(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 111626048 A
(43)申请公布日 2020.09.04

(21)申请号 202010442510.X

(22)申请日 2020.05.22

(71)申请人 腾讯科技(深圳)有限公司
地址 518000 广东省深圳市南山区高新区科技中一路腾讯大厦35层

(72)发明人 洪科元 李斌 章秦 苏晨

(74)专利代理机构 北京派特恩知识产权代理有限公司 11270
代理人 王花丽 张颖玲

(51)Int.Cl.
G06F 40/232(2020.01)
G06F 40/242(2020.01)

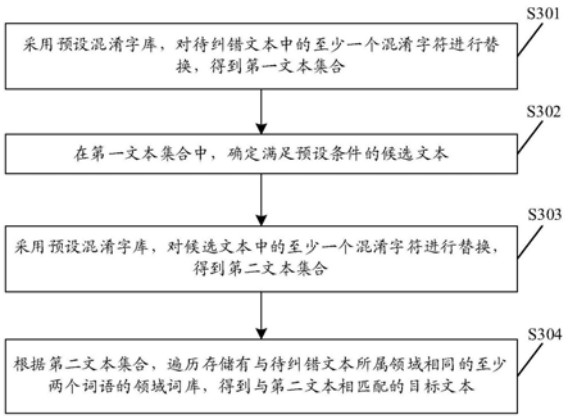
权利要求书2页 说明书17页 附图7页

(54)发明名称

文本纠错方法、装置、设备及存储介质

(57)摘要

本申请实施例提供了一种文本纠错方法、装置、设备及存储介质;所述方法包括:采用预设混淆字库,对待纠错文本中的至少一个混淆字符进行替换,得到第一文本集合;在所述第一文本集合中,确定满足预设条件的候选文本;采用所述预设混淆字库,对所述候选文本中的至少一个混淆字符进行替换,得到第二文本集合;根据所述第二文本集合,遍历存储有与所述待纠错文本所属领域相同的至少两个词语的领域词库,得到与第二文本相匹配的目标文本;这样,通过采用混淆字库和领域词库对待纠错文本进行纠错,能够对领域专有名词进行纠错,从而提高了对文本进行纠错的准确度。



CN 111626048 A

1.一种文本纠错方法,其特征在于,所述方法包括:

采用预设混淆字库,对待纠错文本中的至少一个混淆字符进行替换,得到第一文本集合;

在所述第一文本集合中,确定满足预设条件的候选文本;

采用所述预设混淆字库,对所述候选文本中的至少一个混淆字符进行替换,得到第二文本集合;

根据所述第二文本集合,遍历存储有与所述待纠错文本所属领域相同的至少两个词语的领域词库,得到与第二文本相匹配的目标文本。

2.根据权利要求1所述的方法,其特征在于,所述采用预设混淆字库,对所述待纠错文本中的至少一个混淆字符进行替换,得到第一文本集合,包括:

在所述预设混淆字库中,确定字形与所述待纠错文本中字符的字形之间的相似度大于等于第一预设相似度阈值的第一混淆字集合;

采用所述第一混淆字集合中的至少一个混淆字,对所述待纠错文本中对应的字符进行替换,得到所述第一文本集合。

3.根据权利要求1所述的方法,其特征在于,所述在所述第一文本集合中,确定满足预设条件的候选文本,包括:

确定所述第一文本集合中第一文本的每一字符出现的第一概率;

根据所述每一字符出现的第一概率,确定所述每一字符所属第一文本出现的第二概率;

将第二概率大于等于预设概率阈值的第一文本确定为所述候选文本。

4.根据权利要求1所述的方法,其特征在于,所述采用所述预设混淆字库,对所述候选文本中的至少一个混淆字符进行替换,得到第二文本集合,包括:

在所述预设混淆字库中,确定字形与所述候选文本中字符的字形之间的相似度,大于等于第二预设相似度阈值的第二混淆字集合;

采用所述第二混淆字集合中的至少一个混淆字,对所述候选文本中对应的字符进行替换,得到所述第二文本集合。

5.根据权利要求1至4任一项所述的方法,其特征在于,在所述采用预设混淆字库,对所述待纠错文本中相对应的至少一个混淆字符进行替换,得到第一文本集合之前,所述方法还包括:

获取至少包括两个字符的字符库;

确定所述字符库中字符的字形之间的相似度;

根据字形之间相似度大于预设相似度阈值的字符,创建所述预设混淆字库。

6.根据权利要求1所述的方法,其特征在于,所述根据所述第二文本集合,遍历存储有与所述待纠错文本所属领域相同的至少两个词语的领域词库,得到与第二文本相匹配的目标文本,包括:

确定与所述第二文本所属领域相同且存储至少两个词语的领域词典树;

根据所述第二文本集合,遍历所述词典树,得到所述目标文本。

7.根据权利要求6所述的方法,其特征在于,所述方法还包括:

从包括至少两种领域专有名词的词典库中,确定属于同一领域的M个专有名词;其中,M

为大于0的整数；

将所述同一领域的第i个专有名词的首位字符赋予所述领域词典树的父节点；其中，i为大于0小于等于M的整数；

将与所述首位字符相邻的字符，赋予所述父节点的子节点；

将所述专有名词的末位字符赋予所述子节点的叶子节点，以构建所述领域词典树。

8.一种文本纠错装置，其特征在于，所述装置包括：

第一替换模块，用于采用预设混淆字库，对待纠错文本中的至少一个混淆字符进行替换，得到第一文本集合；

第一确定模块，用于在所述第一文本集合中，确定满足预设条件的候选文本；

第二替换模块，用于采用所述预设混淆字库，对所述候选文本中的至少一个混淆字符进行替换，得到第二文本集合；

第一遍历模块，用于根据所述第二文本集合，遍历存储有与所述待纠错文本所属领域相同的至少两个词语的领域词库，得到与第二文本相匹配的目标文本。

9.一种文本纠错的设备，其特征在于，包括：

存储器，用于存储可执行指令；

处理器，用于执行所述存储器中存储的可执行指令时，实现权利要求1至7任一项所述的方法。

10.一种存储介质，其特征在于，存储有可执行指令，用于引起处理器执行时，实现权利要求1至7任一项所述的方法。

文本纠错方法、装置、设备及存储介质

技术领域

[0001] 本申请涉及文本处理技术领域，尤其涉及文本纠错方法、装置、设备及存储介质。

背景技术

[0002] 对文字进行识别的过程中，进行文字纠错的候选集由全量词典生成，查找候选字时，全量检索会造成搜索的空间过大，耗时较长；而且在形近字错误纠错的场景下，不同形近字组成的词的词向量可能会比较接近，无法保证区分的正确率。

发明内容

[0003] 本申请实施例提供一种文本纠错方法、装置、设备及存储介质，通过采用混淆字库和领域词库对待纠错文本进行纠错，能够对领域专有名词进行纠错，从而提高了对文本进行纠错的准确度。

[0004] 本申请实施例的技术方案是这样实现的：

[0005] 第一方面，本申请实施例提供一种文本纠错方法，包括：

[0006] 采用预设混淆字库，对待纠错文本中的至少一个混淆字符进行替换，得到第一文本集合；

[0007] 在所述第一文本集合中，确定满足预设条件的候选文本；

[0008] 采用所述预设混淆字库，对所述候选文本中的至少一个混淆字符进行替换，得到第二文本集合；

[0009] 根据所述第二文本集合，遍历存储有与所述待纠错文本所属领域相同的至少两个词语的领域词库，得到与第二文本相匹配的目标文本。

[0010] 第二方面，本申请实施例提供一种文本纠错装置，所述装置包括：

[0011] 第一替换模块，用于采用预设混淆字库，对待纠错文本中的至少一个混淆字符进行替换，得到第一文本集合；

[0012] 第一确定模块，用于在所述第一文本集合中，确定满足预设条件的候选文本；

[0013] 第二替换模块，用于采用所述预设混淆字库，对所述候选文本中的至少一个混淆字符进行替换，得到第二文本集合；

[0014] 第一遍历模块，用于根据所述第二文本集合，遍历存储有与所述待纠错文本所属领域相同的至少两个词语的领域词库，得到与第二文本相匹配的目标文本。

[0015] 第三方面，本申请实施例提供一种文本纠错的设备，包括：存储器，用于存储可执行指令；处理器，用于执行所述存储器中存储的可执行指令时，实现上述文本纠错方法。

[0016] 第四方面，本申请实施例提供一种存储介质，存储有可执行指令，用于引起处理器执行时，实现本申请实施例提供的文本纠错方法。

[0017] 本申请实施例具有以下有益效果：对于获取到的待纠错文本，首先利用混淆字库，构造待纠错文本的多个第一文本，然后，从对这多个第一文本进行纠错，从中确定出满足预设条件的候选文本；这样，对待纠错文本进行纠错时，候选字只从混淆集中选取字形上相近

的混淆字构成第一文本，可大幅减少对于句子合法性判断的计算量；然后，再采用混淆字库对候选文本中的混淆字进行替换；并且根据第二文本集合，遍历所述领域相同的领域词库，从而得到目标文本；这样，由领域专有名词构建的领域词库来对领域专有名词进行纠错，能够提高相同词在不同领域的区分准确度。

附图说明

[0018] 图1是本申请实施例提供的文本纠错系统的一个可选的架构示意图；

[0019] 图2A是本申请实施例提供的文本纠错系统的另一个可选的架构示意图；

[0020] 图2B是本申请实施例提供的文本纠错系统的结构示意图；

[0021] 图3是本申请实施例提供的文本纠错方法的实现流程示意图；

[0022] 图4A本申请实施例提供的文本纠错方法的又一实现流程示意图；

[0023] 图4B本申请实施例提供的文本纠错方法的又一实现流程示意图；

[0024] 图5是本申请实施例提供的文本纠错方法的实现流程示意图；

[0025] 图6是本申请实施例文本纠错方法的应用场景图；

[0026] 图7是本申请实施例文本纠错方法的另一应用场景图；

[0027] 图8是本申请实施例提供的文本纠错方法的另一实现流程示意图；

[0028] 图9A是本申请实施例进行OCR识别的过程的架构示意图；

[0029] 图9B是本申请实施例文本纠错方法的应用场景图；

[0030] 图10是本申请实施例领域词典树的组成结构示意图。

具体实施方式

[0031] 为了使本申请的目的、技术方案和优点更加清楚，下面将结合附图对本申请作进一步地详细描述，所描述的实施例不应视为对本申请的限制，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它实施例，都属于本申请保护的范围。

[0032] 在以下的描述中，涉及到“一些实施例”，其描述了所有可能实施例的子集，但是可以理解，“一些实施例”可以是所有可能实施例的相同子集或不同子集，并且可以在不冲突的情况下相互结合。

[0033] 在以下的描述中，所涉及的术语“第一\第二\第三”仅仅是是区别类似的对象，不代表针对对象的特定排序，可以理解地，“第一\第二\第三”在允许的情况下可以互换特定的顺序或先后次序，以使这里描述的本申请实施例能够以除了在这里图示或描述的以外的顺序实施。

