv 的格式单独存储，同时对每类软件需求数据进行标注。

可选地，所述对所述软件需求数据进行文本预处理，确定软件需求文本，具体包括:

利用所述正则表达式剔除所述软件需求数据中的 html 标签;

将剔除 html 标签后的软件需求数据进行缩略词以及乱码单词的校正;

对校正后的软件需求数据进行词干提取以及词形还原;

将处理后的数据在 csv 文件中进行存储。

可选地，所述利用全连接层对所述正则化向量进行特征压缩；并根据所述压缩后的正则化向量，采用编码器确定嵌入特征，具体包括:

利用公式$z = u + exp(\delta) * epsilon$确定嵌入特征;

其中，u和δ为编码器将压缩后的正则化向量转换为隐藏空间的两个参数，分别为均值和方差，$epsilon$为一个服从正态分布的张量，$epsilon \sim N(0,1)$。

可选地，所述利用正则变分嵌入式聚类模型对所述句向量进行聚类，确定聚类结果，之后还包括:

利用聚类指标对所述聚类结果进行评价。

一种基于正则变分嵌入式的软件需求聚类系统，包括:

软件需求数据获取模块，用于获取不同类别软件的软件需求数据;

文本预处理模块，用于对所述软件需求数据进行文本预处理，确定软件需求文本;

句向量确定模块，用于利用 BERT 预训练的句向量模型将所述软件需求文件映射到向量空间，确定句向量;

聚类结果确定模块，用于利用正则变分嵌入式聚类模型对所述句向量进行聚类，确定聚类结果；

利用正则变分嵌入式聚类模型对所述句向量进行聚类的步骤为：

所述正则变分嵌入式聚类模型对所述句向量进行 Dropout 正则化处理，确定正则化向量；

利用全连接层对所述正则化向量进行特征压缩；并根据所述压缩后的正则化向量，采用编码器确定嵌入特征；

利用解码器对所述嵌入特征进行解码，确定原始向量；

根据所述嵌入特征，采用 K-means 算法确定聚类划分结果；

根据所述嵌入特征、所述原始向量以及所述聚类划分结果确定相应的损失函数，并对所述嵌入特征、所述原始向量以及所述聚类划分结果相应的损失函数进行反向传播，确定所述聚类结果。

可选地，所述软件需求数据获取模块具体包括：

软件需求数据获取单元，用于利用 Scrapy 技术获取 Softpedia 网站下下 Windows 平台的不同类别软件的软件需求数据；

软件需求数据存储标注单元，用于将每类软件需求数据以 csv 的格式单独存储，同时对每类软件需求数据进行标注。

可选地，所述文本预处理模块具体包括：

文本剔除单元，用于利用所述正则表达式剔除所述软件需求数据中的 html 标签；

文本校正单元，用于将剔除 html 标签后的软件需求数据进行缩略词以及乱码单词的校正；

文本提取单元，用于对校正后的软件需求数据进行词干提取以及词形还原；

文本存储单元，用于将处理后的数据在 csv 文件中进行存储。

可选地，还包括：

聚类评价模块，用于利用聚类指标对所述聚类结果进行评价。

根据本发明提供的具体实施例，本发明公开了以下技术效果:

本发明所提供的一种基于正则变分嵌入式的软件需求聚类方法及系统，利用全连接层对所述正则化向量进行特征压缩；并根据所述压缩后的正则化向量，采用编码器确定嵌入特征；上述方案解决了两端式聚类不能反向传播优化聚类中心和样本分布的问题，通过学习软件需求数据的内部隐藏分布，保留数据局部结构，以重参数的技巧模拟原始数据，反映数据本质特征，提高了特征质量和聚类准确率，达到了最优效果。在文本表示上，使用 BERT 句向量模型将原始文本映射到向量空间，由于原始文本未经噪声处理，所以在模型输入端融合 Dropout 正则化随机抑制部分神经元的工作，降低噪声干扰，防止模型过拟合，增强模型鲁棒性，经过 Dropout 正则化的向量输入到变分嵌入式聚类方法中，通过编码器使嵌入空间可以学习原始数据分布，重参数技巧确保隐藏层能较好抽象输入数据特点，然后嵌入空间向量一方面进行解码进行特征学习，另一方面使用 K-means 聚类方法进行样本划分，通过反向传播优化损失函数，待达到迭代停止条件输出聚类结果。

附图说明

为了更清楚地说明本发明实施例或现有技术中的技术方案，下面将对实施例中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动性的前提下，还可以根据这些附图获得其他的附图。

图 1 为本发明所提供的一种基于正则变分嵌入式的软件需求聚类方法流程示意图;

图 2 为本发明所提供的一种基于正则变分嵌入式的软件需求聚类的算法流程图;

图 3 为本发明所提供的一种基于正则变分嵌入式的软件需求聚类的网络架构图;

图 4 为实施例中 Dropout 正则化的示意图;

图 5 为传统聚类对比示意图;

图 6 为深度聚类对比示意图;

图 7 为本发明所提供的一种基于正则变分嵌入式的软件需求聚类系统结构示意图。

具体实施方式

下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

本发明的目的是提供一种基于正则变分嵌入式的软件需求聚类方法及系统，提高软件需求文本聚类的准确性。

为使本发明的上述目的、特征和优点能够更加明显易懂，下面结合附图和具体实施方式对本发明作进一步详细的说明。

图 1 为本发明所提供的一种基于正则变分嵌入式的软件需求聚类方法流程示意图，如图 1 所示，本发明所提供的一种基于正则变分嵌入式的软件需求聚类方法，包括:

S101，获取不同类别软件的软件需求数据。

S101 具体包括:

利用 Scrapy 技术获取 Softpedia 网站下下 Windows 平台的 11 类软件的软件需求数据。

将每类软件需求数据以 csv 的格式单独存储，同时对每类软件需求数据进行标注。将功能相同的软件需求描述以 csv 的格式输出存储，同时为功能相同的软件需求描述信息进行打标签。

