(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 106980683 B
(45)授权公告日 2021.02.12

(21)申请号 201710204696.3

(22)申请日 2017.03.30

(65)同一申请的已公布的文献号
申请公布号 CN 106980683 A

(43)申请公布日 2017.07.25

(73)专利权人 中国科学技术大学苏州研究院
地址 215123 江苏省苏州市工业园区独墅湖高教区仁爱路166号

(72)发明人 杨威 周叶子 黄刘生

(74)专利代理机构 苏州创元专利商标事务所有限公司 32103
代理人 范晴 丁浩秋

(51)Int.Cl.
G06F 16/34(2019.01)
G06F 16/33(2019.01)
G06F 16/35(2019.01)
G06N 3/08(2006.01)

(56)对比文件
CN 105930314 A,2016.09.07
CN 103646094 A,2014.03.19
US 8036415 B2,2011.10.11
Alexander M. Rush.A Neural Attention Model for Abstractive Sentence Summarization.《URL:https://arxiv.org/abs/1509.00685》.2015,
Baotian Hu.LCSTS: A Large Scale Chinese Short Text Summarization Dataset.《URL:https://arxiv.org/abs /1506.05865》.2016,

审查员 徐晓孜

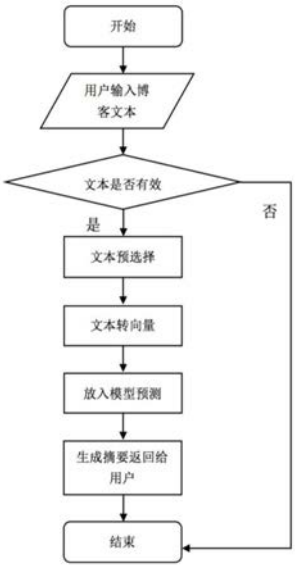
权利要求书2页 说明书8页 附图3页

(54)发明名称

基于深度学习的博客文本摘要生成方法

(57)摘要

本发明公开了一种基于深度学习的博客文本摘要生成方法,包括以下步骤:爬取博客数据;对爬取的博客数据进行预处理,选取博客文本数据;将选取的博客文本数据根据中文词向量词典转换成向量矩阵数据;构建深度学习encoder-decoder(编码器-解码器)模型,并对该模型的encoder编码器和decoder解码器分开训练,训练完成后连接使用;重复步骤S01-S03得到生成数据,将生成数据通过训练完成的模型生成预测摘要。本发明基于深度学习框架encoder-decoder自动生成博客的文本摘要,同时可以获取博客更深层次的语义联系。生成的文本摘要可以直观的显示当前博客的主要内容,具有广泛的应用前景。



CN 106980683 B

1.一种基于深度学习的博客文本摘要生成方法,其特征在于,包括以下步骤:

S01:爬取博客数据;

S02:对爬取的博客数据进行预处理,选取博客文本数据;

S03:将选取的博客文本数据根据中文词向量词典转换成向量矩阵数据;

S04:构建深度学习encoder-decoder(编码器-解码器)模型,并对该模型的encoder编码器和decoder解码器分开训练,训练完成后连接使用;

S05:重复步骤S01-S03得到生成数据,将生成数据通过训练完成的模型生成预测摘要;

所述步骤S04具体包括:

S41:训练模型encoder编码器中的卷积神经网络,将向量矩阵数据转换成句向量,将训练数据与卷积神经网络中的卷积核相互运算,运算公式如下:

$$f_{ij} = \tanh(w_{j:j+c-1} \otimes K + b)$$

其中,f_{ij}表示第i个神经网络的第j个元素,K表示该卷积神经网络的卷积核,$W_{j:j+c-1}$表示网络输入选取第j到j+c-1行,b表示偏置量;

从当前每个神经网络中选取最大值$s_{iK} = \max_j f_{ij}$,将所有的最大值连接组成句向量,s_{iK}表示第i个神经网络在K这个卷积核的作用经过最大池化最终的值;

S42:训练模型encoder编码器中的递归神经网络,将生成的句向量转换成文本向量,计算公式如下:

$$a_h^t = \sum_j w_{ih} s_i^t + \sum_{h'} w_{h'h} b_{h'}^{t-1}$$

$$b_h^t = \tanh(a_h^t)$$

$$a_k^t = \sum_h w_{hk} b_h^t$$

$$y_k^t = \frac{e^{a_k^t}}{\sum_j e^{a_j^t}}$$

其中,s_i^t表示t时刻递归神经网络的输入,b_h^t表示t时刻递归神经网络的隐藏层的输出状态,w_{ih}表示输入层和隐藏层的权值矩阵i*h,$w_{h'h}$表示上一时刻隐藏层与当前时刻隐藏层的权值矩阵h'*h,a_h^t表示递归神经网络中t时刻隐藏层第h个神经元的中间值,tanh表示隐藏层激活函数是双曲正切函数,w_{hk}表示递归神经网络中隐藏层和输出层的权值矩阵,a_k^t表示递归神经网络中t时刻输出层第k个神经元的中间值,e^x表示输出层激活函数是softmax的指数函数形式,y_k^t表示最终输出层的输出;将最后序列生成的b_h^t传递给解码器;

