(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 110717015 B
(45)授权公告日 2021.03.26

(21)申请号 201910956103.8
(22)申请日 2019.10.10
(65)同一申请的已公布的文献号
申请公布号 CN 110717015 A
(43)申请公布日 2020.01.21
(73)专利权人 大连理工大学
地址 116024 辽宁省大连市甘井子区凌工路2号
(72)发明人 姚念民 郭顺
(74)专利代理机构 大连理工大学专利中心 21200
代理人 梅洪玉 刘秋彤
(51)Int.Cl.
G06F 16/33(2019.01)
G06F 16/35(2019.01)
G06F 16/36(2019.01)
审查员 邓慧丽

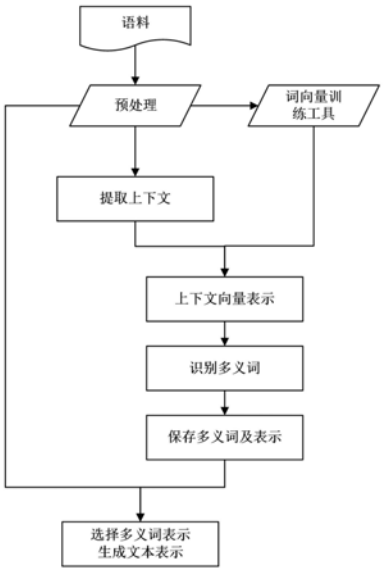
权利要求书2页 说明书5页 附图2页

(54)发明名称

一种基于神经网络的多义词识别方法

(57)摘要

本发明提供一种基于神经网络的多义词识别方法，属于数据挖掘和自然语言处理领域。该方法主要利用了文本中上下文的语义来识别多义词并生成多义词表示，包括五个步骤：1)预处理语料；2)预训练词表示；3)提取上下文；4)识别多义词；5)多义词表示的选择。本发明充分利用了词向量的优良特性，通过词的上下文语义差异来自动标识出多义词。同时，在具体的任务中，该发明也提供了通过多义词的上下文来选择多义词表示的方法，不仅提升了文本表示的质量，也提高了任务的准确率。此外，本发明的实施流程较为简便，具有良好的适用性。



CN 110717015 B

1.一种基于神经网络的多义词识别方法,其特征在于,包括以下步骤:

第一步,预处理语料

1.1)选择自然语言处理任务中的语料库,删除文本中的特殊字符和不可识别字符;

第二步,预训练词表示

2.1)对预处理后的语料使用词向量训练工具预训练词向量;所述的词向量训练工具包括word2vec、doc2vecC、以及基于它们的改进模型;

2.2)预训练结束后,保存词-词向量映射表;

第三步,提取上下文

3.1)定义一个新的上下文窗口,并重新扫描整个语料库,提取每个词在不同句子中的上下文;

3.2)统计每个词对应的上下文中的词,并删除重复的词,为每个词生成其对应的上下文词典;该词典的每一行记录的是一个词的上下文中出现的词的集合;

3.3)将步骤3.2)中的每个上下文词典与相应的词作映射,构建词-上下文词典映射表;

第四步,识别多义词

4.1)加载步骤3.3)得到的词-上下文词典映射表,对映射表中每个词对应的上下文分别进行k-means聚类,k≥2;聚类操作前,上下文中的词需要按照步骤2.2)得到的词-词向量映射表转换成相应的词向量形式;聚类操作后,得到上下文词典中每个词所属的类别,以及每一个类别的中心向量;

4.2)使用聚类评估算法对映射表中每个词的上下文的聚类结果进行评估;聚类评估算法需要以参与聚类的词表示和词所属的类别作为输入,输出为一个评估值;当一个词的上下文的评估结果大于预先定义的阈值,则判定该词为多义词;

4.3)输出多义词,并使用该多义词在步骤4.1)中得到的每个类别的中心向量作为不同词义的词表示;

第五步,多义词表示的选择

5.1)重新扫描语料库中的词,一旦目标词出现在多义词表中,就需要为该多义词选择符合当前上下文语义的词表示;

5.2)使用上下文窗口获取该多义词的上下文;

5.3)从步骤2.2)中的词-词向量映射表中获取该上下文中词的词向量,并计算他们的算数平均作为上下文向量;