[0034] 除非另有定义，本文所使用的所有的技术和科学术语与属于本申请的技术领域的技术人员通常理解的含义相同。本文中所使用的术语只是为了描述本申请实施例的目的，不是旨在限制本申请。

[0035] 对本申请实施例进行进一步详细说明之前，对本申请实施例中涉及的名词和术语进行说明，本申请实施例中涉及的名词和术语适用于如下的解释。

[0036] 1)形近字纠错：对文本中的形近字错误字符进行矫正。

[0037] 2)光学字符识别(Optical Character Recognition，OCR)：是指电子设备(例如扫描仪或数码相机)检查纸上打印的字符，通过检测暗、亮的模式确定其形状，然后用字符识

别方法将形状翻译成计算机文字的过程。

[0038] 3)自然语言处理(Natural Language Processing,NLP):是计算机科学领域与人工智能领域中的一个重要方向。自然语言处理研究能实现人与计算机之间用自然语言进行有效通信的各种理论和方法。

[0039] 4)混淆集:依据字形相似度从中文语料当中人工或自动整理出的易混形近字集合。

[0040] 5)前缀(Trie)树:是一种树形结构,是一种哈希树的变种。典型应用是用于统计,排序和保存大量的字符串(但不仅限于字符串),所以经常被搜索引擎系统用于文本词频统计。它的优点是:利用字符串的公共前缀来减少查询时间,最大限度地减少无谓的字符串比较,查询效率比哈希树高。

[0041] 6)二元语言模型:对于一段包含3个字符的文本“$W_1W_2W_3$”,W_1,W_2,W_3的混淆集分别为序列(W_{11},W_{12},W_{13},W_{14},W_{15}),序列(W_{21},W_{22},W_{23},W_{24},W_{25})和序列(W_{31},W_{32},W_{33},W_{34},W_{35})。在语言模型当中,基于马尔可夫假设,任意一个词出现的概率只与其前一个或几个词有关,与前n个词相关,则对应语言模型称为n-1元语言模型。这里采用二元语言模型,即任意一个词出现的概率只与其前一个词相关。

[0042] 7)区块链(Blockchain):由区块(Block)形成的加密的、链式的交易的存储结构。

[0043] 8)区块链网络(Blockchain Network):通过共识的方式将新区块纳入区块链的一系列的节点的集合。

[0044] 9)云技术(Cloud Technology)基于云计算商业模式应用的网络技术、信息技术、整合技术、管理平台技术、应用技术等的总称,可以组成资源池,按需所用,灵活便利。云计算技术将变成重要支撑。技术网络系统的后台服务需要大量的计算、存储资源,如视频网站、图片类网站和更多的门户网站。伴随着互联网行业的高度发展和应用,将来每个物品都有可能存在自己的识别标志,都需要传输到后台系统进行逻辑处理,不同程度级别的数据将会分开处理,各类行业数据皆需要强大的系统后盾支撑,只能通过云计算来实现。

[0045] 10)云存储(Cloud Storage),是在云计算概念上延伸和发展出来的一个新的概念,分布式云存储系统(以下简称存储系统)是指通过集群应用、网格技术以及分布存储文件系统等功能,将网络中大量各种不同类型的存储设备(存储设备也称之为存储节点)通过应用软件或应用接口集合起来协同工作,共同对外提供数据存储和业务访问功能的一个存储系统。

[0046] 在相关技术中,对文本进行纠错的过程中,当进行纠错的候选集由全量词典生成,查找候选字时,全量检索会造成搜索的空间过大,且因大部分候选字是与当前文字不相关的,所以全量检索候选字会浪费大量时间来做无用功;而且在形近字错误纠错的场景下,不同形近字组成的词的词向量可能会比较接近,无法做到有效区分,从而会造成部分形近字错误无法有效检测出来。

[0047] 基于此,本申请实施例提供一种文本纠错方法、装置、设备及存储介质,基于形近字构造混淆字集,在对待纠错文本进行纠错时,第一文本集合只从混淆集中选取字形上最相近的少量的字作为第一文本,可大幅减少关于句子合法性判断的计算量。同时,在本申请实施例中不同的形近字都是明确不同的表示,所以也可避免神经语言模型因形近字向量较为相似而无法有效区分的问题。再者,本申请实施例采用由领域词典构建的领域词典树来

对领域专有名词进行纠错，可有效解决相同词在不同领域因歧义而无法有效区分的问题。

[0048] 下面说明本申请实施例提供的文本纠错的设备的示例性应用，本申请实施例提供的终端可以实施为各种类型的用户设备，也可以实施为服务器。下面，将说明终端实施为设备或服务器时示例性应用。服务器可以是独立的物理服务器，也可以是多个物理服务器构成的服务器集群或者分布式系统，还可以是提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、以及大数据和人工智能平台等基础云计算服务的云服务器。终端可以是智能手机、平板电脑、笔记本电脑、台式计算机、智能音箱、智能手表等，但并不局限于此。终端以及服务器可以通过有线或无线通信方式进行直接或间接地连接，本申请在此不做限制。

[0049] 参见图1，图1是本申请实施例提供的文本纠错系统的一个可选的架构示意图，为实现支撑一个示例性应用，首先，当获取到识别出的待纠错文本101时，采用混淆字库102对待纠错文本101中的至少一个混淆字符103进行替换，得到第一文本集合；并且从第一文本集合中确定出最合理的候选文本104；这样，对待纠错文本进行纠错时，候选字只从混淆集中选取字形上相近的混淆字构成第一文本，可大幅减少对于句子合法性判断的计算量。然后，再采用混淆字库102对候选文本104中的至少一个混淆字符105进行替换，得到第二文本集合106；最后，按照该第二文本集合106遍历领域词典树107，以通过目标词典树对第二文本中的专有名称进行再次纠错，得到目标文本108，并输出该目标文本；其中，该领域词典树以树结构存储与所述待纠错文本所属领域相同的至少两个词语。这样，由领域词典构建的领域词典树模型来对领域中的专有名词进行纠错，可有效解决相同词在不同领域因歧义而无法有效区分的问题。

[0050] 参见图2A，图2A是本申请实施例提供的文本纠错系统的另一个可选的架构示意图，包括区块链网络20(示例性示出了作为原生节点的服务器200)、监测系统30(示例性示出归属于监测系统30的设备300及其图形界面301)，下面分别进行说明。

[0051] 区块链网络20的类型是灵活多样的，例如可以为公有链、私有链或联盟链中的任意一种。以公有链为例，任何业务主体的电子设备例如用户设备和服务器，都可以在不需要授权的情况下接入区块链网络20；以联盟链为例，业务主体在获得授权后其下辖的电子设备(例如设备/服务器)可以接入区块链网络20，此时，成为区块链网络20中的一类特殊的节点即终端节点。

[0052] 需要指出地，终端节点可以只提供支持业务主体发起交易(例如，用于上链存储数据或查询链上数据)功能，对于区块链网络20的原生节点的功能，例如下文所述的排序功能、共识服务和账本功能等，终端节点可以缺省或者有选择性(例如，取决于业务主体的具体业务需求)地实现。从而，可以将业务主体的数据和业务处理逻辑最大程度迁移到区块链网络20中，通过区块链网络20实现数据和业务处理过程的可信和可追溯。

[0053] 区块链网络20接收来自业务主体(例如图2A中示出的监测系统30)的终端节点(例如，图2A中示出的归属于监测系统30的设备300)提交的交易，执行交易以更新账本或者查询账本，并在设备的用户界面(例如，设备300的图形界面301)显示执行交易的各种中间结果或最终结果。

[0054] 下面以监测系统接入区块链网络以实现文本纠错的上链为例说明区块链网络的示例性应用。

[0055] 监测系统30的设备300接入区块链网络20，成为区块链网络20的终端节点。设备300通过传感器获取待纠错文本；并且，将最终处理完成的指令以及目标文本反馈给区块链网络20中的服务器200或者保存在设备300中；在已对设备300部署上传逻辑或用户进行操作的情况下，设备300根据待处理任务/同步时间查询请求，生成对应更新操作/查询操作的交易，在交易中指定了实现更新操作/查询操作需要调用的智能合约、以及向智能合约传递的参数，交易还携带了监测系统30签署的数字签名(例如，使用监测系统30的数字证书中的私钥，对交易的摘要进行加密得到)，并将交易广播到区块链网络20。其中，数字证书可由监测系统30向认证中心31进行登记注册得到。

[0056] 区块链网络20中的原生节点，例如服务器200在接收到交易时，对交易携带的数字签名进行验证，数字签名验证成功后，根据交易中携带的监测系统30的身份，确认监测系统30是否是具有交易权限，数字签名和权限验证中的任何一个验证判断都将导致交易失败。验证成功后签署原生节点自己的数字签名(例如，使用原生节点的私钥对交易的摘要进行加密得到)，并继续在区块链网络20中广播。

[0057] 区块链网络20中具有排序功能的节点接收到验证成功的交易后，将交易填充到新的区块中，并广播到区块链网络20中提供共识服务的节点。

[0058] 区块链网络20中的提供共识服务的节点对新区块进行共识过程以达成一致，提供账本功能的节点将新区块追加到区块链的尾部，并执行新区块中的交易：对于终端发起的文本纠错请求，可以通过预设混淆字库和领域词典树对待纠错文本进行多次纠错，从而得出准备率较高的目标文本，并将目标文本显示于设备300的图形界面301中。

[0059] 区块链网络20中的原生节点可从区块链中读取待纠错文本，并将待纠错文本呈现于原生节点的监测页面，原生节点也可以利用在区块链存储的待纠错文本，对该待纠错文本进行处理。

[0060] 在实际应用中，可为区块链网络20的不同原生节点设置不同的功能，例如设置服务器200具有文本纠错功能和记账功能。对于该情况，可在交易过程中，服务器200接收设备300发送的待文本纠错，在服务器200中，对待纠错文本进行多次纠错，通过采用混淆字库和领域词典树对待纠错文本进行纠错，能够对领域专有名词进行纠错，从而提高了相同词在不同领域的区分准确度。

[0061] 参见图2B，图2B是本申请实施例提供的文本纠错系统的结构示意图，图2B所示的设备400包括：至少一个处理器410、存储器450、至少一个网络接口420和用户接口430。设备400中的各个组件通过总线系统440耦合在一起。可理解，总线系统440用于实现这些组件之间的连接通信。总线系统440除包括数据总线之外，还包括电源总线、控制总线和状态信号总线。但是为了清楚说明起见，在图2B中将各种总线都标为总线系统440。

[0062] 处理器410可以是一种集成电路芯片，具有信号的处理能力，例如通用处理器、数字信号处理器，或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等，其中，通用处理器可以是微处理器或者任何常规的处理器等。