S43:训练模型decoder解码器中的长短期记忆网络LSTM,将编码器中递归神经网络生成的隐藏状态作为输入,在LSTM中结合上一时刻隐藏层的状态和当前时刻的输入决定当前时刻隐藏层的状态h_t,通过输出层得到预测摘要,计算公式如下:

$f_t=\sigma(W_f \cdot [h_{t-1},x_t]+b_f)$

$i_t=\sigma(W_i \cdot [h_{t-1},x_t]+b_i)$

$C'_t=\tanh(W_C \cdot [h_{t-1},x_t]+b_C)$

$C_t=f_t*C_{t-1}+i_t*C'_t$

$o_t=\sigma(W_o \cdot [h_{t-1},x_t]+b_o)$

$h_t=o_t*\tanh(C_t)$

其中，C_t表示t时刻当前LSTM中的状态，C'_t表示t时刻LSTM中神经元新的状态候选值，f_t表示t时刻LSTM中忘记门层的输出，i_t表示t时刻LSTM中输入门层的输出，o_t表示t时刻输出层的输出，h_t表示t时刻当前网络隐藏层状态，x_t表示t时刻网络的输入，即摘要训练数据的向量，b_f表示忘记门层的偏置值，b_i表示输入门层的偏置值，b_c表示神经元新旧状态之间的偏置值，b_o表示输出层的偏置值，σ表示激活函数sigmoid，w_f表示忘记门层与输入层的权值矩阵，w_i表示输入门层与输入层的权值矩阵，w_c表示神经元新旧状态的权值矩阵，w_o表示输出层的权值矩阵，tanh表示激活函数双曲正切函数；上述公式表示在LSTM中结合上一时刻隐藏层的状态和当前时刻的输入决定当前时刻隐藏层的状态，得到h_t后，通过同递归神经网络相似的输出层softmax得到预测摘要，softmax的输出层是300维大小词向量。

2.根据权利要求1所述的基于深度学习的博客文本摘要生成方法，其特征在于，所述步骤S01包括：

S11：爬取csdn的多个专家博客，多个主题；

S12：选取专家博客网页标签中的摘要部分作为实际摘要，如果该博客没有摘要，则将专家博客的标题以及通过传统文本摘要生成算法选取的权值最大语句联合作为该博客实际摘要，用于在训练时使用。

3.根据权利要求1所述的基于深度学习的博客文本摘要生成方法，其特征在于，所述步骤S02具体包括以下步骤：

S21：滤除博客数据中的视频元素、图片元素、数学计算公式元素，只保留博客数据的文本部分；

S22：将博客文本数据分段，提取分段文本数据的第一段，随机选择其余文本数据的任意一段，组成初始文本数据；

S23：对初始文本数据进行分句，设定每一分句的词数A；

S24：如果每一分句的词数超过A，根据词频大小选择词频最高的A个词，按照原先的顺序连接起来；如果词数少于A，使用0向量填充，对初始文本数据句对齐。

4.根据权利要求3所述的基于深度学习的博客文本摘要生成方法，其特征在于，所述步骤S03中，在中文词向量词典中没有查询到的词使用近似词替换。

5.根据权利要求1所述的基于深度学习的博客文本摘要生成方法，其特征在于，对训练完成的模型进行评估，具体包括：

步骤一、采用ROUGE指标，通过比较预测摘要和实际摘要的重合程度进行评估；

步骤二、使用博客数据进行训练，使用DUC-200数据集用于模型测评；

步骤三、将该模型与当前已存在的其他摘要生成模型对比。

基于深度学习的博客文本摘要生成方法

技术领域

[0001] 本发明涉及一种文本摘要生成方法，具体地涉及一种基于深度学习的博客文本摘要生成方法。

背景技术

[0002] 自然语言处理(Natural Language Processing)是当前人工智能特别重要的一部分，它包括文本分类、情感分析、机器翻译、阅读理解等多个子任务，几乎一个子任务就是一个相当重要的专业研究领域，它们之间相互独立又相互联系。

[0003] 深度学习是在近年来提出的一种新型的端到端的学习方式，在普通的处理任务中比如分类也许与普通神经网络的效果相差无几，但是在高维数据的计算以及特征提取的过程中深度学习使用深度网络来拟合，显示了其强大的计算能力。目前深度学习已经运用到了多个领域--图像处理、音频处理、视频处理、自然语言处理，自从2006年由hinton提出以来，它使得众多智能摆脱了繁琐复杂的特征工程过程，比如数据预处理标注等，通过选择不同的模型组合直接由输入数据经过训练便可得到想要的输出形式。