5.4)分别计算该词的上下文向量和其不同词义的词表示之间的距离;

5.5)最终选择与该上下文向量距离最近的多义词向量作为该多义词在当前上下文中的词表示。

2.根据权利要求1所述的一种基于神经网络的多义词识别方法,其特征在于,步骤1.1)所述的语料库为与文本表示相关的任意语料库。

3.根据权利要求1或2所述的一种基于神经网络的多义词识别方法,其特征在于,步骤3.1)所述的新的上下文窗口与word2vec中的上下文窗口相同,用于定义提取上下文的范围;步骤3.1)定义的新的上下文窗口尺寸不能大于步骤2.1)中预训练词表示时所定义的窗口尺寸。

4.根据权利要求1或2所述的一种基于神经网络的多义词识别方法,其特征在于,步骤

4.2)所述的聚类评估算法包括轮廓系数、CH指标。

5.根据权利要求3所述的一种基于神经网络的多义词识别方法,其特征在于,步骤4.2)所述的聚类评估算法包括轮廓系数、CH指标。

6.根据权利要求1、2或5所述的一种基于神经网络的多义词识别方法,其特征在于,步骤5.2)所述的上下文窗口与步骤3.1)定义的上下文窗口保持一致;步骤5.4)所述的距离的度量方式采取欧氏距离或余弦距离。

7.根据权利要求3所述的一种基于神经网络的多义词识别方法,其特征在于,步骤5.2)所述的上下文窗口与步骤3.1)定义的上下文窗口保持一致;步骤5.4)所述的距离的度量方式采取欧氏距离或余弦距离。

一种基于神经网络的多义词识别方法

技术领域

[0001] 本发明属于数据挖掘和自然语言处理领域,特别涉及一种基于神经网络的多义词识别方法,具体可以应用在文本分类和情感分析等多项自然语言处理任务中。

背景技术

[0002] 在数据挖掘和自然语言处理领域,词表示是一项基础而又重要的工作。近年来,基于神经网络的方法来学习词的分布式表示备受关注。其中,著名的word2vec模型更是凭借着高效性和易用性脱颖而出。Word2vec的原理是使用目标词的上下文来训练目标词,并将意思相近的词映射成向量空间中相近的点。该模型已经在很多基于生成高质量的词表示的任务中取得了成功,例如语言建模、文本理解和机器翻译等。

[0003] 多义词识别是自然语言处理中一个热门的研究问题。多义词是指具有两个或更多意义的词,它们大多是一些和生活关系最密切的常用词,以动词与形容词居多。多义词在比拟、比喻、借代等修辞中,因其“多义”的特点,可以得到良好的表达效果。多义词识别任务就是让计算机能够自动的识别出给定的段落或句子中存在的多义词,并赋予该词更精确的词表示。多义词识别具有很重要的意义,它不仅能够提高词表示和段表示的质量,也能更准确地挖掘出句子所表达的情感,提高自然语言处理任务的准确率。

[0004] 目前,人们对多义词识别的研究较少。现有的方法只是盲目地将文本中的每一个词训练成多个词表示,并没有做到自动识别的目的。此外,这种方式不仅耗费了大量的训练时间,也占用了大量的存储资源。

发明内容

[0005] 本发明的目的是提供一种基于神经网络的多义词识别方法,该方法能够根据上下文的语义自动标识出文本中的多义词,并为每个多义词生成更贴近上下文语义的词表示,进而得到高质量的文本表示,提高自然语言处理任务的准确率。

[0006] 本发明的技术方案为:一种基于神经网络的多义词识别方法,包括以下步骤:

[0007] 第一步,预处理语料

[0008] 1.1)选择自然语言处理任务中的语料库,删除文本中的特殊字符和不可识别字符。

[0009] 第二步,预训练词表示

[0010] 2.1)对预处理后的语料使用词向量训练工具预训练词向量。我们可以选择word2vec中的CBOW模型,doc2vecC,以及基于它们的改进模型等多种模型。附图1给出了CBOW模型和doc2vecC模型的示意图。