[0063] 用户接口430包括使得能够呈现媒体内容的一个或多个输出装置431，包括一个或多个扬声器和/或一个或多个视觉显示屏。用户接口430还包括一个或多个输入装置432，包括有助于用户输入的用户接口部件，在一些示例中键盘、鼠标、麦克风、触屏显示屏、摄像头、其他输入按钮和控件。

[0064] 存储器450可以是可移除的,不可移除的或其组合。示例性的硬件设备包括固态存储器,硬盘驱动器,光盘驱动器等。存储器450可选地包括在物理位置上远离处理器410的一个或多个存储设备。

[0065] 存储器450包括易失性存储器或非易失性存储器,也可包括易失性和非易失性存储器两者。非易失性存储器可以是只读存储器(Read Only Memory,ROM),易失性存储器可以是随机存取存储器(Random Access Memory,RAM)。本申请实施例描述的存储器450旨在包括任意适合类型的存储器。

[0066] 在一些实施例中,存储器450能够存储数据以支持各种操作,这些数据的示例包括程序、模块和数据结构或者其子集或超集,下面示例性说明。

[0067] 操作系统451,包括用于处理各种基本系统服务和执行硬件相关任务的系统程序,例如框架层、核心库层、驱动层等,用于实现各种基础业务以及处理基于硬件的任务;

[0068] 网络通信模块452,用于经由一个或多个(有线或无线)网络接口420到达其他计算设备,示例性的网络接口420包括:蓝牙、无线相容性认证、和通用串行总线(Universal Serial Bus,USB)等;

[0069] 呈现模块453,用于经由一个或多个与用户接口430相关联的输出装置431(例如,显示屏、扬声器等)使得能够呈现信息(例如,用于操作外围设备和显示内容和信息的用户接口);

[0070] 输入处理模块454,用于对一个或多个来自一个或多个输入装置432之一的一个或多个用户输入或互动进行检测以及翻译所检测的输入或互动。

[0071] 在一些实施例中,本申请实施例提供的装置可以采用软件方式实现,图2B示出了存储在存储器450中的服务器455,其可以是程序和插件等形式的软件,包括以下软件模块:第一替换模块4551、第一确定模块4552、第二替换模块4553和第一遍历模块4554;这些模块是逻辑上的,因此根据所实现的功能可以进行任意的组合或进一步拆分。将在下文中说明各个模块的功能。

[0072] 在另一些实施例中,本申请实施例提供的装置可以采用硬件方式实现,作为示例,本申请实施例提供的装置可以是采用硬件译码处理器形式的处理器,其被编程以执行本申请实施例提供的文本纠错方法,例如,硬件译码处理器形式的处理器可以采用一个或多个应用专用集成电路(Application Specific Integrated Circuit,ASIC)、DSP、可编程逻辑器件(Programmable Logic Device,PLD)、复杂可编程逻辑器件(Complex Programmable Logic Device,CPLD)、现场可编程门阵列(Field-Programmable Gate Array,FPGA)或其他电子元件。

[0073] 将结合本申请实施例提供的设备的示例性应用和实施,说明本申请实施例提供的文本纠错方法。

[0074] 参见图3,图3是本申请实施例提供的文本纠错方法的实现流程示意图,结合图3示出的步骤进行说明。

[0075] 步骤S301,采用预设混淆字库,对待纠错文本中的至少一个混淆字符进行替换,得到第一文本集合。

[0076] 在一些实施例中,首先获取到待纠错文本,比如,待纠错文本可以是通过对图片进行光学字符识别得到的识别结果,还可以是对语音进行文本识别得到的识别结果。在一些

可能的实现方式中，如果待纠错文本来自于待识别图片，可以通过首先，获取包括文本信息的待识别图像；比如，在智能核保产品的场景下，对用户的体检报告进行采集，得到待识别图像。然后，确定所述文本信息在所述待识别图像中占据的文本区域；比如，通过对待识别图像进行预处理，利用图片的像素点将图片进行二值化表示以便能够被后续的算法模型处理，然后对图片可能存在的倾斜、弯曲、褶皱等形变异常进行还原处理。最后，对所述文本区域进行特征提取，得到所述待纠错文本。比如，利用算法模型对图片中的文字区域进行检测，最后利用分类识别算法对识别出的文字区域的文字进行分类识别，得到识别结果，将该识别结果作为待纠错文本。然后，采用预设混淆字库，对待纠错文本中的至少一个混淆字符进行替换，得到第一文本集合。

[0077] 在一些实施例中，预设混淆字库包括多种字形相近的字，比如，硝的混淆字包括：消、销、削、绡和肖等，人的混淆字包括：个和大等。对于上述采用的混淆字库，可以通过以下过程进行创建：首先，获取至少包括两个字符的字符库；比如，先是建立包括大量汉字的汉字库；然后，确定字符库中字符的字形之间的相似度；比如，对字符库中的字符按照字符之间的字形相似度进行分类；最后，根据字形之间相似度大于预设相似度阈值的字符，创建预设混淆字库。比如，对于字符库中的字符C，从字符库中找出与字符C之间的相似度大于等于50％的字符，将这些字符作为字符C的混淆字，以构建混淆字库；这样，通过将字形之间相似度较大的字符作为一个字符的混淆字，减少了混淆字库中的字符数量，从而能够提高混淆字的检索速递，不会造成因为全量检索候选字而浪费大量时间的问题。

[0078] 在一些可能的实现方式中，首先，在待纠错文本中确定出包含混淆字库中的混淆字符，然后，对于这些混淆字符，采用混淆字库中对应的混淆字中的至少一个进行替换。比如，待纠错文本中包括两个混淆字符A和B，且每个混淆字符在混淆字库中有3个混淆字，即分别有3个字形相近的字(A1 A2 A3)和(B1 B2 B3)，那么用这3个字中的至少一个替换对应的混淆字符，经过多次替换，最终得到16个第一文本；比如，B不被替换的情况下，分别用(A1 A2 A3)中的每一个替换A，得到3种；A不被替换的情况下，分别用(B1 B2 B3)替换B，得到3种，然后，分别用(A1 A2 A3)中的每一个替换A，分别用(B1 B2 B3)中的一个替换B，得到9种，再加上AB均未被替换的一种文本，最终得到16个第一文本，即第一文本集合。在一些实施例中，为了减少计算量，可以是仅替换混淆字中的一部分，比如，基于上述例子，B不被替换的情况下，分别用(A1 A2 A3)中的任意两个替换A，得到2种；A不被替换的情况下，分别用(B1 B2 B3)中的任意两个替换B，得到2种，然后，分别用(A1 A2 A3)中的任意两个替换A，分别用(B1 B2 B3)中的任意两个替换B，得到4种，再加上AB均未被替换的一种文本，最终得到9个第一文本；但是为了保证第一文本集合的丰富性，要保证第一文本的数量大于等于最大数量的一半；也就是说，得到的第一文本集合的数量可以是大于等于最大文本数量的一半并且，小于等于最大文本数量。这样，对待纠错文本进行纠错时，候选字只从混淆集中选取字形上相近的混淆字构成第一文本，既可大幅减少对于句子合法性判断的计算量，还可以保证第一文本的丰富性。

[0079] 步骤S302，在第一文本集合中，确定满足预设条件的候选文本。

[0080] 在第一文本集合中，选择出满足预设条件的候选文本。在一些可能的实现方式中，可以采用经过通用语料训练得到的用于对文本进行语法检测的语言模型对每一第一文本的语法和语义的合理性进行检测，将合理度最高的作为候选文本；这样，采用语言模型对第

一文本的语法和语义的合理性进行判断，得到较为合理的候选文本，从而达到了对待纠错文本进行初次纠错的目的。

[0081] 步骤S303，采用预设混淆字库，对候选文本中的至少一个混淆字符进行替换，得到第二文本集合。

[0082] 在一些可能的实现方式中，首先，在候选文本中确定出包含混淆字库中的混淆字符，然后，对于这些混淆字符，采用混淆字库中对应的混淆字中的至少一个进行替换，得到第二文本集合。如上述例子所示，如果候选文本中包括两个混淆字符A和B，且每个混淆字符在混淆字库中有3个混淆字，即分别有3个字形相近的字(A1 A2 A3)和(B1 B2 B3)，那么用这3个字中的至少一个替换对应的混淆字符，经过多次替换，最终得到16个第二文本。在一些实施例中，为了减少计算量，可以是仅替换混淆字中的一部分，比如，基于上述例子，B不被替换的情况下，分别用(A1 A2 A3)中的任意两个替换A，得到2种；A不被替换的情况下，分别用(B1 B2 B3)中的任意两个替换B，得到2种，然后，分别用(A1 A2 A3)中的任意两个替换A，分别用(B1 B2 B3)中的任意两个替换B，得到4种，再加上AB均未被替换的一种文本，最终得到9个第二文本；但是为了保证第二文本集合的丰富性，要保证第二文本的数量大于等于最大第二文本数量的一半；也就是说，得到的第二文本集合的数量可以是大于等于最大第二文本数量的一半并且，小于等于最大第二文本数量。

[0083] 步骤S304，根据第二文本集合，遍历存储有与待纠错文本所属领域相同的至少两个词语的领域词库，得到与第二文本相匹配的目标文本。

[0084] 在一些实施例中，首先，确定与第二文本所属领域相同且存储至少两个词语的领域词典树；然后，根据第二文本集合，遍历词典树，得到目标文本。其中，领域词典树可以是由领域词典构建的前缀树模型，领域词典可以是由专家知识，由专家人工整理出领域专有名词；还可以是通过语义识别和场景识别对文本所属的领域进行自动分类，得到的名词库。

[0085] 在一些可能的实现方式中，首先要确定待纠错文本所属的领域，然后获取属于该领域的领域词典树，对于第二文本集合中的每一个第二文本，在词典树中进行遍历，如果能够在领域词典树中遍历到与该第二文本相同的文本序列，那么将该第二文本可作为目标文本。

[0086] 在本申请实施例中，对待纠错文本进行纠错时，候选字只从混淆集中选取字形上相近的混淆字构成第一文本，可大幅减少对于句子合法性判断的计算量；而且通过采用由领域词典构建的领域词典树来对第二文本中的领域专有名词进行纠错，可有效解决相同词在不同领域因歧义而无法有效区分的问题，从而提高对文本进行纠错的准确度。

[0087] 在一些实施例中，将预设混淆字库中的字形与待纠错文本的字形之间的相似度大于预设阈值的混淆字，作为能够替换待纠错文本中对应字符的混淆字，能够大幅减少纠错过程中关于句子合法性判断的计算量，即步骤S301可以通过以下步骤实现，参见图4A，图4A本申请实施例提供的文本纠错方法的又一实现流程示意图，结合图3进行以下说明：

[0088] 步骤S401，在预设混淆字库中，确定字形与待纠错文本中字符的字形之间的相似度大于等于第一预设相似度阈值的第一混淆字集合。

[0089] 在一些实施例中，混淆字库中包含多种字的混淆字，对于待纠错文本中的每一个字符，分别在混淆字库中查找该字符是字形相似度较高的混淆字，从而得到由多个字符的混淆字组合而成的第一混淆字集合。