[0004] 将深度学习运用到自然语言处理上的想法由来已久，但是从英文和中文的语言差别性我们可以看到目前深度学习在中文语言上的处理还不尽如意。2014年，”encoder-decoder”深度学习的机器翻译框架被提出，使得深度学习在机器翻译、摘要生成、阅读理解方面有了很大的突破，获得更深层次的文本语义联系。

[0005] 自然语言中文本摘要生成方式主要分成两个方式：第一抽取型，基于规则和统计的文摘要生成，目前已有大量的运用实践证明；第二是抽象型，基于深度学习模型的摘要生成，2014年得到巨大改进，从机械型文本摘要生成跨向理解型文本摘要生成，当前使用encoder-decoder框架，嵌入递归神经网络来实现，在中文方面运用还不明显。

[0006] 随着互联网影响力的扩大，人们使用互联网来相互交流学习愈加频繁，从海量的互联网数据中迅速获取我们所需要的信息，解决信息过载是当前重要的自然语言任务之一，特别是针对于博客一类的数据更是重要，博客往往属于中长型的文本，所表达的信息包含了专业、娱乐、生活等方面，在专业方面的博客往往被大量浏览学习收藏。在信息快速更替的时代，为了方便用户有效率地浏览相应博客，可以快速地获取博客摘要主要内容是必须的。

发明内容

[0007] 针对上述存在的技术问题，本发明目的是：提供了一种基于深度学习的博客文本摘要生成方法，基于深度学习框架encoder-decoder(编码器-解码器)自动生成博客的文本摘要，同时可以获取博客更深层次的语义联系。生成的文本摘要可以直观的显示当前博客的主要内容，具有广泛的应用前景。

[0008] 本发明的技术方案是：

[0009] 一种基于深度学习的博客文本摘要生成方法，包括以下步骤：

[0010] S01:爬取博客数据;

[0011] S02:对爬取的博客数据进行预处理,选取博客文本数据;

[0012] S03:将选取的博客文本数据根据中文词向量词典转换成向量矩阵数据;

[0013] S04:构建深度学习encoder-decoder(编码器-解码器)模型,并对该模型的encoder编码器和decoder解码器分开训练,训练完成后连接使用;

[0014] S05:重复步骤S01-S03得到生成数据,将生成数据通过训练完成的模型生成预测摘要。

[0015] 优选的,所述步骤S01包括:

[0016] S11:爬取csdn的多个专家博客,多个主题;

[0017] S12:选取专家博客网页标签中的摘要部分作为实际摘要,如果该博客没有摘要,则将专家博客的标题以及通过传统文本摘要生成算法选取的权值最大语句联合作为该博客实际摘要,用于在训练时使用。

[0018] 优选的,所述步骤S02具体包括以下步骤:

[0019] S21:滤除博客数据中的视频元素、图片元素、数学计算公式元素,只保留博客数据的文本部分;

[0020] S22:将博客文本数据分段,提取分段文本数据的第一段,随机选择其余文本数据的任意一段,组成初始文本数据;

[0021] S23:对初始文本数据进行分句,设定每一分句的词数A;

[0022] S24:如果每一分句的词数超过A,根据词频大小选择词频最高的A个词,按照原先的顺序连接起来;如果词数少于A,使用0向量填充,对初始文本数据句对齐。

[0023] 优选的,所述步骤S03中,在中文词向量词典中没有查询到的词使用近似词替换。

[0024] 优选的,所述步骤S04具体包括:

[0025] S41:训练模型encoder编码器中的卷积神经网络,将向量矩阵数据转换成句向量,将训练数据与卷积神经网络中的卷积核相互运算,运算公式如下:

[0026] $$f_{ij} = \tanh(w_{j:j+c-1} \otimes K + b)$$

[0027] 其中,f_{ij}表示第i个神经网络的第j个元素,K表示该卷积神经网络的卷积核,$w_{j:j+c-1}$表示网络输入选取第j到j+c-1行,b表示偏置量;

[0028] 从当前每个神经网络中选取最大值$s_{iK} = \max_j f_{ij}$,将所有的最大值连接组成句向量,s_{iK}表示第i个神经网络在K这个卷积核的作用经过最大池化最终的值;

[0029] S42:训练模型encoder编码器中的递归神经网络,将生成的句向量转换成文本向量,计算公式如下:

[0030] $$a_h^t = \sum_j w_{ih} s_i^t + \sum_{h'} w_{h'h} b_{h'}^{t-1}$$

[0031] $$b_h^t = \tanh(a_h^t)$$

[0032] $$a_k^t = \sum_h w_{hk} b_h^t$$

[0033] $$y_k^t = \frac{e^{a_k^t}}{\sum_j e^{a_j^t}}$$

[0034] 其中，s_i^t表示t时刻递归神经网络的输入，b_h^t表示t时刻递归神经网络的隐藏层的输出状态，w_{ih}表示输入层和隐藏层的权值矩阵i＊h，$w_{h'h}$表示上一时刻隐藏层与当前时刻隐藏层的权值矩阵h‘＊h，a_h^t表示递归神经网络中t时刻隐藏层第h个神经元的中间值，tanh表示隐藏层激活函数是双曲正切函数，w_{hk}表示递归神经网络中隐藏层和输出层的权值矩阵，a_k^t表示递归神经网络中t时刻输出层第k个神经元的中间值，e^x表示输出层激活函数是softmax的指数函数形式，y_k^t表示最终输出层的输出；将最后序列生成的b_h^t传递给解码器；