由于网上关于软件需求描述的开源数据很少，所以采用 Scrapy 爬虫技术在 Softpedia 网站下获取 Windows 平台下的 11 类数据集，这 11 类数据分别是 Antivirus 、 authoring-Tools 、 CD-DVD-Blu-ray-Tools 、 Compression-tools 、

Desktop-Enhancements、File-managers、Gaming-Related、iPod-Tools、Maps&GPS、Mobile-Phone-Tools、Network-Tools，由于防爬程序设置 IP 代理池和时间延迟，每类数据以 csv 的格式单独存储，同时对 11 类数据中的样本进行标注，类别名按照 Ascii 码排序，然后以数字 0-10 的形式给同类的样本中打标签，最红将标注好的 11 类数据合成一个文件。

S102，对所述软件需求数据进行文本预处理，确定软件需求文本。

S102 具体包括:

利用所述正则表达式剔除所述软件需求数据中的 html 标签。

将剔除 html 标签后的软件需求数据进行缩略词以及乱码单词的校正。

对校正后的软件需求数据进行词干提取以及词形还原。

将处理后的数据在 csv 文件中进行存储。

由于软件需求数据是直接在网站上获取的，无法避免 html 标签内容，标签的格式有以下几种： ，<img />，自定义标签<My-Tag></My-Tag>，使用正则表达式 re.compile(“/<V?.+?V?>/g”)等去除英文文本中的 html 标签，表达式中<表示尖括号,第一个 V?表示 这种标签的情况,第二个 V?表示<img />这种情况；后更改文本中的缩略词，例如 re.sub(r"can\'t", "can not", text)，使用 pyenchant 库拼写检查，找出错误后改正；使用 SnowballStemmer 完成词干提取，例如 protecting 提取后为 protect，WordNetLemmatizer 完成词形还原,最终将所有类的数据放在一个 csv 文件中

S103，利用 BERT 预训练的句向量模型将所述软件需求文件映射到向量空间，确定句向量；句向量代表整个句子的文本信息，考虑了单词前后位置关系和内部关联。BERT 预训练的句向量模型输出维度相同的句向量表达，原始软件需求描述经过训练输出软件需求描述记录数*维度数的向量空间。

S104，利用正则变分嵌入式聚类模型对所述句向量进行聚类，确定聚类结果；对于处理好的句向量输入到正则变分嵌入式聚类模型中，通过编码将高维稀疏特征压缩成低维密集向量，以概率形式生成于原始数据类似的样本，用来反映数据本质特征，然后新生成的向量同时进行特征学习和聚类划分。

如图 2 和图 3 所示，利用正则变分嵌入式聚类模型对所述句向量进行聚类的步骤为：

S401，所述正则变分嵌入式聚类模型对所述句向量进行 Dropout 正则化处理，确定正则化向量。

在输入端融合 Dropout 正则化，按照一定概率将神经网络的部分神经元停止工作，防止模型对数据中不重要的特征进行贪婪学习，这会造成特征在压缩过程中融合过多噪声，对文本聚类造成损失，随机抑制神经元相当于降噪，在模型学习时使模型具有鲁棒性，增强模型泛化能力。神经网络训练过程无法逃避两个问题：（1）费时；（2）容易过拟合。融合 Dropout 恰好解决了这两个问题，它随机停止一些神经元不参与训练，其余神经元共同工作，简化了网络模型，降低了时间复杂度，消除减弱了神经元节点间的联合适应性。假设输入向量集合为$x=[x_1,x_2,x_3,...,x_m]$，其中x_i代表每个样本，隐藏层向量为$z=[z_1,z_2,z_3,...,z_k]$，重构向量为$x^{'}=[x_1^{'},x_2^{'},x_3^{'},...,x_m^{'}]$，网络的全连接层的激活函数为$relu$函数。在输入端融合正则化，通过 Dropout 随机映射成正则化向量$\tilde{x}$：

$x \xrightarrow{Dropout} \tilde{x}$。

S402，利用全连接层对所述正则化向量进行特征压缩；并根据所述压缩后的正则化向量，采用编码器确定嵌入特征。

S402 具体包括：

利用公式$z=u+exp(\delta)*epsilon$确定嵌入特征

其中，u和δ为编码器将压缩后的正则化向量转换为隐藏空间的两个参数，分别为均值和方差，$epsilon$为一个服从正态分布的张量，$epsilon \sim N(0,1)$。

S103 采用全连接层对高维稀疏化向量进行特征压缩，编码器网络施加一个约束项，让网络生成服从高斯分布的样本集，利用 Reparameterization trick 按照高斯分布里的均值 μ和方差δ规则就可以取任意相关的数据，使嵌入空间学习原始数据分布，δ动态调节噪声的强度，然后解码层重构样本，生成与原始样本相似的数据：

$$u,\delta = f(W_z\tilde{x} + b_z)$$
;

$$epsilon \sim N(0,1)$$
;

$z = u + exp(\delta) * epsilon$。

S403，利用解码器对所述嵌入特征进行解码，确定原始向量；将重参数形成的嵌入特征z解码，解码同样采用全连接层，解码过程与编码过程相反，经过解码还原成原始向量x'。

$x' = f(W_{x'}z + b_{x'})$。

S404，根据所述嵌入特征，采用 K-means 算法确定聚类划分结果；嵌入特征z作为 K-means 算法的输入，K-means 按照近邻原则划分样本到最近的类中，即满足$distance(x_i,c_j) = min\{distance(x_i,c_j)$，$j = \{1,2,3,...,k\}$，则$x_i \in y_i$。每轮训练重新计算聚类中心$c_j$:

$$c_j = \frac{1}{n}\sum_{x_i \epsilon y_i} x_i$$
。

利用t分布刻画嵌入特征z_i与聚类中心c_j的相似度:

$$q_{ij} = \frac{(1 + \|z_i - c_j\|^2)^{-1}}{\sum_j (1 + \|z_i - c_j\|^2)^{-1}};$$