[0090] 步骤S402,采用第一混淆字集合中的至少一个混淆字,对待纠错文本中对应的字符进行替换,得到第一文本集合。

[0091] 在一些实施例中,采用至少一个混淆字替换待纠错文本中与之字形相似的字符,从而得到一个第一文本,比如,如果待纠错文本中包括两个混淆字符A和B,且每个混淆字符在混淆字库中有2个混淆字,即分别有2个字形相近的字(A1 A2 A3)和(B1 B2 B3),那么用这2个字中的至少一个替换对应的混淆字符,经过多次替换,最终得到9个第一文本。这样,对待纠错文本进行纠错时,候选字只从混淆集中选取字形上相近的混淆字构成第一文本,可大幅减少对于句子合法性判断的计算量。

[0092] 在一些实施例中,可以采用训练好的语言模型对第一文本进行分类,以得到最合理的候选文本,从而达到纠错的目标,即步骤S302可以通过以下步骤实现:

[0093] 步骤S331,确定第一文本集合中第一文本的每一字符出现的第一概率。

[0094] 比如,首先,采用通用语料对神经网络进行训练,得到能够检测输入文本是否语法正确语义合理的语言模型;然后,利用在该语言模型中,确定第一文本中每一个字符出现的第一概率。比如,采用二元语言模型,确定该第一文本中每一个字符出现的概率。

[0095] 步骤S332,根据每一字符出现的第一概率,确定每一字符所属第一文本出现的第二概率。

[0096] 比如,将第一文本中每一字符出现的第一概率进行相乘,得到该文本序列出现的第二概率,在一个具体例子中,对于一段包含3个字符的文本“$W_1W_2W_3$”,W_1,W_2,W_3的混淆集分别为序列921(W_{11},W_{12},W_{13},W_{14},W_{15}),序列922(W_{21},W_{22},W_{23},W_{24},W_{25})和序列923(W_{31},W_{32},W_{33},W_{34},W_{35})。。这里采用二元语言模型,即任意一个词出现的概率只与其前一个词相关,则对于序列$W_{11}W_{21}W_{31}$,词W_{31}出现的概率与W_{21}相关,即W_{31}出现的概率$P(W_{31})$,可以表示为:$P(W_{31})=P(W_{31}|W_{21})=P(W_{21})*P(W_{31})*P(W_{21}W_{31})$;则序列$W_{11}W_{21}W_{31}$在$W_{11}W_{21}W_{31}$都出现的二元连乘概率为如公式(1)所示:

[0097]
$$P=P(W_{11},W_{21},W_{31})=P(W_{11})*P(W_{21}W_{31}|W_{11})$$
$$=P(W_{11})*P(W_{21})*P(W_{11}W_{21})*P(W_{31}|W_{11},W_{21}) \quad (1);$$
$$=P(W_{11})*P(W_{21})*P(W_{11}W_{21})*P(W_{31})*\ P(W_{21}W_{31})$$

[0098] 步骤S333,将第二概率大于等于预设概率阈值的第一文本确定为候选文本。

[0099] 在一个具体例子中,可以将第一文本集合中第二概率最大的第一文本作为候选文本;第二概率最大,说明该第一文本的语法最正确,语义最合理,即该文本序列是第一文本集合中最合理的文本序列。

[0100] 在本申请实施例中,通过采用通用语料对神经网络进行训练,得到能够检测输入文本是否语法正确语义合理的语言模型,将每一第一文本输入语言模型,即可得到该第一文本可能出现的概率,即该第一文本的合理度,概率越高,说明该第一文本越合理,从而能够从包括多个第一文本的第一文本集合中确定出语法正确语义合理的候选文本,提高了纠错的准确度。

[0101] 在一些实施例中,将预设混淆字库中的字形与候选文本的字形之间的相似度大于预设阈值的混淆字,作为能够替换候选文本中对应字符的混淆字,能够大幅减少纠错过程中关于句子合法性判断的计算量,即步骤S303可以通过以下步骤实现,参见图4B,图4B本申

请实施例提供的文本纠错方法的又一实现流程示意图，结合图3进行以下说明：

[0102] 步骤S421，在预设混淆字库中，确定字形与候选文本中字符的字形之间的相似度，大于等于第二预设相似度阈值的第二混淆字集合。

[0103] 在一些实施例中，混淆字库中包含多种字的混淆字，对于候选文本中的每一个字符，分别在混淆字库中查找该字符是字形相似度较高的混淆字，从而得到由多个字符的混淆字组合而成的第一混淆字集合。

[0104] 步骤S422，采用第二混淆字集合中的至少一个混淆字，对候选文本中对应的字符进行替换，得到第二文本集合。

[0105] 在一些实施例中，采用至少一个混淆字替换候选文本中与之字形相似的字符，从而得到一个第二文本，比如，如果候选文本中包括两个混淆字符A和B，且每个混淆字符在混淆字库中有3个混淆字，即分别有3个字形相近的字(A1 A2 A3)和(B1 B2 B3)，那么用这3个字中的至少一个替换对应的混淆字符，经过多次替换，最终得到16个第二文本。这样，对候选文本进行纠错时，候选字只从混淆集中选取字形上相近的混淆字构成第二文本，能够减少文字歧义带来的误纠、漏纠等问题。

[0106] 在一些实施例中，通过采用包含多种领域的专有名词的名词库，构建包含同一领域专有名词的前缀树，在该前缀树中遍历第二文本，从而得到准确率较高的目标文本，构建领域词典树的过程如下：

[0107] 首先，从包括至少两种领域专有名词的词典库中，确定属于同一领域的M个专有名词。

[0108] 在一些可能的实现方式中，M为大于0的整数。从按领域存储专有名词的词典库中，找出同一领域的多个专有名词，

[0109] 其次，将同一领域的第i个专有名词的首位字符赋予所述领域词典树的父节点。

[0110] 在一些可能的实现方式中，领域词典树的结构可以为前缀树结构。对于从词典库中找出的同一领域的多个专有名词，将该专有名词的首位字符，即该专有名词的第一个字，赋予领域词典树的父节点。也就是说，对于任意一个专有名词，将该名字的首字存储在词典树中的父节点，将首字的下一个字存储在该父节点的子节点，依次类推，这样，可以在查找的时候，可以在该父节点的路径上找到该专有名词。i为大于0小于等于M的整数。

[0111] 再次，将与首位字符相邻的字符，赋予父节点的子节点。

[0112] 在一个专有名词中，将首位子符的相邻字符，即首位字符的下一个字符，赋予存储首位字符的节点的子节点；然后，再将该下一个字符下一个字符，即第三个字符，赋予存储该下一个字符(即第二个字符)的节点的子节点，依次类推，将该专有名词的所有字符均赋予到领域词典树的节点中。

[0113] 最后，将专有名词的末位字符赋予所述子节点的叶子节点，以构建领域词典树。

[0114] 比如，该有专有名词中有三个字符，将首个字符赋予父节点，将第二个字符赋予父节点的子节点，然后，将第三个字符赋予该子节点的叶子节点，这样，就构建成了包含同一领域专有名词的领域词典树。

[0115] 在一些实施例中，通过领域词典构建的领域词典树来对领域专有名词进行纠错，以提高相同词在不同领域的区分准确度，即步骤S304可以通过以下步骤实现：

[0116] 步骤S351，确定待纠错文本所属的目标领域。

[0117] 比如，通过对待纠错文本进行语义分析和场景分析，确定待纠错文本中的内容所属的领域，比如，待纠错文本是“树莓派”，通过进行场景分析，确定该待纠错文本的应用场景是在描述互联网的文段中，那么确定这里的“树莓派”的目标领域为计算机领域。如果确定该待纠错文本的应用场景是在描述食物的文段中，那么确定这里的“树莓派”的目标领域为烹饪领域。

[0118] 步骤S352，确定第二文本集合中第j个第二文本的首位字符。

[0119] 在步骤S351中确定待纠错文本所属的目标领域之后，也就得到了第二文本集合中任一第二文本所属的领域；对于任一第二文本，首先确定该第二文本中的首位字符，以便于根据该首位字符在领域词典树中查找存储该字符的节点。

[0120] 步骤S353，根据第j个第二文本的首位字符，遍历属于目标领域的领域词典树中的节点。

[0121] 在一些实施例中，先根据待纠错文本所属的领域确定第二文本所属的领域，基于该领域，确定出属于该领域的领域词典树；然后，对于任一第二文本，比如，对于第j个第二文本，确定出第j个第二文本的首位字符，然后在领域词典树中的节点中查找是否存储有该首位字符；如果没有，则解释查找，说明第j个第二文本中可能仍然存在错误，最好不作为目标文本；如果有，则继续在该节点的子节点中查找是否存储有首位字符的下一个字符，同理，如果有下一个字符，则继续在子节点的子节点中查找是否存储有下一个字符的下一个字符；如果能够在这一条路径中查找到第j个第二文本的所有字符，那么第j个第二文本即可作为目标文本。

[0122] 在一些可能的方式中，首先，根据第j个第二文本的首位字符，遍历领域词典树中的父节点；比如，比如，待纠错文本是“亚销酸盐”，第j个第二文本为“亚消酸盐”，由于待纠错文本所属领域为化学领域，那么第j个第二文本所属的领域也是化学领域，基于此，首先，确定属于化学领域的领域词典树，然后在领域词典树中查找存储了字符“亚”的节点

[0123] 其次，如果领域词典树中的第k个父节点中存储有第j个第二文本的首位字符，确定首位字符的下一个字符。

[0124] 比如，如果找到了存储“亚”的节点，确定首字符“亚”的下一个字符“消”。

[0125] 再次，根据首位字符的下一个字符，遍历第k个父节点的子节点。

[0126] 比如，在该节点的子节点中查找“消”。

[0127] 最后，如果从所述第k个父节点的子节点到所述子节点的叶子节点的路径中存储有所述第j个第二文本中除所述首位字符之外的字符，将所述第j个第二文本确定为所述目标文本。

[0128] 比如，如果在该节点的子节点中没有找到“消”，则结束遍历，说明第j个第二文本“亚消酸盐”不是目标文本，即说明该第j个第二文本“亚消酸盐”可能是语法不合理或者语义不正确的文本，仍然有错误，所以舍弃该文本。如果在子节点中找了“消”，那么接着在该子节点的子节点中查找“酸”，同理，如果能够找到“酸”，则继续在该子节点的子节点的子节点中查找“盐”，同理，如果能够找到“盐”，可以将该第j个第二文本“亚消酸盐”作为一个目标文本。