[0035] S43：训练模型decoder解码器中的长短期记忆网络LSTM，将编码器中递归神经网络生成的隐藏状态作为输入，在LSTM中结合上一时刻隐藏层的状态和当前时刻的输入决定当前时刻隐藏层的状态h_t，通过输出层得到预测摘要，计算公式如下：

[0036] $f_t=\sigma(W_f \cdot [h_{t-1},x_t]+b_f)$

[0037] $i_t=\sigma(W_i \cdot [h_{t-1},x_t]+b_i)$

[0038] $C'_t=\tanh(W_C \cdot [h_{t-1},x_t]+b_C)$

[0039] $C_t=f_t*C_{t-1}+i_t*C'_t$

[0040] $o_t=\sigma(W_o \cdot [h_{t-1},x_t]+b_o)$

[0041] $h_t=o_t*\tanh(C_t)$

[0042] 其中，C_t表示t时刻当前LSTM中的状态，C'_t表示t时刻LSTM中神经元新的状态候选值，f_t表示t时刻LSTM中忘记门层的输出，i_t表示t时刻LSTM中输入门层的输出，o_t表示t时刻输出层的输出，h_t表示t时刻当前网络隐藏层状态，x_t表示t时刻网络的输入，即摘要训练数据的向量，b_f表示忘记门层的偏置值，b_i表示输入门层的偏置值，b_c表示神经元新旧状态之间的偏置值，bo表示输出层的偏置值，σ表示激活函数sigmoid，w_f表示忘记门层与输入层的权值矩阵，w_i表示输入门层与输入层的权值矩阵，w_c表示神经元新旧状态的权值矩阵，w_o表示输出层的权值矩阵，tanh表示激活函数双曲正切函数；公式3-1到公式3-6表示在LSTM中结合上一时刻隐藏层的状态和当前时刻的输入决定当前时刻隐藏层的状态，得到h_t之后，会通过同递归神经网络相似的输出层softmax得到预测摘要，softmax的输出层是300维大小同词向量。

[0043] 优选的，对训练完成的模型进行评估，具体包括：

[0044] 步骤一、采用ROUGE指标，通过比较预测摘要和实际摘要的重合程度进行评估；

[0045] 步骤二、使用博客数据进行训练，使用DUC-200数据集用于模型测评；

[0046] 步骤三、将该模型与当前已存在的其他摘要生成模型对比。

[0047] 与现有技术相比，本发明的优点是：

[0048] (1)利用深度学习技术生成文本摘要，可以直观有效的了解博客文本的主要内容，同时此技术可以扩展向其他类型文本的摘要生成或者文本总结领域，在中英文语料均可，具有广泛的应用前景。

[0049] (2)通过深度学习模型自动生成摘要,研究了语义更深层的联系,建立了完善的语言模型,生成的多种语言副产品包括句向量、文本向量,可以用于语言情感分析以及文本分类等语言任务中。

[0050] (3)与基于统计与规则的摘要生成方式相比,更佳端到端,省略了以往自然语言处理中繁琐的流程,比如分词、标注等。

[0051] (4)使用深度学习机器翻译框架,可使得运用领域扩展至其他,比如阅读理解、故事生成等。

附图说明

[0052] 下面结合附图及实施例对本发明作进一步描述:

[0053] 图1为本发明用户使用的整体流程图;

[0054] 图2为本发明文本预选择方法的流程图;

[0055] 图3为本发明博客数据生成词典的流程图;

[0056] 图4为本发明文本到向量转换的流程图;

[0057] 图5为本发明基于深度学习的摘要生成模型训练的流程图。

具体实施方式

[0058] 以下结合具体实施例对上述方案做进一步说明。应理解,这些实施例是用于说明本发明而不限于限制本发明的范围。实施例中采用的实施条件可以根据具体厂家的条件做进一步调整,未注明的实施条件通常为常规实验中的条件。

[0059] 实施例:

[0060] 一种基于深度学习的中文博客摘要生成方法,具体步骤包含:

[0061] 步骤一、博客训练数据爬取和整理

[0062] 博客训练数据爬取自csdn网站的人气博客,得到的博客内容多样,但都是专业性较强的文本,同时博客训练数据中也有些数据存在缺陷,比如博客过于短小,博客中没有文本,只包含了视频和图片,对于这种文本我们会丢弃。