[0011] 2.2)预训练结束后,保存词-词向量映射表。

[0012] 第三步,提取上下文

[0013] 3.1)定义一个新的上下文窗口,并重新扫描整个语料库,提取每个词在不同句子中的上下文。

[0014] 3.2)统计这些上下文中的词,并删除重复的词,为每个词生成其对应的上下文词典。该词典的每一行记录的是一个词的上下文中出现的词的集合。

[0015] 3.3)将步骤3.2)中的每个上下文词典与相应的词作映射,构建词-上下文词典映射表。

[0016] 第四步,识别多义词

[0017] 4.1)加载步骤3.3)得到的词-上下文词典映射表,对映射表中每个词对应的上下文分别进行k-means聚类(k≥2)。聚类操作前,上下文中的词需要按照步骤2.2)得到的词-词向量映射表转换成相应的词向量形式。聚类操作后,我们可以得到上下文词典中每个词所属的类别,以及每一个类别的中心向量。

[0018] 4.2)使用聚类评估算法对映射表中每个词的上下文的聚类结果进行评估。其中,聚类评估算法可以使用轮廓系数、CH指标等。聚类评估算法需要以参与聚类的词表示和词所属的类别作为输入,输出为一个评估值。如果一个词的上下文的评估结果大于预先定义的阈值,则判定该词为多义词。

[0019] 4.3)输出多义词,并使用该多义词在步骤4.1)中得到的每个类别的中心向量作为不同词义的词表示。

[0020] 第五步,多义词表示的选择

[0021] 以上步骤已经完成了多义词的识别,同时也得到了每个多义词不同词义的词表示。在具体的任务中,使用符合当前语义的多义词表示能够提升文本表示的质量,提高任务的准确率。接下来,我们介绍选择多义词表示的操作步骤:

[0022] 5.1)重新扫描语料库中的词,一旦目标词出现在多义词表中,我们就需要为该多义词选择符合当前上下文语义的词表示。

[0023] 5.2)使用上下文窗口获取该多义词的上下文。

[0024] 5.3)从步骤2.2)中的词-词向量映射表中获取该上下文中词的词向量,并计算他们的算数平均作为上下文向量。

[0025] 5.4)分别计算该词的上下文向量和其不同词义的词表示之间的距离。其中,距离的度量可以采取欧氏距离、余弦距离等多种方式。

[0026] 5.5)最终选择与该上下文向量距离最近的多义词向量作为该多义词在当前上下文中的词表示。附图2给出了本发明的技术方案图。

[0027] 本发明的有益效果为:充分利用了词向量的优良特性,通过词的上下文语义差异来标识出多义词,真正做到了自动识别。同时,在具体的任务中,该发明也提供了通过多义词的上下文来选择多义词表示的方法,不仅提升了文本表示的质量,也提高了任务的准确率。此外,本发明的实施流程较为简便,具有良好的适用性。

附图说明

[0028] 图1是CBOW模型和doc2vecC模型的示意图。其中,(a)表示CBOW的模型架构;(b)表示doc2vecC的模型架构。

[0029] 图2是多义词识别的技术方案图。

具体实施方式

[0030] 所述的具体实施例仅用于说明本发明的实现方式，而不限制本发明的范围。下面结合附图对本发明的实施方式进行详细说明。如附图2所示，总体的实施流程包括五个步骤，以下是针对每一步骤的详细说明：

[0031] 第一步，预处理语料

[0032] 1.1)选择自然语言处理任务中的语料库，记为D＝{D_1,...,D_n}，该语料库包含n个段落，D_i表示语料库D中的第i个段落。删除每个段落D_i中的特殊字符和不可识别字符后，得到不同长度的字符序列，记为$D_i=\{w_i^1,...,w_i^{T_i}\}$。

[0033] 第二步，预训练词表示

[0034] 2.1)对预处理后的语料使用词向量训练工具预训练词向量。我们可以选择word2vec中的CBOW模型、doc2vecC、以及基于它们的改进模型等多种模型。在这里我们以word2vec中的CBOW模型为例进行详细说明。

[0035] CBOW模型包含三个网络层：输入层、隐藏层和输出层。在输入层，模型定义了本地上下文窗口，记为c。该窗口表示取目标词前后各c个词，即上下文窗口的总词数为|2c|。模型以段落D_i为单位进行训练，并且以上下文窗口中的词表示作为输入。在隐藏层，模型对输入的上下文窗口的词向量求和，记为$\hat{v}$。最后，输出层预测出目标词属于词表中每一个词所对应的概率值。模型的训练过程就是不断地使用当前段落中上下文窗口的词预测目标词，并最大化模型的目标函数L，