[0129] 步骤S354，如果从领域词典树中的第k个父节点到第k个父节点的叶子节点的路径中存储的字符，与第j个第二文本相匹配，将第j个第二文本确定为目标文本。

[0130] 在一些实施例中，对于第j个第二文本，如果能够从第k个父节点到第k个父节点的叶子节点的这一条路径中，查找到第j个第二文本中包含的所有字符，那么将第j个第二文本可以作为目标文本。这样，由领域词典构建的领域词典树来对领域专有名词进行纠错，能够提高相同词在不同领域的区分准确度。

[0131] 在一些实施例中，如果从领域词典树中能够找到多个第二文本中的所有字符，那么即可得到多个目标文本，为了进一步从这多个目标文本中确定出正确率最高的文本，可以通过以下过程实现：

[0132] 首先，如果目标文本的数量大于等于2，确定每一目标文本与待纠错文本之间的相似度。

[0133] 这里，分别确定每一个目标文本与待纠错文本之间的相似度。

[0134] 然后，将相似度最大的目标文本，确定为最终文本。

[0135] 比如，将相似度最大的目标文本作为最终文本，并将该最终文本进行输出。这样，在目标文本为多个的情况下，通过确定目标文本与待纠错文本之间的相似度，来确定出最为相似的文本，作为最终文本，能够有助于减少利用混淆字替换过程中造成的替换不够准确的情况。

[0136] 下面，将说明本申请实施例在一个实际的应用场景中的示例性应用，以针对包括文本信息的待识别图像，采用光学字符识别得到识别结果，进行文本纠错为例，进行说明。

[0137] 在一些实施例中，中文中存在大量的形近字，对于OCR文字识别结果，由于受到图片质量影响，会存在大量的形近字识别错误，例如“亚硝酸盐”可能会识别为“亚消酸盐”，“项目名称”可能会识别为“项日名称”等。同时，由不同形近字组成的相近实体词可能都是合理的实体词，只有在特定领域才能判断其是否存在错误，如：“尿素氮”和“尿素氦”等。本申请实施例在保险领域的核保系统上进行了实践，发现了三个难点：

[0138] 1)数据量大，且涉及到大量的形近字，需要专业的语言学家来进行校对，标注难度大。

[0139] 2)具有强领域依赖属性，大量专有名词需要专业的知识才能确定是否存在字错误。

[0140] 3)如果图片质量较差，可能会存在多个字符连续错误的情况。

[0141] 在一些实施例中，对于文本进行纠错可以采用以下两种实现方式：

[0142] 方式一，基于神经语言模型来纠错，过程如下：

[0143] 首先利用词向量来对字词进行向量化表示，然后再根据神经网络来训练语言模型，最后利用语言模型对文本进行纠错。这种方案在此场景下存在候选字搜索空间较大，且形近字词之间因词向量可能相似而无法有效区分的问题。

[0144] 方式二，基于同音字来纠错，过程如下：

[0145] 基于同音字的纠错，首先基于文字的汉语拼音相同来构造出同音字候选集，然后再基于语言模型来对文本就行纠错。这种方案在因字形相近造成的识别错误的场景下并不适用。

[0146] 由此可见，当候选集由全量词典生成，查找候选字时，全量检索会造成搜索的空间过大，且因大部分候选字是与当前文字不相关的，所以全量检索候选字会浪费大量时间来做无用功；而且在形近字错误纠错的场景下，不同形近字组成的词的词向量可能会比较接

近，无法做到有效区分，从而会造成部分形近字错误无法有效检测出来。

[0147] 基于此，本申请实施例提供一种文本纠错方法，基于形近字构造混淆集，在用语言模型进行纠错时，候选字只从混淆集中选取字形上最相近的5个字作为候选项，可大幅减少语言模型中关于句子合法性判断的计算量。同时，本发明采用统计语言模型，不同的形近字都是明确不同的表示，所以也可避免神经语言模型因形近字向量较为相似而无法有效区分的问题。再者，本申请实施例集成了一个由领域词典构建的前缀树模型来对领域专有名词进行纠错，可有效解决相同词在不同领域因歧义而无法有效区分的问题。如图5所示，图5是本申请实施例提供的文本纠错方法的实现流程示意图，结合图5进行以下说明：

[0148] 步骤S501，获取通用语料。

[0149] 通用语料可以理解为是通用的词语和文段库。

[0150] 步骤S502，采用语料训练通用语言模型。

[0151] 步骤S503，确定领域专有名词，得到领域词典。

[0152] 比如，基于专家知识，由专家人工整理出领域专有名词，得到领域词典。

[0153] 步骤S504，采用领域词典和前缀树树构建查找树，得到领域词典树。

[0154] 在本实施例中，基于汉字字形相似度，整理形近字混淆集。

[0155] 步骤S505，获取对图像进行OCR的文字识别结果。

[0156] 步骤S506，采用训练好的语言模型对文字识别结果进行纠错，得到纠错结果。

[0157] 步骤S507，利用领域词典对纠错结果进行再纠错，得到最终的纠错结果。

[0158] 以下为形近字短文本纠错技术在智能核保产品及疾病预测应用场景中的场景图：

[0159] 在智能核保产品中，为了对申保人的投保资格进行审核，会基于申保人近期的体检报告对申保人的健康情况进行评估。对于申保人的体检报告，首先会先利用OCR文字识别从体检报告的拍照件或扫描件当中识别出所有的文字信息，然后利用形近字短文本纠错技术对可能存在的文字错误进行纠错，之后再利用文本分类、实体识别、关系抽取等自然语言处理技术对纠正后的正确文本做结构化信息提取，最后再基于核保模型，对提取出的体检信息特征进行核保评估。

[0160] 图6是本申请实施例文本纠错方法的应用场景图，结合图6进行以下说明：

[0161] 步骤S601，对体检报告进行拍照或扫描。

[0162] 步骤S602，对体检报告图像进行OCR文字识别，得到OCR识别结果。

[0163] 步骤S603，采用形近字短文本纠错对OCR识别结果进行纠错，得到纠错结果。

[0164] 步骤S604，对纠错结果进行文本分类，得到分类结果。

[0165] 对纠错结果进行文本分类包括，对纠错结果进行实体词识别41和实体关系抽取42，两个过程，从而将经过实体词识别41和实体关系抽取42进行处理后的分类结果，再进行结构化信息提取。

[0166] 步骤S605，对分类结果进行结构化信息提取，得到提取结果。

[0167] 步骤S606，将提取结果输入到核保模型，进行核保，得到核保结果。

[0168] 步骤S607，输出核保结果。

[0169] 在本申请实施例中，在疾病预测中，同样也是首先会基于OCR文字识别技术与NLP自然语言处理技术从体检报告中将有用的体检特征信息提取出来，然后再利用形近字短文本纠错技术对可能存在的文字错误进行纠正，之后再将纠正后的体检特征信息输入到疾病

预测模型当中，从而得到对体检人的可能存在的疾病情况进行预测的结果。

[0170] 图7是本申请实施例文本纠错方法的另一应用场景图，结合图7进行以下说明：

[0171] 步骤S701，获取体检报告。

[0172] 步骤S702，对体检报告进行OCR文字识别，得到OCR识别结果。

[0173] 步骤S703，对OCR识别结果进行NLP信息提取，得到有用的体检特征信息。

[0174] 步骤S704，对体检特征信息进行文本纠错，得到纠错结果。

[0175] 步骤S705，将纠错结果输入到疾病预测模型中，得到疾病预测结果。

[0176] 在本申请实施例中，对文本进行纠错的过程中，数据在算法内的流转过程如图8所示，结合图8所示步骤进行以下说明：

[0177] 步骤S801，获取待识别图片。

[0178] 步骤S802，对待识别图片进行OCR识别，得到OCR识别结果。

[0179] 在一些实施例中，对于体检报告、银行单据、发票等票据图片，首先利用OCR文字识别技术将图片上的文字识别出来。OCR文字识别算法一般包括预处理、版面处理、特征提取与模型训练和识别后处理等过程。如图9A所示，图9A是本申请实施例进行OCR识别的过程的架构示意图，包括：

[0180] 图片输入模块901，用于输入待识别图片。

[0181] 这里，待识别图片可以是一张包含文字信息的图片。

[0182] 预处理模块902，用于对图片进行预处理，利用图片的像素点将图片进行二值化表示以便能够被后续的算法模型处理。

[0183] 版面处理模块903，用于对图片可能存在的倾斜、弯曲、褶皱等形变异常进行还原处理。

[0184] 特征提取和模型训练模块904，用于利用算法模型对图片中的文字区域进行检测。

[0185] 识别后处理模块905，用于利用分类识别算法对识别出的文字区域的文字进行分类识别，得到文字。

[0186] 文字输出模块906，用于输出识别后的文字。

[0187] 对于识别出来的文本字段，基于二元语言模型，以形近字混淆集构造第一文本序列，然后计算每个序列的二元连乘概率，并从中选出概率最大的序列作为最合理的序列作为纠错后的正确序列。例如：如图9B所示，对于一段包含3个字符的文本“$W_1W_2W_3$”，W_1，W_2，W_3的混淆集分别为序列921(W_{11}，W_{12}，W_{13}，W_{14}，W_{15})，序列922(W_{21}，W_{22}，W_{23}，W_{24}，W_{25})和序列923(W_{31}，W_{32}，W_{33}，W_{34}，W_{35})。在语言模型当中，基于马尔可夫假设，任意一个词出现的概率只与其前一个或几个词有关，与前n个词相关，则对应语言模型称为n-1元语言模型。这里采用二元语言模型，即任意一个词出现的概率只与其前一个词相关，则对于序列$W_{11}W_{21}W_{31}$，词W_{31}出现的概率与W_{21}相关，即W_{31}出现的概率P(W_{31})，可以表示为：

[0188] P(W_{31})＝P(W_{31}|W_{21})＝P(W_{21})*P(W_{31})*P($W_{21}W_{31}$)；则序列$W_{11}W_{21}W_{31}$在$W_{11}W_{21}W_{31}$都出现的二元连乘概率为如公式(1)所示。

[0189] 二元语言模型计算公式，如公式(2)所示：

[0190] $$P(w_1,\cdots,w_n)=P(w_1)*P(w_2\mid w_1)*P(w_3\mid w_1,w_2)*\cdots P(w_n\mid w_1,\cdots,w_{n-1}) \quad (2);$$

[0191] 步骤S803，获取通用的语言知识。

[0192] 步骤S804,根据语言知识对语言模型训练,得到已训练语言模型。

[0193] 步骤S805,获取领域知识。

[0194] 这里,根据领域知识构建领域词典树81和混淆字集82。

[0195] 步骤S806,采用已训练语言模型、词典树和混淆字集对OCR识别结果进行纠错排序,得到正确文本。

[0196] 在本申请实施例中,对于语言模型纠错后的结果,利用Trie树模型进行领域名词纠错。如图10所示:作为父节点的“亚1001”包括子节点“硝1011”、“州1012”、“运1013”;作为父节点的“中1002”包括子节点“国1021”、“华1022”和“介1023”;“硝1011”包括子节点“酸1031”;“运1013”包括子节点:“村1032”和“会1033”;“国1021”包括子节点“人1034”和“心1035”;“酸1031”包括子节点:“盐1041”和“钠1042”。