[0063] 使用beautifulsoup中的find和get_text得到最终的博客文本并且选取网页标签类别为article_description的文本内容作为博客实际摘要。如果该博客没有摘要,则将专家博客的标题以及通过textRank选取的权值最大语句联合作为该博客实际摘要,在训练时使用。

[0064] textRank方法是一种基于统计和规则的文本摘要生成算法,用于通过权值大小提取关键字和关键句,目前被封装在多种语言平台包括java、python、c++的类库中,可以直接调用。

[0065] 步骤二、文本预选择及文本到向量转换

[0066] 1)将博客文本训练数据,通过‘\n’标识分段;

[0067] 2)选取博客数据的首段,通过多篇论文得出的结论即一篇文章大多时候会在开头和结尾体现出要表达的主要思想,此外再结合通过random函数随机选取的其他任意一段,作为最终训练的博客文本数据,其中这里处理的是训练数据中博客文本部分,训练数据中的博客摘要部分不需要选择;

[0068] 3)将初步选择的博客文本摘要数据,以‘,’和‘。’为标识分句,使用nltk工具进行分词,并且统计各词词频,词频的统计是在全文中进行的;将每一句的词量控制在20词(词数还可以为其他值)以内,如果超过20词即通过词频大小选择出该剧中词频最高的20个词,按照顺序连接起来,组成句子代替原来的句子;如果该句包含词语少于20,即使用0来代替padding来完成对初步选择的博客文本数据的句对齐;

[0069] 4)从已经完成句对齐的文本当中,随机选择10个句子,来表示成我们最终将放入学习模型的训练数据;

[0070] 5)使用word2vec对收集的博客训练数据生成词向量词典,生成的词向量为300维,训练参数设置如表1;

[0071]

Cbow	Size	Window	Negative	Binary	Iter
0	300	5	0	1	15

[0072] 6)对已经整理好的文均200词的博客摘要数据进行文本到向量的转换,遍历文中各词在生成的词典中进行查找,将查找到的词向量按照原来文本的顺序连接起来,即每篇博客数据的句子用20*300的矩阵表示,最终会有10个这样的矩阵。

[0073] 步骤三、基于深度学习的摘要生成模型训练

[0074] 该步骤关键在于模型的构建以及训练,深度学习模型有多层网络,这里使用encoder-decoder(编码器解码器)框架,在编码器中嵌入卷积神经网络CNN和递归神经网络RNN对初始文本进行编码,在解码器中嵌入长短期记忆神经网络LSTM对训练数据进行预测。

[0075] 训练模型encoder编码器中的卷积神经网络,将文本选择生成的向量数据转换成句向量,其中的卷积神经网络featuremap大小为300,卷积核为(3,300),池化方式为max-pooling即最大池化方式,相关公式如下:

[0076] $$f_{ij} = \tanh(w_{j:j+c-1} \otimes K + b) \quad 1\text{-}1$$

[0077] $$s_{iK} = \max_j f_{ij} \quad 1\text{-}2$$

[0078] 公式1-1表示训练数据与卷积神经网络中的卷积核相互运算,f_{ij}表示第i个feature map的第j个元素,K表示该卷积神经网络的卷积核,这里卷积核的大小是3*300,$W_{j:j+c-1}$表示网络输入选取第j到j+c-1行,这里的c值为3,b表示偏置量;公式1-2是经过从当前每个feature map中选取最大值,最终300个最大值连接组成句向量,s_{iK}表示第i个feature map在K这个卷积核的作用经过最大池化最终的值。

[0079] 步骤二、训练模型encoder编码器中的递归神经网络,将生成的300维句向量转换成文本向量,相关公式如下:

[0080] $$a_h^t = \sum_j w_{ih} s_i^t + \sum_{h'} w_{h'h} b_{h'}^{t-1} \quad 2\text{-}1$$

[0081] $$b_h^t = \tanh(a_h^t) \quad 2\text{-}2$$

[0082] $$a_k^t = \sum_h w_{hk} b_h^t \quad 2\text{-}3$$

[0083] $$y_k^t = \frac{e^{a_k^t}}{\sum_j e^{a_j^t}} \qquad 2\text{-}4$$

[0084] 在上述公式中，s_i^t表示t时刻递归神经网络的输入，b_h^t表示t时刻递归神经网络的隐藏层的输出状态，w_{ih}表示输入层和隐藏层的权值矩阵i＊h，$w_{h'h}$表示上一时刻隐藏层与当前时刻隐藏层的权值矩阵h‘＊h，y_k^t表示最终输出层的输出，这里是softmax生成的750维向量，最后一个句子输入完成后的y_k^t表示生成的文本向量共750维；公式2-1表示，输入句向量和上一层隐藏状态在隐藏层中的计算结果；公式2-2表示隐藏层的输出，即隐藏层的状态；公式2-3表示隐藏层到输出层的计算结果；公式2-4表示输出层最终的结果。之后会将最后序列生成的b_h^t传递给解码器。