利用该相似度，定义目标分布为:

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j (q_{ij}^2 / \sum_i q_{ij})}$$
。

S405，根据所述嵌入特征、所述原始向量以及所述聚类划分结果确定相应的损失函数，并对所述嵌入特征、所述原始向量以及所述聚类划分结果相应的损失函数进行反向传播，确定所述聚类结果。

S105 为优化损失函数，经过解码和聚类划分后，通过反向传播的形式优化损失函数，损失函数涉及三个，样本重构产生的损失，聚类产生的损失，重

参数产生新样本的损失:

$$Reconstruction\,Loss = ||x - x'||^2;$$

$$Cluster\,Loss = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}};$$

$$KL\,Loss = KL(N(z;\mu,\sigma^2)||N(z;0,1)) = \frac{1}{2}\sum_{j=1}^{J}(1 - \log\sigma_j^2 + \mu_j^2 + \sigma_j^2)$$

。

损失函数定义为: $L = KL\,Loss + Reconstruction\,Loss + \alpha \cdot Cluster\,Loss$。

S104 之后还包括:

利用聚类指标对所述聚类结果进行评价。

以“少数服从多数”的原则计算统计每个类的样本，同时使用 Silhouette coefficient（SC）、Calinski Harabasz（CH）和 Davies-Bouldin index（DBI）等评价指标来衡量方法好坏，具体的公式如下:

$$SC(i) = \frac{b(i) - a(i)}{max\{a(i),b(i)\}};$$

$$CH(k) = \frac{tr(B_k)/(k-1)}{tr(W_k)/(m-k)};$$

$$DB = \frac{1}{N}\sum_{i=1}^{N} \underset{j \neq i}{max}\left(\frac{\sigma_i + \sigma_j}{d(c_i,c_j)}\right)$$

。

作为一个具体的实施例，本发明采用 Scrapy 爬虫技术获取 Softpedia 平台上的软件需求，这些需求都是对软件功能的客观描述。Softpedia 平台为用户提供了各种系统平台下的各种工具，这些系统包括 Windows、Linux、Android 和 IOS 等几乎所有的平台。每个平台下有杀毒软件、压缩软件、文件管理、游戏软件和地图定位软件等几十种软件类别，不同平台下的工具使用描述也会有所差异。截至北京时间 2019 年 5 月 27 号，该网站总共收录应用程序有 113016 个，应用程序被下载的总次数为 3320687342 次。

本发明的软件需求数据为 Windows 平台下的软件功能描述，该平台下拥有不同类别的软件，每个软件类别下拥有大量的 App，每个 App 的功能信息和作用都有对应的描述信息，我们一共爬取了 11 类软件数据，共 15598 条，数据长度各不相同，爬取到的软件类型和数量如表 1 所示。

表 1 软件需求数据表

软件类别	数量
Antivirus	625
Authoring-Tools	676
CD-DVD-Blu-ray-Tools	1545
Compression-tools	466
Desktop-Enhancements	7612
File-managers	546
Gaming-Related	114
iPod-Tools	174
Maps&GPS	25
Mobile-Phone-Tools	433
Network-Tools	3382

本发明主要在聚类准确率方面进行对比，采用纯度作为聚类准确率，公式如下：

$$acc = \frac{1}{N}\sum_{k}\max_{j}|c_k \cap t_j|$$

。

本发明将不同长度的文本经过 BERT 训练后，每个样本特征维度是 768 维，由于文本长度不同所带来的向量不规则问题，且考虑了上下文语义信息，为了验证 BERT 句嵌入模型优于平均向量，本文在传统聚类算法 DBSCAN，Spectral Clustering，Hierarchical Clustering，GMM，K-means 和 SOM 与本发明进行对比，如图 5 所示.

根据图 5 可知，本文的软件需求数据比较适用于 K-means 和 SOM 聚类算

法，对于其它传统聚类算法提升效果不明显，在 SOM 算法上，聚类准确率相比于平均向量提高了 7.79%，在 K-means 聚类算法上，聚类准确率提升了 5.16%，说明明句向量模型相比于平均向量确实可以有效提高聚类算法性能。在传统的聚类算法与本章模型相比下，可以观察到 DVEC 模型的聚类准确率是最高的，平均向量是 60.11%，句向量是 62.92%，相比于传统聚类准确率最高的 K-means 算法，在平均向量上提高了 5.72%，在句向量上提高了 3.37%，说明正则变分嵌入式聚类算法在学习特征表示的同时，其嵌入空间的向量可以提高聚类算法准确率，两者共同优化对算法性能具有很大的提升。同时，本章聚类算法模型也更加适用于软件需求文本数据。

本发明还将与比较流行的深度聚类算法 AE+K-means、DEC、IDEC 上进行对比，准确率对比如图 6 所示。

通过对比可以发现，AE+K-means 的准确率最低，其原因在于自编码器只是压缩数据特征，减小了维度，将压缩特征由 K-means 聚类算法完成，降低了算法复杂度，但是却未做其它方面的改进，与本章模型相比，平均向量上低 4.09%，句向量上低 3%。DEC 模型在平均向量基础上低于本章模型 2.03%，在句向量基础上低于 2.75%，本章模型相比于 DEC 并未去除解码器结构，维护特征局部结构，避免聚类损失影响特征空间。由于 IDEC 模型只是学习样本输入与输出的误差，未考虑向量的噪声和样本分布的问题，本文所提出的 DVEC 模型在输入端去除噪声，同时使用 VEC 学习样本分布和聚类划分。在 Average Embedding 上，DVEC 模型的聚类准确率略低于 IDEC，这是由于平均向量加权求和未能充分表达语义，造成准确率不增反降。在 Sentence Embedding 上，DVEC 的效果是最好的，准确率高达 62.92%，对比 Average Embedding 提高了 2.81%，对比 IDEC 提高了 1.14%，句向量考虑了词语前后位置关系，但未考虑噪声影响，DVEC 输入端融合 Dropout 正则化解决了这个问题，同时利用重参数技巧使嵌入空间根据正太分布来重构样本，适用于数据量比较少的情况，其编码解码过程类似于生成对抗的过程，该过程使嵌入空间的特征向量提取效果较好，对于提高聚类算法准确率有效。