[0197] 从图10可以看出,对于待纠错文本中的每一个字符,基于形近字混淆集候选字替换原始文本中对应的文字,然后从Trie树中查找替换后的文本是否在树中,如在,则用候选文字替换原始文本中对应的文字作为纠错后的结果。例如对于文本“亚消酸盐”,“消”的形近字混淆集为“消硝肖销绡”,当用“硝”第二文本中的“消”字时,发现“亚硝酸盐”在Trie树中,则用“硝”替换“亚消酸盐”中的“消”,将“亚硝酸盐”作为纠错后的结果。如图10所示,可以从图10中遍历到的词语有“亚硝酸盐、亚硝酸钠、亚洲、亚运村、亚运会、中国人、中国心、中华和中介”,这些词构成了图10所示的词典的Trie树结构。

[0198] 在医疗保险行业,由于技术设备限制及历史遗留等原因,积累了大量的纸质文档,文档电子化有助于提高信息处理效率,同时也是未来的一个趋势。由于现阶段技术的限制,对图片上文字的识别会存在一定的错误,基于形近字混淆集的文本纠错方案有助于减少文字识别的错误,提高文档电子化的准确性。这样,采用形近字混淆集来进行纠错,可有效缩小纠错的范围,提高纠错精度。利用领域词典构造树模型来进行纠错,可增加模型在专有领域的适应能力,减少文字歧义带来的误纠、漏纠等问题。

[0199] 下面继续说明本申请实施例提供的文本纠错的服务器455的实施为软件模块的示例性结构,在一些实施例中,如图2B所示,存储在存储器450的文本纠错的服务器455中的软件模块可以包括:

[0200] 第一替换模块4551,用于采用预设混淆字库,对待纠错文本中的至少一个混淆字符进行替换,得到第一文本集合;

[0201] 第一确定模块4552,用于在所述第一文本集合中,确定满足预设条件的候选文本;

[0202] 第二替换模块4553,用于采用所述预设混淆字库,对所述候选文本中的至少一个混淆字符进行替换,得到第二文本集合;

[0203] 第一遍历模块4554,用于根据所述第二文本集合,遍历存储有与所述待纠错文本所属领域相同的至少两个词语的领域词库,得到与第二文本相匹配的目标文本。

[0204] 在一些实施例中,所述第一替换模块4551,用于:在所述预设混淆字库中,确定字形与所述待纠错文本中字符的字形之间的相似度大于等于第一预设相似度阈值的第一混淆字集合;采用所述第一混淆字集合中的至少一个混淆字,对所述待纠错文本中对应的字符进行替换,得到所述第一文本集合。

[0205] 在一些实施例中,所述第一确定模块4552,用于:确定所述第一文本集合中第一文本的每一字符出现的第一概率;根据所述每一字符出现的第一概率,确定所述每一字符所

属第一文本出现的第二概率;将第二概率大于等于预设概率阈值的第一文本确定为所述候选文本。

[0206] 在一些实施例中,所述第二替换模块4553,还用于:在所述预设混淆字库中,确定字形与所述候选文本中字符的字形之间的相似度,大于等于第二预设相似度阈值的第二混淆字集合;采用所述第二混淆字集合中的至少一个混淆字,对所述候选文本中对应的字符进行替换,得到所述第二文本集合。

[0207] 在一些实施例中,在所述第一替换模块4551,还用于:获取至少包括两个字符的字符库;确定所述字符库中字符的字形之间的相似度;根据字形之间相似度大于预设相似度阈值的字符,创建所述预设混淆字库。

[0208] 在一些实施例中,所述第一遍历模块4554,用于:确定与所述第二文本所属领域相同且存储至少两个词语的领域词典树;根据所述第二文本集合,遍历所述词典树,得到所述目标文本。

[0209] 在一些实施例中,在所述第一替换模块4551,还用于:从包括至少两种领域专有名词的词典库中,确定属于同一领域的M个专有名词;其中,M为大于0的整数;将所述同一领域的第i个专有名词的首位字符赋予所述领域词典树的父节点;其中,i为大于0小于等于M的整数;将与所述首位字符相邻的字符,赋予所述父节点的子节点;将所述专有名词的末位字符赋予所述子节点的叶子节点,以构建所述领域词典树。

[0210] 在一些实施例中,在所述第一替换模块4551,用于:确定所述待纠错文本所属的目标领域;确定所述第二文本集合中第j个第二文本的首位字符;其中,j为大于0的整数;根据所述第j个第二文本的首位字符,遍历属于所述目标领域的领域词典树中的节点;如果从所述领域词典树中的第k个父节点到所述第k个父节点的叶子节点的路径中存储的字符,与所述第j个第二文本相匹配,将所述第j个第二文本确定为所述目标文本。

[0211] 在一些实施例中,在所述第一替换模块4551,还用于:根据所述第j个第二文本的首位字符,遍历所述领域词典树中的父节点;如果所述领域词典树中的第k个父节点中存储有所述第j个第二文本的首位字符,确定所述首位字符的下一个字符;根据所述首位字符的下一个字符,遍历所述第k个父节点的子节点;如果从所述第k个父节点的子节点到所述子节点的叶子节点的路径中存储有所述第j个第二文本中除所述首位字符之外的字符,将所述第j个第二文本确定为所述目标文本。

[0212] 在一些实施例中,在所述第一替换模块4551,还用于:如果所述目标文本的数量大于等于2,确定每一目标文本与所述待纠错文本之间的相似度;将相似度最大的目标文本,确定为最终文本。

[0213] 本申请实施例提供一种存储有可执行指令的存储介质,其中存储有可执行指令,当可执行指令被处理器执行时,将引起处理器执行本申请实施例提供的方法。在一些实施例中,存储介质可以是闪存、磁表面存储器、光盘、或光盘存储器等存储器;也可以是包括上述存储器之一或任意组合的各种设备。

[0214] 在一些实施例中,可执行指令可以采用程序、软件、软件模块、脚本或代码的形式,按任意形式的编程语言(包括编译或解释语言,或者声明性或过程性语言)来编写,并且其可按任意形式部署,包括被部署为独立的程序或者被部署为模块、组件、子例程或者适合在计算环境中使用的其它单元。作为示例,可执行指令可以但不一定对应于文件系统中的文

件,可以可被存储在保存其它程序或数据的文件的一部分,例如,存储在超文本标记语言(Hyper Text Markup Language,HTML)文档中的一个或多个脚本中,存储在专用于所讨论的程序的单个文件中,或者,存储在多个协同文件(例如,存储一个或多个模块、子程序或代码部分的文件)中。作为示例,可执行指令可被部署为在一个车载计算设备上执行,或者在位于一个地点的多个计算设备上执行,又或者,在分布在多个地点且通过通信网络互连的多个计算设备执行。综上所述,本申请实施例对于获取到的待纠错文本,首先利用混淆字库,构造待纠错文本的多个第一文本,然后,从对这多个第一文本进行纠错,从中确定出满足预设条件的候选文本;这样,对待纠错文本进行纠错时,候选字只从混淆集中选取字形上相近的混淆字构成第一文本,可大幅减少对于句子合法性判断的计算量;然后,再采用混淆字库对候选文本中的混淆字进行替换;并且根据第二文本集合,遍历所述领域相同的词典树,从而得到目标文本;这样,由领域词典构建的领域词库来对领域专有名词进行纠错,能够提高相同词在不同领域的区分准确度。

[0215] 以上所述,仅为本申请的实施例而已,并非用于限定本申请的保护范围。凡在本申请的精神和范围之内所作的任何修改、等同替换和改进等,均包含在本申请的保护范围之内。

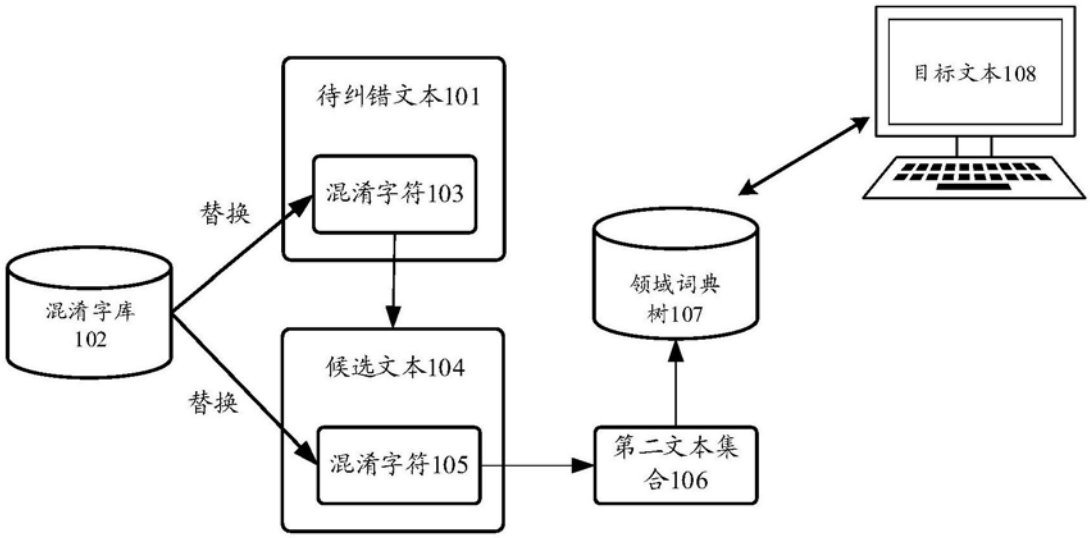

待纠错文本101
混淆字符103
替换
混淆字库
102
替换
候选文本104
混淆字符105
第二文本集
合106
领域词典
树107
目标文本108

图1

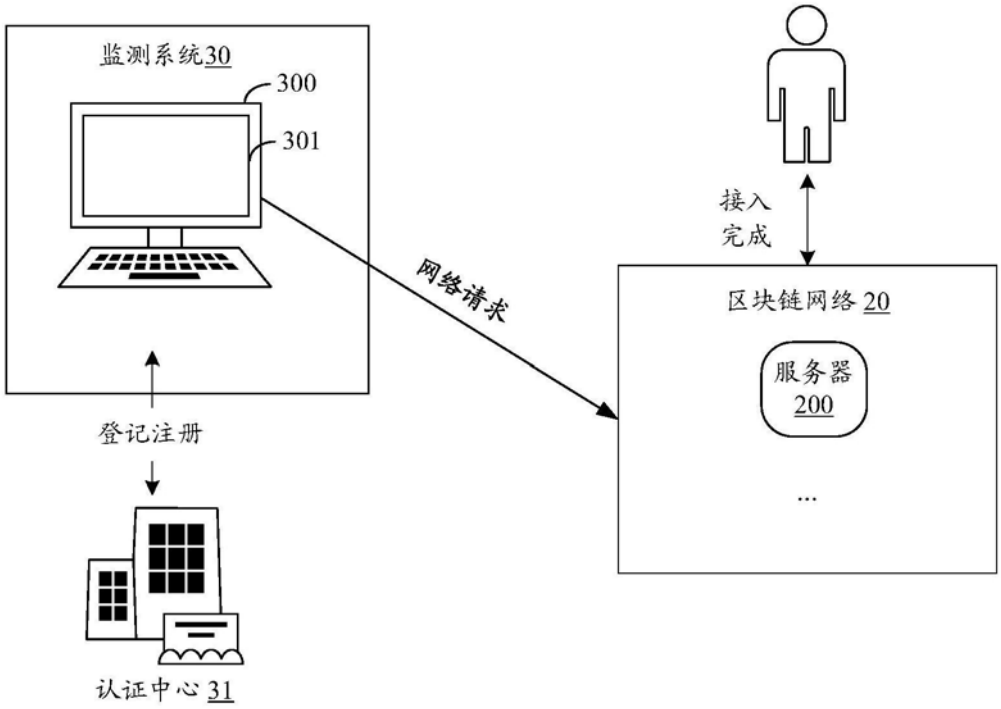

监测系统30
300
301
网络请求
登记注册
认证中心 31
接入
完成
区块链网络 20
服务器
200
...