[0085] 步骤三、训练模型decoder解码器中的长短期记忆网络LSTM，将编码器中递归神经网络生成的隐藏状态作为输入，结合摘要训练数据(在之前转换成向量的形式)放入网络中，生成预测摘要，相关公式如下：

[0086] $f_t=\sigma(W_f \cdot [h_{t-1},x_t]+b_f)$ 3-1

[0087] $i_t=\sigma(W_i \cdot [h_{t-1},x_t]+b_i)$ 3-2

[0088] $C'_t=\tanh(W_C \cdot [h_{t-1},x_t]+b_C)$ 3-3

[0089] $C_t=f_t*C_{t-1}+i_t*C_t$ 3-4

[0090] $o_t=\sigma(W_o \cdot [h_{t-1},x_t]+b_o)$ 3-5

[0091] $h_t=o_t*\tanh(C_t)$ 3-6

[0092] 在上述公式中，C_t表示当前LSTM中的状态，h_t表示当前网络隐藏层状态，x_t表示网络的输入，即摘要训练数据的向量；公式3-1到公式3-6表示在LSTM中结合上一时刻隐藏层的状态和当前时刻的输入决定当前时刻隐藏层的状态，得到h_t之后，会通过同递归神经网络相似的输出层softmax得到预测摘要，softmax的输出层是300维大小同词向量。

[0093] 整个网络的训练是分层训练，原始训练数据80％用于训练，20％用于微调。

[0094] 1)进入编码器第一步生成句向量，将传入的文本词向量数据中的每句所有的词向量作为卷积神经网络的输入，经过卷积核(3,300)，以及max-pooling的池化方式，最终生成300维的句向量；

[0095] 2)将生成的句向量，一共10句传入递归神经网络中，生成初始参数设置在[-1,1]，满足高斯分布，其中递归神经网络第一步的隐藏状态设置为0，最终生成750维的句向量，以及最后一步的隐藏状态；

[0096] 3)将编码器生成的最后一步隐藏状态传入解码器作为长短期记忆神经网络的第一步的隐藏状态输入，第一步输入层的输入数据是文本结束标志<EOS>，后面步的输入是训练数据中的摘要数据部分，摘要数据被转换成词向量形式同文本。

[0097] 4)对模型进行评估，这里用到DUC-200数据。

[0098] 模型评估指标是ROUGE，主要是比较实际摘要和预测摘要重合程度，ROUGE-1表示就单个词的重复程度，ROUGE-2表示就两个词相连的重复程度。

[0099] 模型的训练使用hinton提出的分层训练方式，梯度参数的调整是反向传播方式，训练数据是收集的博客摘要数据，运用80％的数据进行训练，20％的数据进行测试。

[0100] 整个模型构建训练将在谷歌深度学习平台tensorflow上进行，训练将调用GPU，GPU在处理高维数据计算上效果明显，是调用CPU的5到8倍。

[0101] 步骤四、使用摘要生成模型生成预测摘要

[0102] 1)将要预测的数据进行文本预选择及向量的转换；

[0103] 2)将生成的向量数据放入训练好的深度学习摘要生成模型中，生成预测摘要。

[0104] 下面以具体实施案例对本发明进行进一步的详细说明。

[0105] 1)博客训练数据爬取自csdn网站的人气博客，内容包括移动开发、web前端、架构设计、编程语言、互联网、数据库、系统运维、云计算、研发管理9个专业方向，共21600篇博客，命名格式为姓名_索引号。得到的博客内容包括了多种元素，文本、图片链接、计算公式、代码等，由于图片、计算公式、代码元素在文本摘要生成的过程中并没有帮助，因此过滤掉这些元素，只留下文本；

[0106] 2)对博客数据进行预选择，选择首段加上其他任意一段，可以更加有效地生成摘要；将数据限制在每篇博客20*10的词量，是为了方便在模型中运用，深度学习训练复杂，大量的参数调整会耗费时间，将数据尽可能精简、提取文本特征是必要的，同时这样也对变长文本的问题进行了，将变长文本转换成定长文本，可以拥有更多的训练数据；

[0107] 3)通过word2vec生成的词典将文本训练数据转换成向量；

[0108] 4)构建深度学习摘要生成模型，使用数据进行训练，数据中的80％进行训练，20％进行测试。

[0109] 5)使用DUC-200评估模型，评估指标是ROUGE-1、ROUGE-2、ROUGE-L，ROUGE指标和BLEU指标都是用来针对机器翻译等系列自然语言处理任务进行评估的，它们的核心都是分析候选译文和参考译文n元组共同出现的程度，这里的1、2表示1元组、2元组，L表示最长子序列共同出现的程度相关公式如下：

[0110]
$$\text{ROUGE-N} = \frac{\sum_{S\in\{ReferemceSummaries\}}\sum_{gram_n\in S} Count_{match}(gram_n)}{\sum_{S\in\{ReferenceSummaries\}}\sum_{gram_n\in S} Count(gram_n)} \qquad 4\text{-}1$$