在 SC、CH、DBI 上评价聚类结果的性能，如表 2 所示。

表 2 多种评价指标的对比

Method	Silhouette Coefficient	Calinski Harabasz	Davies Bouldin
AE+K-means	0.036	383.631	3.688
DEC	0.015	328.96	4.05
IDEC	0.024	328.66	3.97
DVEC	0.026	321.324	3.681

由表 2 可知，DVEC 的 Silhouette Coefficient 仅次于 AE+K-means，比 DEC 和 IDEC 的值都要高，说明了 DVEC 的类内距离比较小，而类间距离比较大，能较好的划分类别。在 Calinski Harabasz 评价指标上，DVEC 的值在对比模型中最小，这也就表明了本章算法模型在类别内部数据的协方差最小，类之间的协方差最大，聚类划分效果最优。在 Davies Bouldin 评价指标上，DVEC 模型的值同样也是最小的，值越小代表了不同类之间的相似度也最低，说明聚类划分比较清晰。

图 7 为本发明所提供的一种基于正则变分嵌入式的软件需求聚类系统结构示意图，如图 7 所示，本发明所提供的一种基于正则变分嵌入式的软件需求聚类系统，包括:

软件需求数据获取模块 701，用于获取不同类别软件的软件需求数据;

文本预处理模块 702，用于对所述软件需求数据进行文本预处理，确定软件需求文本;

句向量确定模块 703，用于利用 BERT 预训练的句向量模型将所述软件需求文件映射到向量空间，确定句向量;

聚类结果确定模块 704，用于利用正则变分嵌入式聚类模型对所述句向量进行聚类，确定聚类结果;

利用正则变分嵌入式聚类模型对所述句向量进行聚类的步骤为:

所述正则变分嵌入式聚类模型对所述句向量进行 Dropout 正则化处理，确定正则化向量;

利用全连接层对所述正则化向量进行特征压缩; 并根据所述压缩后的正则化向量，采用编码器确定嵌入特征;

利用解码器对所述嵌入特征进行解码，确定原始向量;

根据所述嵌入特征，采用 K-means 算法确定聚类划分结果;

根据所述嵌入特征、所述原始向量以及所述聚类划分结果确定相应的损失函数，并对所述嵌入特征、所述原始向量以及所述聚类划分结果相应的损失函数进行反向传播，确定所述聚类结果。

所述软件需求数据获取模块 701 具体包括:

软件需求数据获取单元，用于利用 Scrapy 技术获取 Softpedia 网站下下 Windows 平台的不同类别软件的软件需求数据;

软件需求数据存储标注单元，用于将每类软件需求数据以 csv 的格式单独存储，同时对每类软件需求数据进行标注。

所述文本预处理模块 702 具体包括:

文本剔除单元，用于利用所述正则表达式剔除所述软件需求数据中的 html 标签;

文本校正单元，用于将剔除 html 标签后的软件需求数据进行缩略词以及乱码单词的校正;

文本提取单元，用于对校正后的软件需求数据进行词干提取以及词形还原;

文本存储单元，用于将处理后的数据在 csv 文件中进行存储。

本发明所提供的一种基于正则变分嵌入式的软件需求聚类系统，还包括:

聚类评价模块，用于利用聚类指标对所述聚类结果进行评价。

本说明书中各个实施例采用递进的方式描述，每个实施例重点说明的都是与其他实施例的不同之处，各个实施例之间相同相似部分互相参见即可。对于实施例公开的系统而言，由于其与实施例公开的方法相对应，所以描述的比较简单，相关之处参见方法部分说明即可。

本文中应用了具体个例对本发明的原理及实施方式进行了阐述，以上实施例的说明只是用于帮助理解本发明的方法及其核心思想；同时，对于本领域的一般技术人员，依据本发明的思想，在具体实施方式及应用范围上均会有改变之处。综上所述，本说明书内容不应理解为对本发明的限制。

权　利　要　求　书

1、一种基于正则变分嵌入式的软件需求聚类方法，其特征在于，包括:

获取不同类别软件的软件需求数据;

对所述软件需求数据进行文本预处理，确定软件需求文本;

利用 BERT 预训练的句向量模型将所述软件需求文件映射到向量空间，确定句向量;

利用正则变分嵌入式聚类模型对所述句向量进行聚类，确定聚类结果;

利用正则变分嵌入式聚类模型对所述句向量进行聚类的步骤为:

所述正则变分嵌入式聚类模型对所述句向量进行 Dropout 正则化处理，确定正则化向量;

利用全连接层对所述正则化向量进行特征压缩; 并根据所述压缩后的正则化向量，采用编码器确定嵌入特征;

利用解码器对所述嵌入特征进行解码，确定原始向量;

根据所述嵌入特征，采用 K-means 算法确定聚类划分结果;

根据所述嵌入特征、所述原始向量以及所述聚类划分结果确定相应的损失函数，并对所述嵌入特征、所述原始向量以及所述聚类划分结果相应的损失函数进行反向传播，确定所述聚类结果。

2、根据权利要求 1 所述的一种基于正则变分嵌入式的软件需求聚类方法，其特征在于，所述获取不同类别软件的软件需求数据，具体包括:

利用 Scrapy 技术获取 Softpedia 网站下下 Windows 平台的 11 类软件的软件需求数据;

将每类软件需求数据以 csv 的格式单独存储，同时对每类软件需求数据进行标注。

3、根据权利要求 1 所述的一种基于正则变分嵌入式的软件需求聚类方法，其特征在于，所述对所述软件需求数据进行文本预处理，确定软件需求文本，具体包括:

利用所述正则表达式剔除所述软件需求数据中的 html 标签;

将剔除 html 标签后的软件需求数据进行缩略词以及乱码单词的校正;

对校正后的软件需求数据进行词干提取以及词形还原;

将处理后的数据在 csv 文件中进行存储。

4、根据权利要求 1 所述的一种基于正则变分嵌入式的软件需求聚类方法，其特征在于，所述利用全连接层对所述正则化向量进行特征压缩；并根据所述压缩后的正则化向量，采用编码器确定嵌入特征，具体包括:

利用公式$z = u + exp(\delta) * epsilon$确定嵌入特征;

其中，u和δ为编码器将压缩后的正则化向量转换为隐藏空间的两个参数，分别为均值和方差，$epsilon$为一个服从正态分布的张量，$epsilon \sim N(0,1)$。

5、根据权利要求 1 所述的一种基于正则变分嵌入式的软件需求聚类方法，其特征在于，所述利用正则变分嵌入式聚类模型对所述句向量进行聚类，确定聚类结果，之后还包括:

利用聚类指标对所述聚类结果进行评价。

6、一种基于正则变分嵌入式的软件需求聚类系统，其特征在于，包括:

软件需求数据获取模块，用于获取不同类别软件的软件需求数据;

文本预处理模块，用于对所述软件需求数据进行文本预处理，确定软件需求文本;

句向量确定模块，用于利用 BERT 预训练的句向量模型将所述软件需求文件映射到向量空间，确定句向量;

聚类结果确定模块，用于利用正则变分嵌入式聚类模型对所述句向量进行聚类，确定聚类结果;

利用正则变分嵌入式聚类模型对所述句向量进行聚类的步骤为:

所述正则变分嵌入式聚类模型对所述句向量进行 Dropout 正则化处理，确定正则化向量;

利用全连接层对所述正则化向量进行特征压缩；并根据所述压缩后的正则化向量，采用编码器确定嵌入特征;

利用解码器对所述嵌入特征进行解码，确定原始向量;

根据所述嵌入特征，采用 K-means 算法确定聚类划分结果;

根据所述嵌入特征、所述原始向量以及所述聚类划分结果确定相应的损失函数，并对所述嵌入特征、所述原始向量以及所述聚类划分结果相应的损失函数进行反向传播，确定所述聚类结果。

7、根据权利要求 6 所述的一种基于正则变分嵌入式的软件需求聚类系统，其特征在于，所述软件需求数据获取模块具体包括:

软件需求数据获取单元，用于利用 Scrapy 技术获取 Softpedia 网站下下 Windows 平台的不同类别软件的软件需求数据;

软件需求数据存储标注单元，用于将每类软件需求数据以 csv 的格式单独存储，同时对每类软件需求数据进行标注。

8、根据权利要求 6 所述的一种基于正则变分嵌入式的软件需求聚类系统，其特征在于，所述文本预处理模块具体包括:

文本剔除单元，用于利用所述正则表达式剔除所述软件需求数据中的 html 标签;

文本校正单元，用于将剔除 html 标签后的软件需求数据进行缩略词以及乱码单词的校正;

文本提取单元，用于对校正后的软件需求数据进行词干提取以及词形还原;

文本存储单元，用于将处理后的数据在 csv 文件中进行存储。

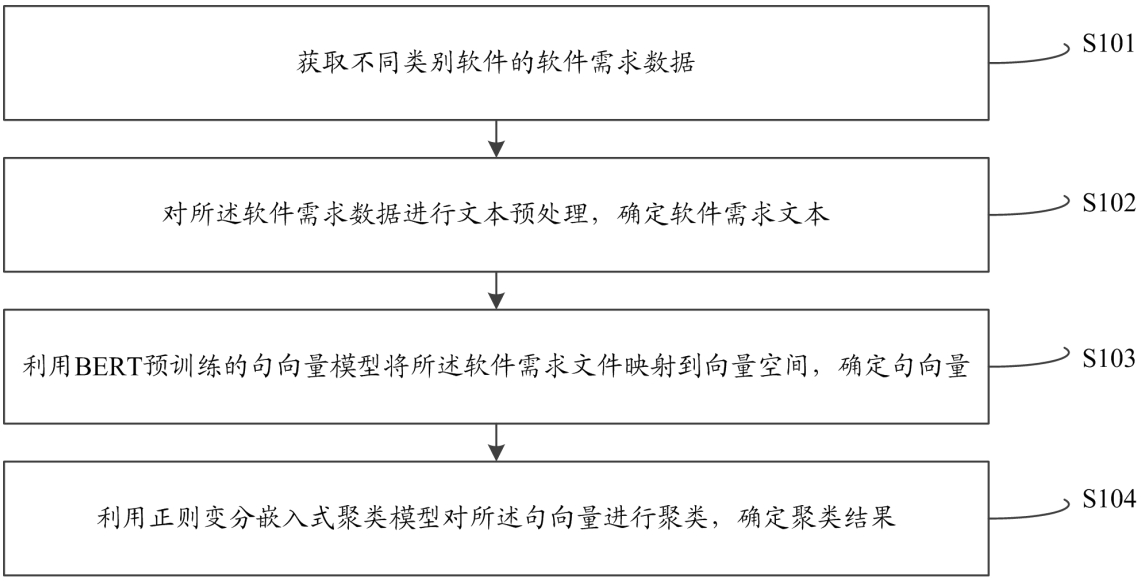
9、根据权利要求 6 所述的一种基于正则变分嵌入式的软件需求聚类系统，其特征在于，还包括:

聚类评价模块，用于利用聚类指标对所述聚类结果进行评价。

说　明　书　摘　要

本发明涉及一种基于正则变分嵌入式的软件需求聚类方法及系统。该方法包括：获取不同类别软件的软件需求数据；对软件需求数据进行文本预处理；利用 BERT 预训练的句向量模型将软件需求文件映射到向量空间；利用正则变分嵌入式聚类模型对句向量进行聚类；进行聚类的步骤为：对句向量进行 Dropout 正则化处理，确定正则化向量；利用全连接层对正则化向量进行特征压缩；并根据压缩后的正则化向量，采用编码器确定嵌入特征；利用解码器对嵌入特征进行解码；根据嵌入特征，采用 K-means 算法确定聚类划分结果；根据嵌入特征、原始向量以及聚类划分结果确定相应的损失函数，并对相应的损失函数进行反向传播，确定聚类结果。本发明提高了软件需求文本聚类的准确性。

摘　要　附　图


获取不同类别软件的软件需求数据
S101
对所述软件需求数据进行文本预处理，确定软件需求文本
S102
利用BERT预训练的句向量模型将所述软件需求文件映射到向量空间，确定句向量
S103
利用正则变分嵌入式聚类模型对所述句向量进行聚类，确定聚类结果
S104

说　明　书　附　图

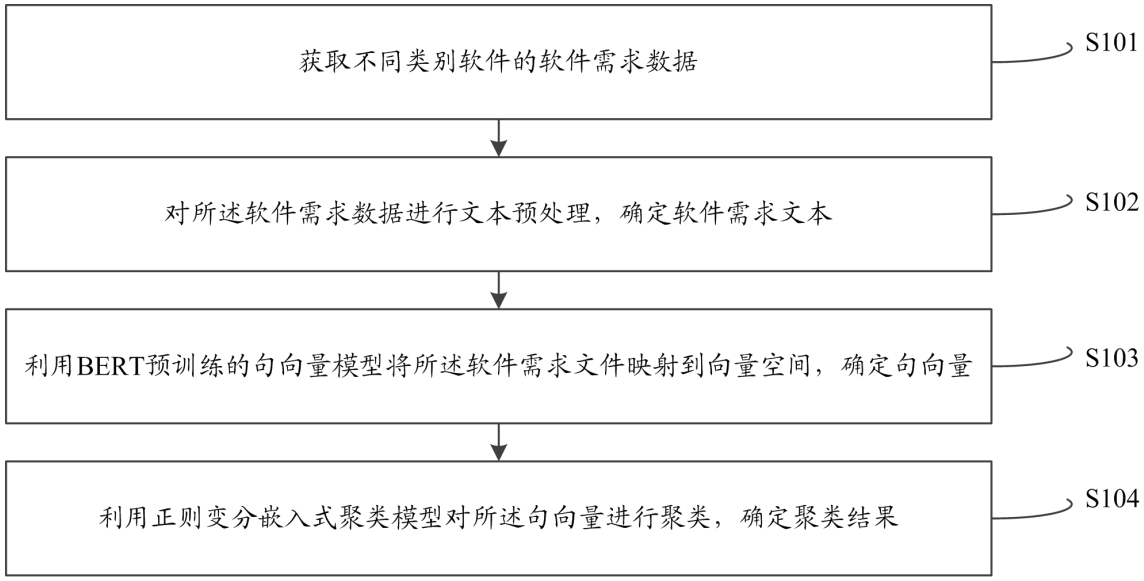

获取不同类别软件的软件需求数据
S101
对所述软件需求数据进行文本预处理，确定软件需求文本
S102
利用BERT预训练的句向量模型将所述软件需求文件映射到向量空间，确定句向量
S103
利用正则变分嵌入式聚类模型对所述句向量进行聚类，确定聚类结果
S104

图 1

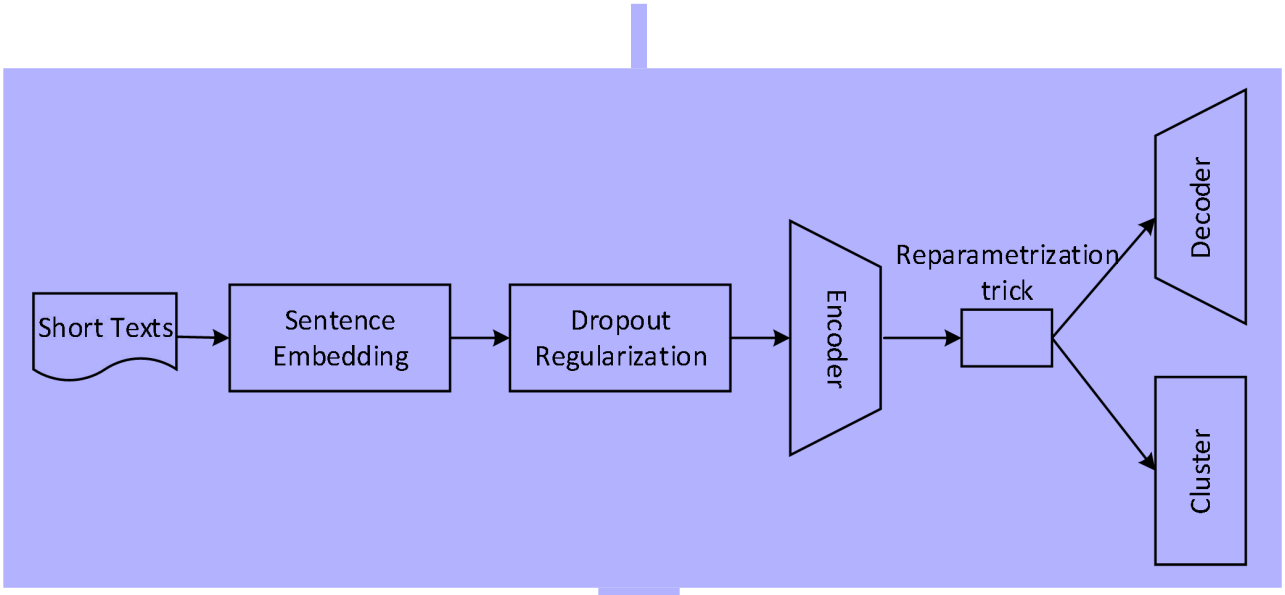

Short Texts
Sentence Embedding
Dropout Regularization
Encoder
Reparametrization trick
Decoder
Cluster