图2A

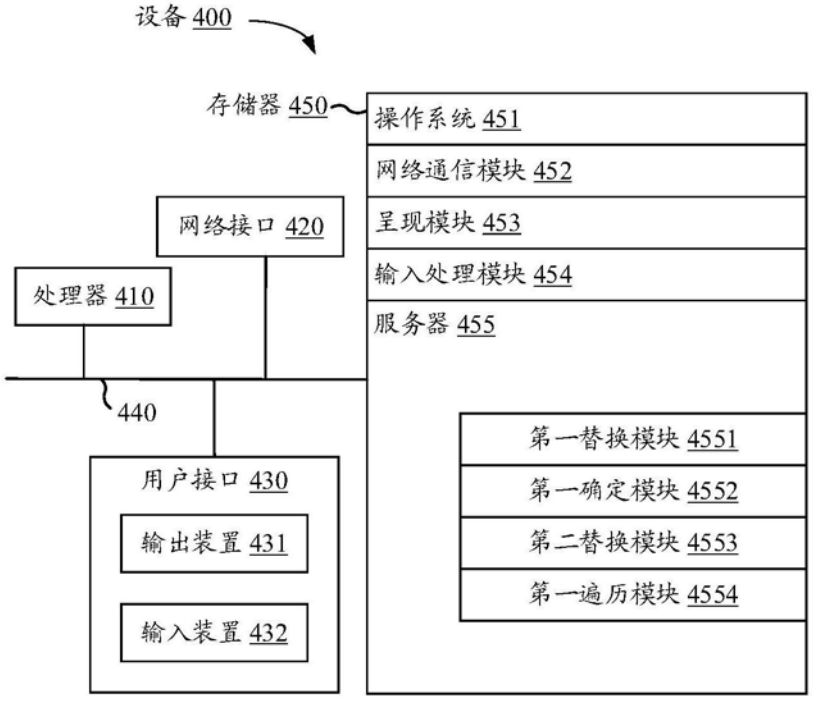

设备 400
存储器 450
操作系统 451
网络通信模块 452
呈现模块 453
输入处理模块 454
服务器 455
网络接口 420
处理器 410
440
第一替换模块 4551
第一确定模块 4552
第二替换模块 4553
第一遍历模块 4554
用户接口 430
输出装置 431
输入装置 432

图2B

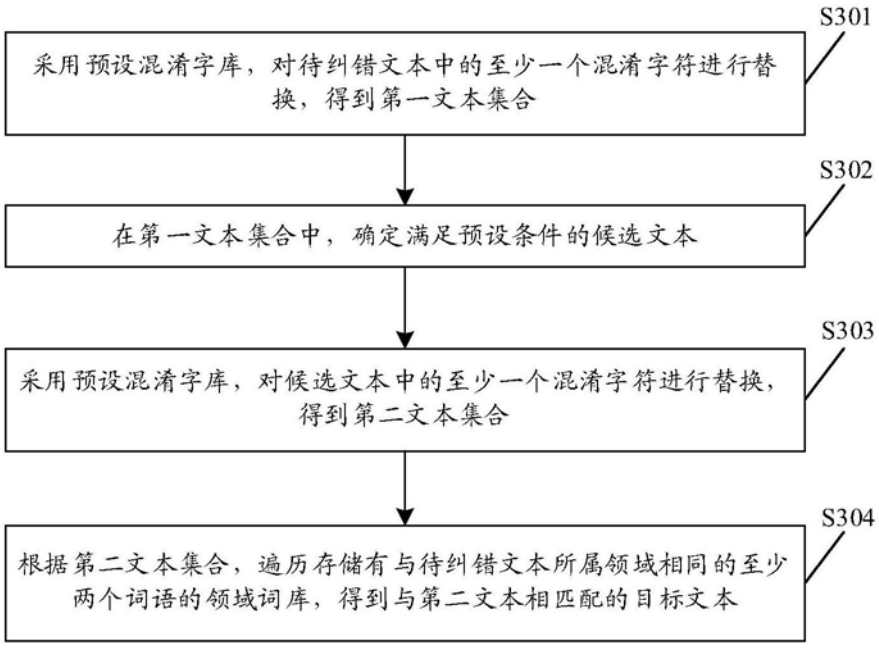

S301
采用预设混淆字库，对待纠错文本中的至少一个混淆字符进行替换，得到第一文本集合
S302
在第一文本集合中，确定满足预设条件的候选文本
S303
采用预设混淆字库，对候选文本中的至少一个混淆字符进行替换，得到第二文本集合
S304
根据第二文本集合，遍历存储有与待纠错文本所属领域相同的至少两个词语的领域词库，得到与第二文本相匹配的目标文本

图3

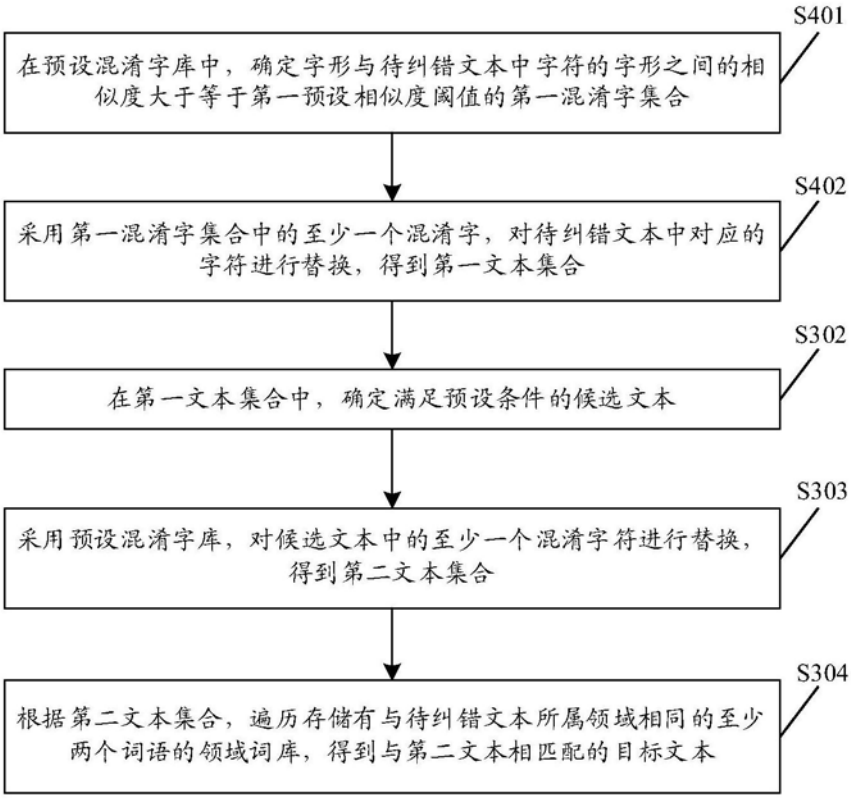

S401
在预设混淆字库中，确定字形与待纠错文本中字符的字形之间的相似度大于等于第一预设相似度阈值的第一混淆字集合
S402
采用第一混淆字集合中的至少一个混淆字，对待纠错文本中对应的字符进行替换，得到第一文本集合
S302
在第一文本集合中，确定满足预设条件的候选文本
S303
采用预设混淆字库，对候选文本中的至少一个混淆字符进行替换，得到第二文本集合
S304
根据第二文本集合，遍历存储有与待纠错文本所属领域相同的至少两个词语的领域词库，得到与第二文本相匹配的目标文本

图4A

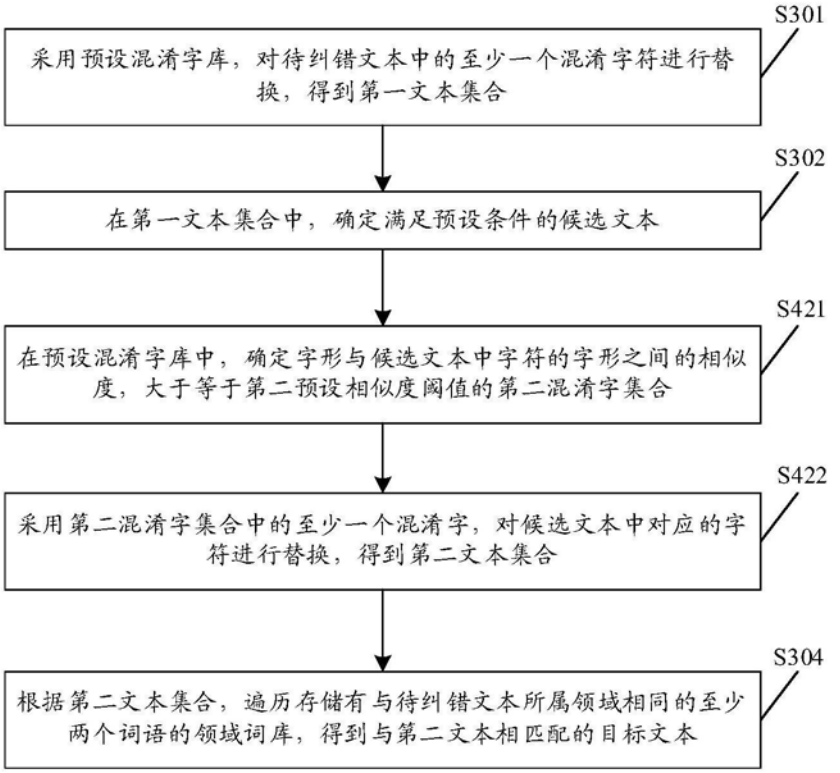

S301
采用预设混淆字库，对待纠错文本中的至少一个混淆字符进行替换，得到第一文本集合
S302
在第一文本集合中，确定满足预设条件的候选文本
S421
在预设混淆字库中，确定字形与候选文本中字符的字形之间的相似度，大于等于第二预设相似度阈值的第二混淆字集合
S422
采用第二混淆字集合中的至少一个混淆字，对候选文本中对应的字符进行替换，得到第二文本集合
S304
根据第二文本集合，遍历存储有与待纠错文本所属领域相同的至少两个词语的领域词库，得到与第二文本相匹配的目标文本