[0111] 6)为比较本发明的技术优势，设置对比试验，对比本发明使用的模型和当前已有摘要生成模型的效果。

[0112] 深度学习模型间摘要生成对比实验结果如表2所示

[0113]

DUC-200	ROUGE-1	ROUGE-2	ROUGE-L
CRL	47.4	23.0	43.5

[0114]

ILP	45.4	21.3	42.8
LEAD	43.6	21.0	40.2
URANK	48.5	21.5	-
TGRAPH	48.1	24.3	-

[0115] 在上表中，CRL是本发明使用的深度学习模型，ILP、LEAD、URANK、TGRAPH是已经存在的另外四种摘要生成模型。

[0116] 表2实验结果比较

[0117] 通过上述分析可见，本发明使用的模型在当前已有模型中的总体效果是最优的，虽然URANK、TGRAPH在ROUGE-1、ROUGE-2指标上的表现稍好，但是在ROUGE-L上基本不能表现出来。因此，本模型适合用来实现摘要生成任务，同时对机器翻译、阅读理解等自然语言处理方面的效果也较理想。由此可见，本发明具有实质性技术特点，其应用前景非常广阔。

[0118] 7)将想要进行摘要预测的博客，如果该博客只有图片、视频之类的，判定博客无效无法生成摘要；传入该深度学习摘要生成系统中，系统对其进行文本预选择和向量转换，传入训练的模型中，最终系统将模型预测的摘要返回给用户，效果如表3所示。

[0119]

博客文本	实际摘要	预测摘要
近日在新浪微博关于“数学该滚出高考吗”的调查显示，在近10万名网友当中，居然有7成支持，并吐槽被数学虐过的那些年自己成了“做题机器”。 数学赋予理性以鲜活生命，涤尽有史以来人类的蒙昧和无知，摧毁和构建了诸多宗教教义，给人类思想增加了无尽光辉。欧几里得第一次提出了认识宇宙的数学设计图使命，第一次提出了人的理性思维应该遵循的规则。“文艺复兴”时期，数学成为当时科技革命的旗帜，其主题为“认识宇宙，也认识人类自己”。在牛顿时代，机械唯物决定论成为当时科技革命的指导思想，而微积分则是最主要的武	数学在当前社会仍然有举足轻重的作用，有利于形成优秀严谨的思维习惯。	数学在优化思维方式等方面非常重要。

[0120]

器。在社会数学化的今天，公理化的无穷小微积分更是起着举足轻重的作用。		

[0121] 注：由于博客过长，因此不全部显示，只展示最终结果，原博链接如下：

[0122] http://blog.csdn.net/yuanmeng001/article/details/58871130

[0123] 上述实例只为说明本发明的技术构思及特点，其目的在于让熟悉此项技术的人是能够了解本发明的内容并据以实施，并不能以此限制本发明的保护范围。凡根据本发明精神实质所做的等效变换或修饰，都应涵盖在本发明的保护范围之内。