图 2

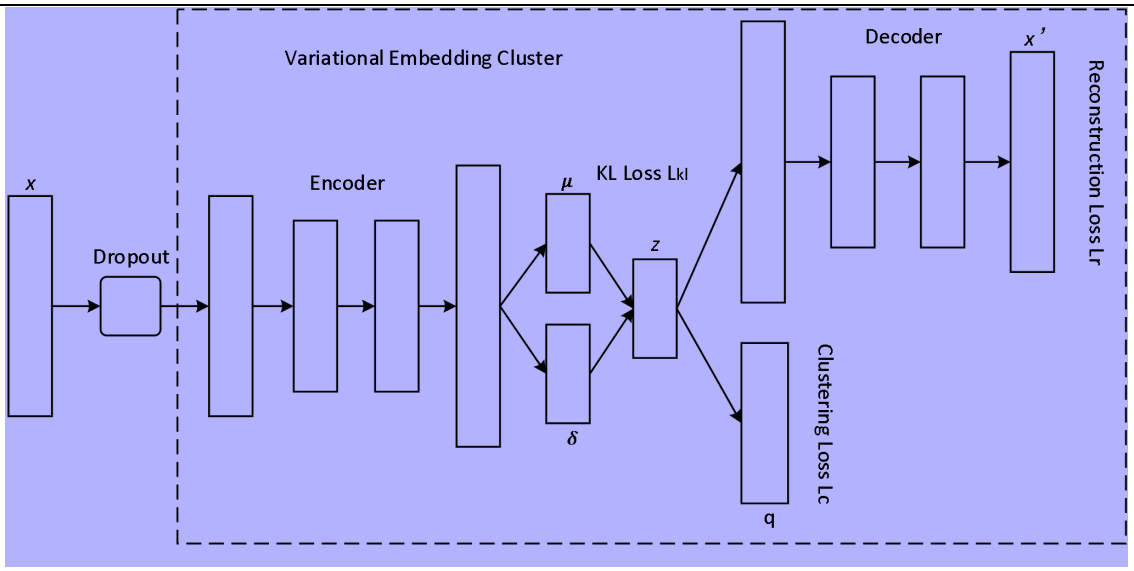

Variational Embedding Cluster
x
Dropout
Encoder
μ
δ
KL Loss Lkl
z
Decoder
x'
Reconstruction Loss Lr
q
Clustering Loss Lc