[0036] $$L=\log P(w_i^t\mid Context(w_i^t))=\log(u_{w_i^t}\mid\hat{v})=\log\frac{\exp(u_{w_i^t}{}^{\mathrm{T}}\hat{v})}{\sum_{w'\in V}\exp(u_{w'}{}^{\mathrm{T}}\hat{v})} \quad (1)$$

[0037] 其中，w_i^t表示目标词是语料库中段落D_i中的第t个词，$u_{w_i^t}$表示w_i^t的词向量，V表示当前语料库的词典。

[0038] 在开始训练之前，需要对模型的参数进行设置。词表示的维度可以设置为100-1000之间，上下文窗口c的大小设置为2-10之间。因为我们提出的方法需要根据上下文的语义识别多义词，所以文本中需要保留精确的上下文信息。因此，我们保留所有的低频词。其他参数则使用默认值。

[0039] 2.2)预训练结束后，我们得到词-词向量映射表，记为$\{[w_1,u_{w_1}],...,[w_n,u_{w_n}]\}$。

[0040] 第三步，提取上下文

[0041] 3.1)定义一个新的上下文窗口，该窗口的尺寸不能大于步骤2.1)中设置的尺寸。例如，使用CBOW模型定义的上下文窗口的尺寸是5，那么此步骤设置的新上下文窗口的尺寸范围应是1-5之间。使用该上下文窗口重新扫描整个语料库，提取词-词向量映射表中每个词在不同句子中的上下文，生成集合$\{[w_1,[l_{w_1}^1,...,l_{w_1}^m]],...,[w_n,[l_{w_n}^1,...,l_{w_n}^k]]\}$。$l_{w_1}^m$和$l_{w_n}^k$分别表示词$w_1$的第m个上下文和词$w_n$的第k个上下文。此外，集合中不同词的上下文数目不一定相同。

[0042] 3.2)统计这些上下文中的词，并删除重复的词，为每个目标词生成上下文词典，记为$V_{w_t}=\{v_{w_t}^1,...,v_{w_t}^d\}$。即$w_t$的上下文中所有词的集合。

[0043] 3.3)构建词和其上下文词典的映射表，记为$P=\{[w_1,V_{w_1}],...,[w_n,V_{w_n}]\}$，这个映射表

的每一行记录的是词典中的一个词和在它的上下文中出现过的所有词的集合。

[0044] 第四步，识别多义词

[0045] 4.1)加载步骤3.3)得到的映射表，对映射表中每个词对应的上下文词典V_{w_t}分别进行k-means聚类。k表示聚类的类别数，并且不同的类别表示不同的上下文语义。我们以映射表中第一行$[w_1,V_{w_1}]\in P$为例，其中$V_{w_1}=[v_{w_1}^1,...,v_{w_1}^d]$。聚类操作前，$V_{w_1}$中的词需要按照步骤2.2)得到的词-词向量映射表转换成相应的词向量形式。公式(2)是k-means算法的目标函数，

[0046] $$\arg\min_{C_{w_1}}\sum_{i=1}^{k}\sum_{v_{w_1}^h\in C_{w_1}^i}\left\|u_{v_{w_1}^h}-\mu_{w_1}^i\right\|^2=\arg\min_{C_{w_1}}\sum_{i=1}^{k}\left|C_{w_1}^i\right|\operatorname{var}C_{w_1}^i \qquad (2)$$

[0047] 其中，$C_{w_1}=\{C_{w_1}^1,...,C_{w_1}^k\}$，并且$C_{w_1}^i$表示在词$w_1$的上下文中划分到第i类的词的集合，$u_{v_{w_1}^h}$表示$w_1$的上下文集合中第h个词的向量表示，$\mu_{w_1}^i$表示类别集合$C_{w_1}^i$中的中心向量。聚类操作后，我们可以得到每个类别的上下文词的集合$C_{w_1}=\{C_{w_1}^1,...,C_{w_1}^k\}$，以及相应的中心向量集合$\mu_{w_1}=\{\mu_{w_1}^1,...,\mu_{w_1}^k\}$。对映射表中所有的词进行相同的操作后，得到集合$C=\{C_{w_1},...,C_{w_n}\}$和集合$\mu=\{\mu_{w_1},...,\mu_{w_n}\}$。