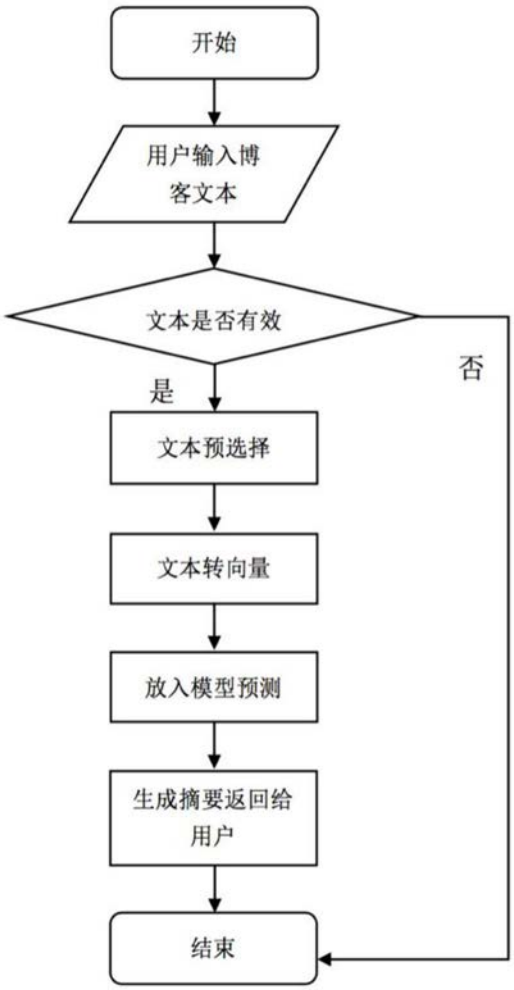


图1

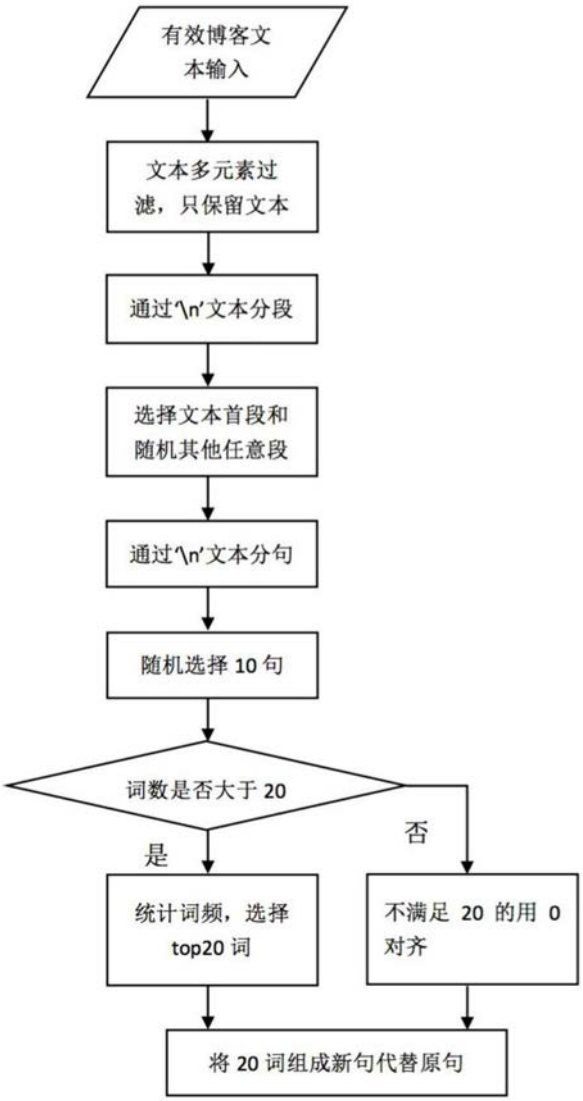

有效博客文本输入
文本多元素过滤，只保留文本
通过'\n'文本分段
选择文本首段和随机其他任意段
通过'\n'文本分句
随机选择 10 句
词数是否大于 20
是
否
统计词频，选择 top20 词
不满足 20 的用 0 对齐
将 20 词组成新句代替原句

图2

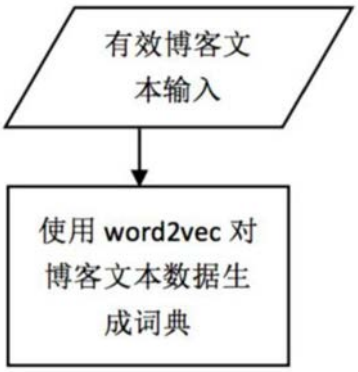

有效博客文本输入
使用 word2vec 对博客文本数据生成词典

图3

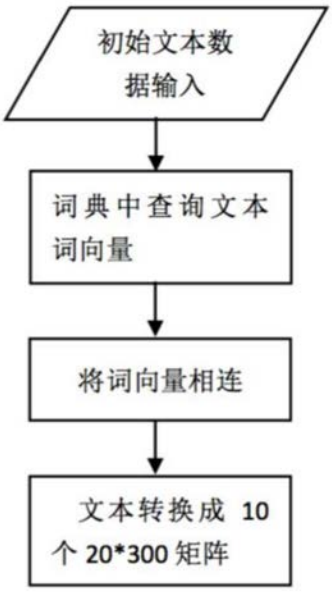

初始文本数据输入
词典中查询文本词向量
将词向量相连
文本转换成 10 个 20*300 矩阵

图4

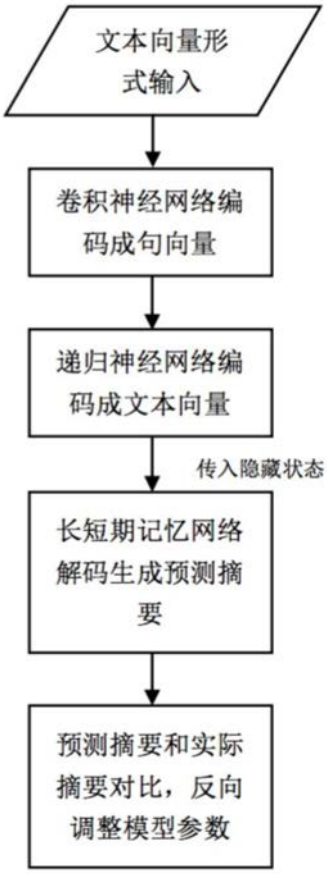

文本向量形式输入
卷积神经网络编码成句向量
递归神经网络编码成文本向量
传入隐藏状态
长短期记忆网络解码生成预测摘要
预测摘要和实际摘要对比，反向调整模型参数

图5