图 3

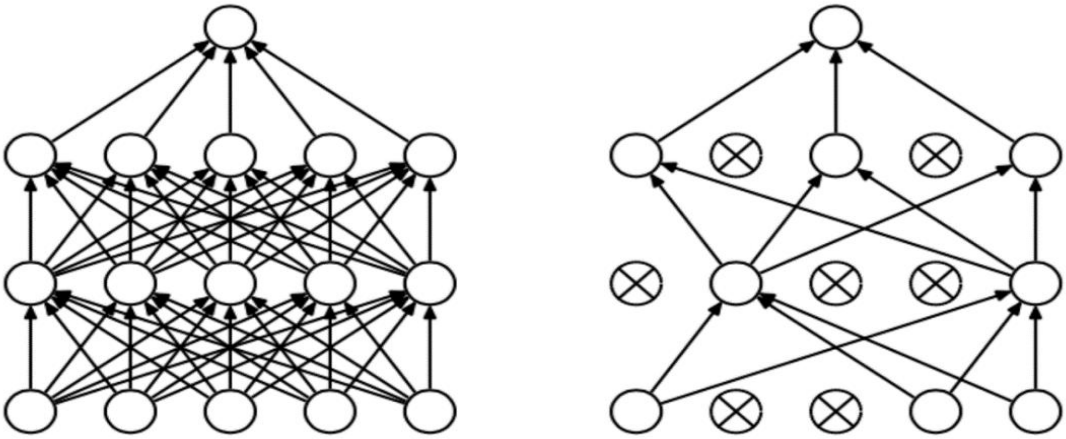


图 4

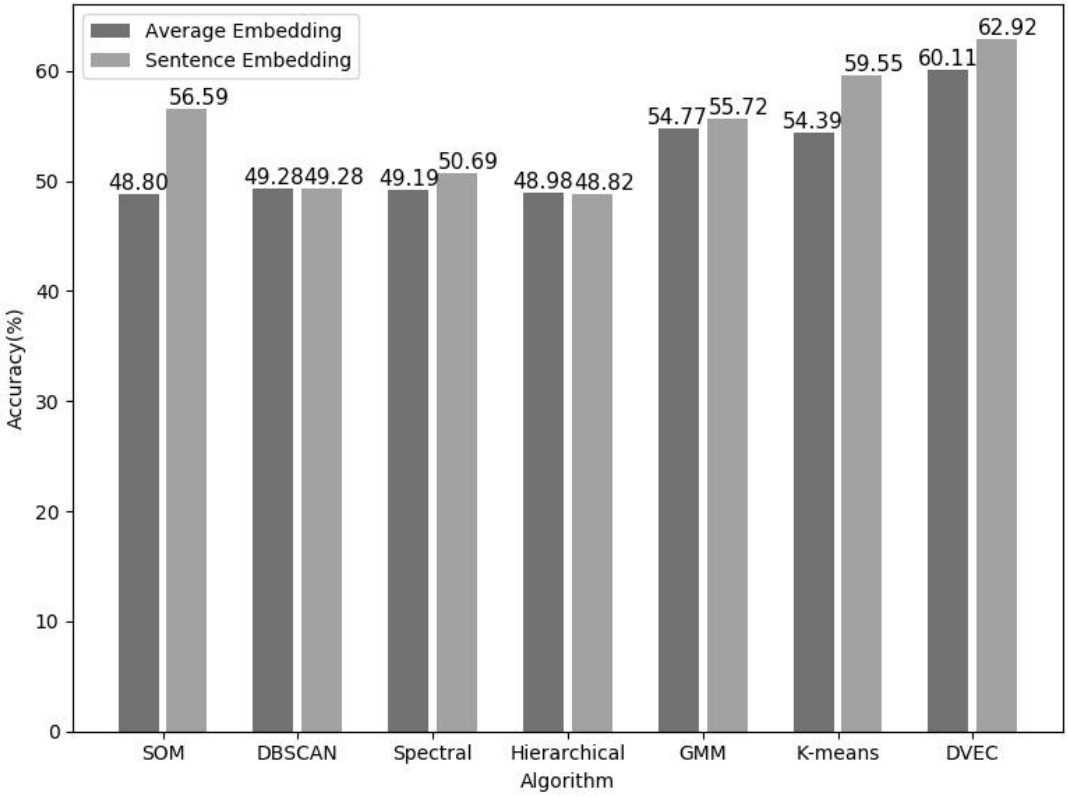

Average Embedding
Sentence Embedding
Accuracy(%)
0
10
20
30
40
50
60
48.80
56.59
49.28
49.28
49.19
50.69
48.98
48.82
54.77
55.72
54.39
59.55
60.11
62.92
SOM
DBSCAN
Spectral
Hierarchical
GMM
K-means
DVEC
Algorithm

图 5

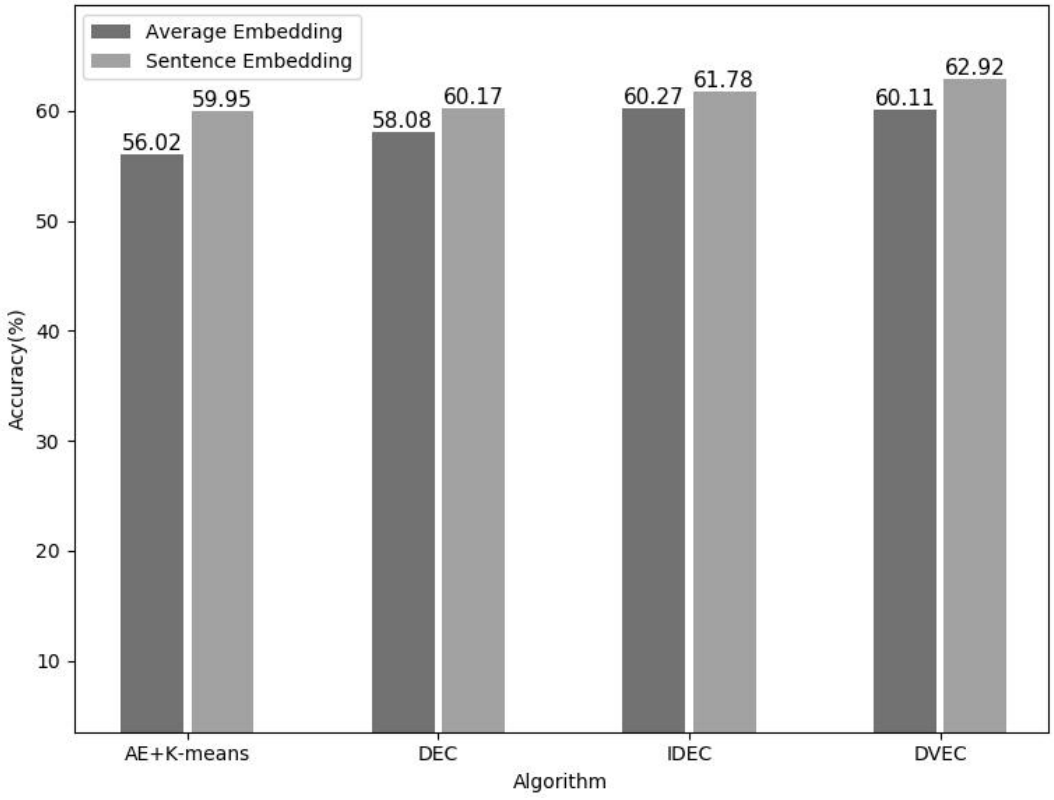

Average Embedding
Sentence Embedding
Accuracy(%)
10
20
30
40
50
60
56.02
59.95
58.08
60.17
60.27
61.78
60.11
62.92
AE+K-means
DEC
IDEC
DVEC
Algorithm

图 6

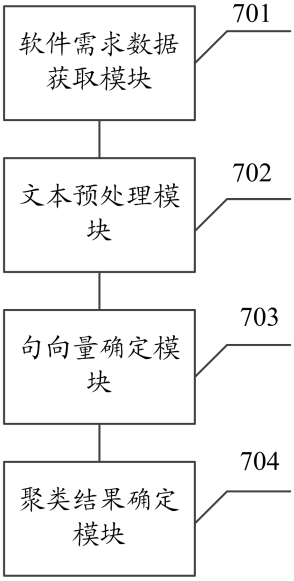


图 7