[0048] 在具体实施中，聚类类别的设定范围为2-5之间。因为大多数的语料库中的多义词基本都是包含2个词义，很少出现更多的词义，因此，我们通常对所有词的上下文聚类类别设置2。

[0049] 4.2)使用聚类评估算法对映射表中每个词的上下文的聚类结果进行评估。其中，聚类评估算法可以使用轮廓系数、CH指标等。我们以轮廓系数为例，对词w_t的上下文的聚类结果进行评估，具体操作步骤如下：

[0050] 4.2.1)加载步骤4.1)得到的集合C，取出词w_t的上下文聚类结果集合，即C_{w_t}。

[0051] 4.2.2)计算集合中的每个词的轮廓系数。首先，取出集合中的单个词$v_{w_t}^h\in V_{w_t}$，计算该词到同类别其他词的平均距离，记为$a_{v_{w_t}^h}$，称为词$v_{w_t}^h$的簇内不相似度。$a_{v_{w_t}^h}$越小，说明该词越应该被聚类到该类。计算词$v_{w_t}^h$到其他类别的所有词的平均距离$b_{v_{w_t}^h}$，称为词$v_{w_t}^h$与其他类别的不相似度。$b_{v_{w_t}^h}$越大，说明该词越不属于其他类别。然后，根据词$v_{w_t}^h$的簇内不相似度$a_{v_{w_t}^h}$和簇间不相似度$b_{v_{w_t}^h}$，定义词$v_{w_t}^h$的轮廓系数为：

[0052] $$S_{v_{w_t}^h}=\frac{b_{v_{w_t}^h}-a_{v_{w_t}^h}}{\max(a_{v_{w_t}^h},b_{v_{w_t}^h})} \qquad (3)$$

[0053] 4.2.3)计算集合C_{w_t}的整体轮廓系数，

[0054] $$S(w_t)=\frac{S_{v_{w_t}^1}+,...,+S_{v_{w_t}^d}}{d} \qquad (4)$$

[0055] 其中，d表示w_t的上下文的集合中的总词数。轮廓系数的输出范围是[-1,1]，值越

高表示聚类的效果越好。对集合C中所有词的上下文聚类结果进行评估后，得到结果集合S＝{S(w1)，...，S(wn)}。

[0056] 4.2.4)定义多义词判别阈值α，将集合S中的每一个值与α进行比较，如果S(wt)>α，则判定该词wt为多义词。

[0057] 4.3)输出多义词，并使用该多义词在步骤4.1)中得到的每个类别的中心向量$\mu_{w_t}=\{\mu_{w_t}^1,...,\mu_{w_t}^k\}$作为不同词义的词表示。

[0058] 第五步，多义词表示的选择

[0059] 以上步骤已经完成了多义词的识别，同时也得到了每个多义词不同词义的词表示。在具体的任务中，使用符合当前语义的多义词表示能够提升文本表示的质量，提高任务的准确率。接下来，我们详细地介绍选择多义词表示的操作步骤：

[0060] 5.1)重新扫描语料库中的词，一旦目标词出现在多义词表中，我们就需要为该多义词选择符合当前上下文语义的词表示。设当前处理的句子为Dt，词wt∈Dt表示该段落中的多义词。

[0061] 5.2)使用步骤3.1)定义的上下文窗口获取该多义词wt的上下文，记为$l_{w_t}=[v_{w_t}^1,...,v_{w_t}^d]$。

[0062] 5.3)从步骤2.2)中的词-词向量映射表中获取该上下文中词的词向量，并计算他们的算数平均作为上下文向量，

[0063] $$u_{l_{w_t}}=\frac{u_{v_{w_t}^1}+,...,+u_{v_{w_t}^d}}{d} \qquad (5)$$

[0064] 5.4)分别计算$u_{l_{w_t}}$和wt不同词义的词表示$\mu_{w_t}=\{\mu_{w_t}^1,...,\mu_{w_t}^k\}$之间的距离。其中，距离的度量可以采取欧氏距离公式(6)、余弦距离公式(7)等多种方式。

[0065] $$d(u_{l_{w_t}},\mu_{w_t}^i)=\sqrt{(u_{l_{w_t}}-\mu_{w_t}^i)(u_{l_{w_t}}-\mu_{w_t}^i)^{\mathrm{T}}}(i=1,...,k) \qquad (6)$$

[0066] $$\cos(\theta)=\frac{u_{l_{w_t}}\bullet\mu_{w_t}^i}{\left|u_{l_{w_t}}\right|\left\|\mu_{w_t}^i\right|}(i=1,...,k) \qquad (7)$$

[0067] 5.5)最终从集合$\mu_{w_t}=\{\mu_{w_t}^1,...,\mu_{w_t}^k\}$中选择一个与wt上下文向量$u_{l_{w_t}}$距离最近的向量作为该多义词在当前上下文中的词表示。

[0068] 以上内容是结合具体的优选技术方案对本发明所作的进一步详细说明，不能认定本发明的具体实施只局限于这些说明。对于本发明所属技术领域的普通技术人员来说，在不脱离本发明构思的前提下，还可以做出若干简单推演或替换，都应当视为属于本发明的保护。

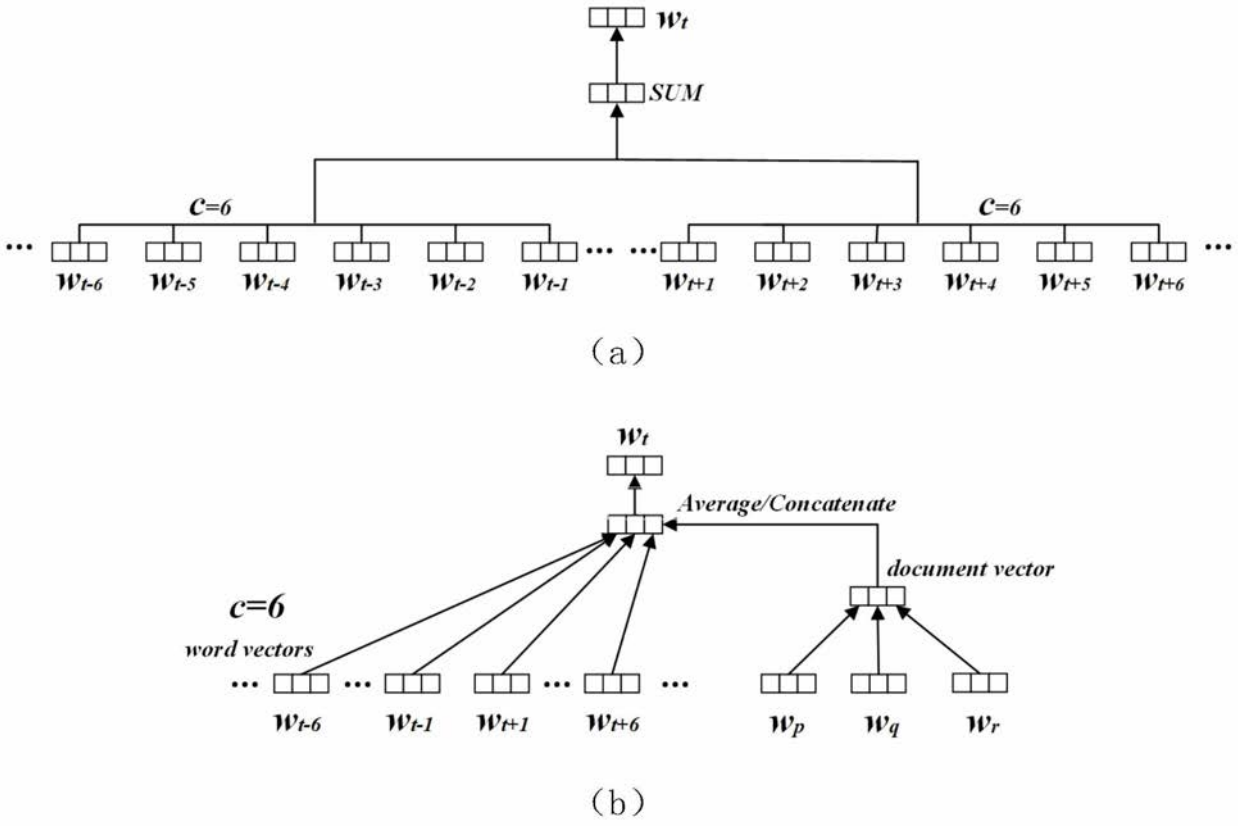

w_t
SUM
c=6
c=6
w_{t-6} w_{t-5} w_{t-4} w_{t-3} w_{t-2} w_{t-1} w_{t+1} w_{t+2} w_{t+3} w_{t+4} w_{t+5} w_{t+6}
(a)
w_t
Average/Concatenate
document vector
c=6
word vectors
w_{t-6} w_{t-1} w_{t+1} w_{t+6} w_p w_q w_r
(b)

图1

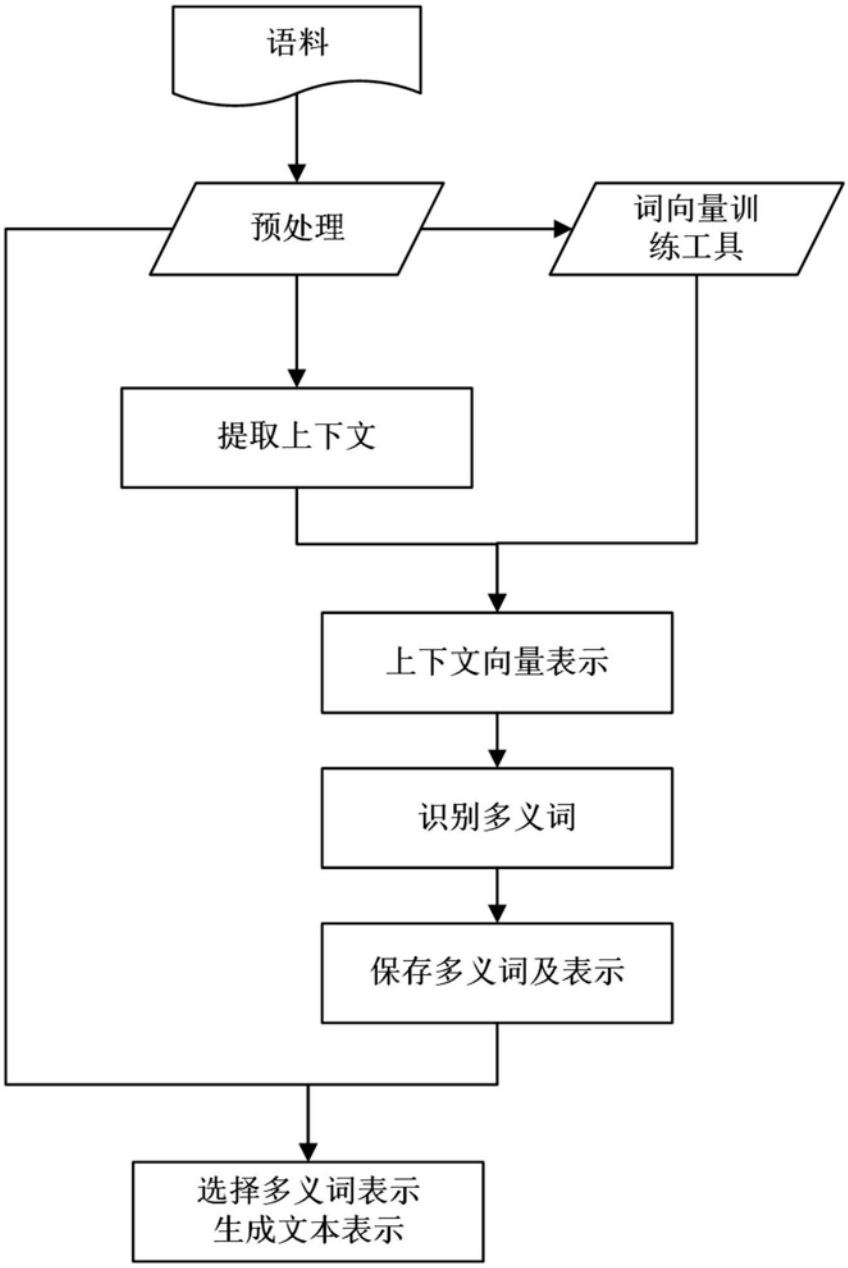

语料
预处理
词向量训练工具
提取上下文
上下文向量表示
识别多义词
保存多义词及表示
选择多义词表示生成文本表示

图2