图4B

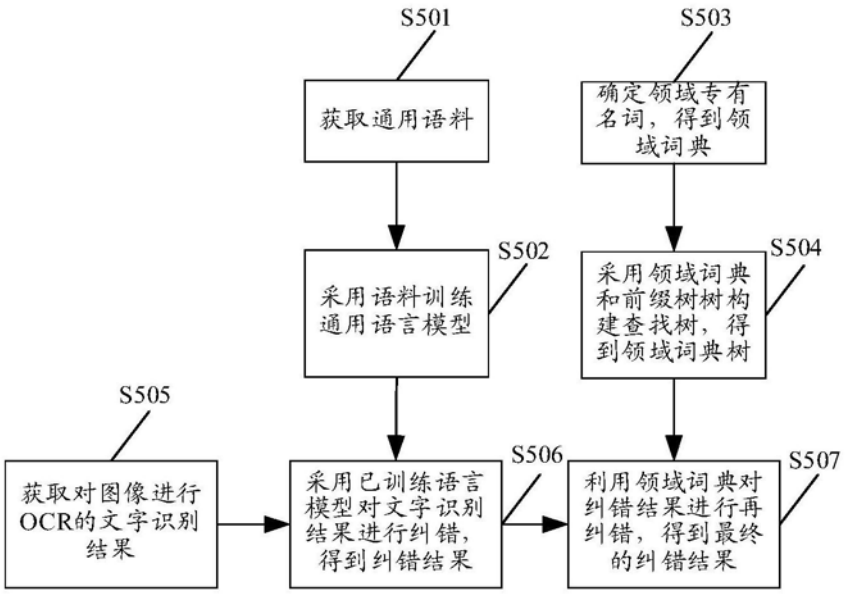

S501
获取通用语料
S503
确定领域专有名词，得到领域词典
S502
采用语料训练通用语言模型
S504
采用领域词典和前缀树树构建查找树，得到领域词典树
S505
获取对图像进行OCR的文字识别结果
S506
采用已训练语言模型对文字识别结果进行纠错，得到纠错结果
S507
利用领域词典对纠错结果进行再纠错，得到最终的纠错结果

图5

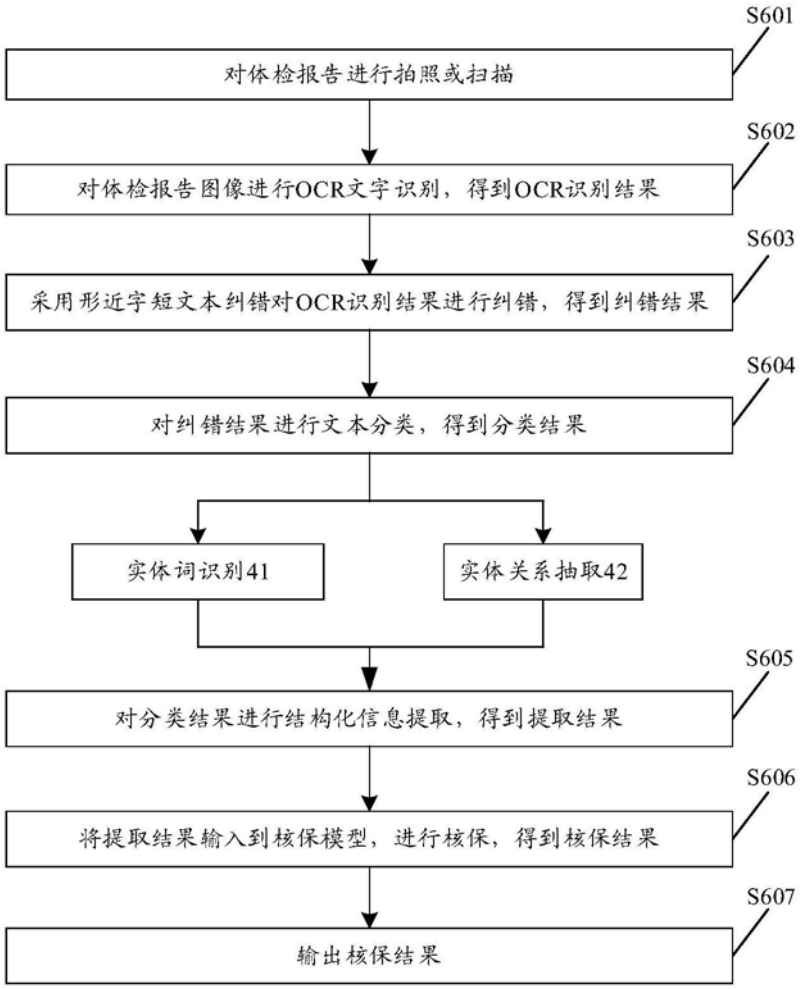

S601
对体检报告进行拍照或扫描
S602
对体检报告图像进行OCR文字识别，得到OCR识别结果
S603
采用形近字短文本纠错对OCR识别结果进行纠错，得到纠错结果
S604
对纠错结果进行文本分类，得到分类结果
实体词识别41
实体关系抽取42
S605
对分类结果进行结构化信息提取，得到提取结果
S606
将提取结果输入到核保模型，进行核保，得到核保结果
S607
输出核保结果

图6

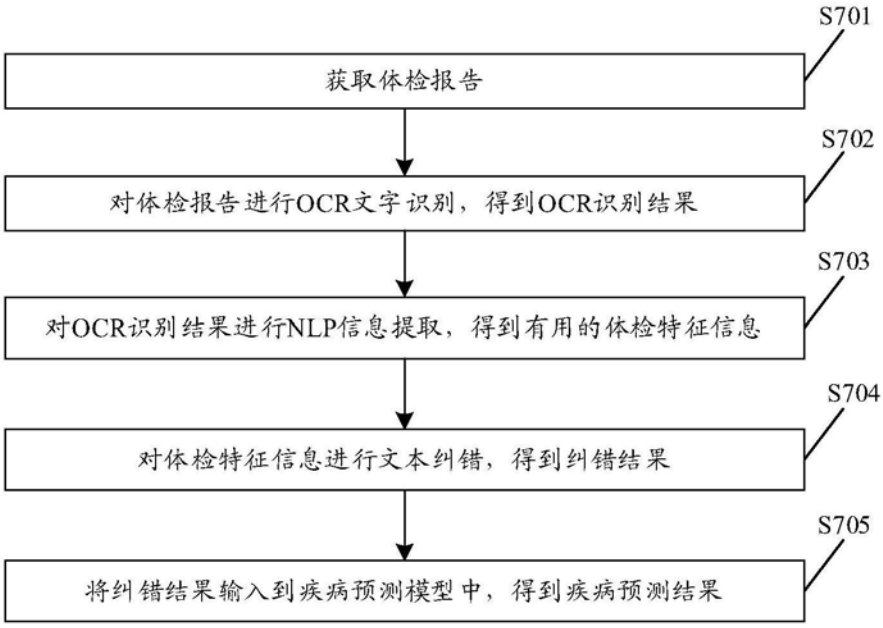

S701
获取体检报告
S702
对体检报告进行OCR文字识别，得到OCR识别结果
S703
对OCR识别结果进行NLP信息提取，得到有用的体检特征信息
S704
对体检特征信息进行文本纠错，得到纠错结果
S705
将纠错结果输入到疾病预测模型中，得到疾病预测结果

图7

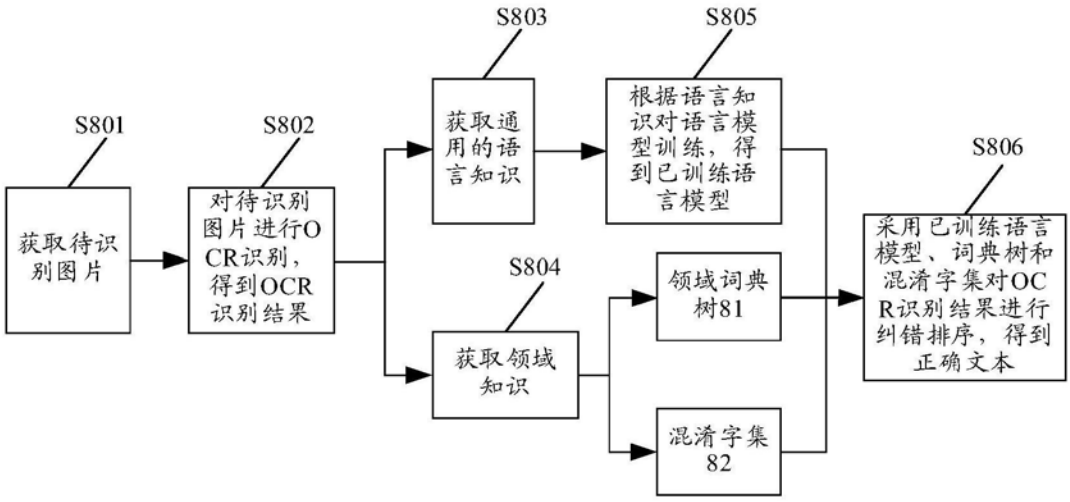

S801
获取待识别图片
S802
对待识别图片进行OCR识别，得到OCR识别结果
S803
获取通用的语言知识
S804
获取领域知识
S805
根据语言知识对语言模型训练，得到已训练语言模型
领域词典树81
混淆字集82
S806
采用已训练语言模型、词典树和混淆字集对OCR识别结果进行纠错排序，得到正确文本

图8

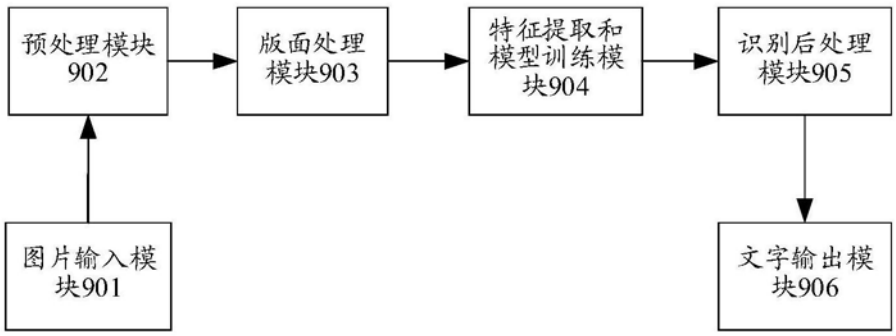

预处理模块902
版面处理模块903
特征提取和模型训练模块904
识别后处理模块905
图片输入模块901
文字输出模块906

图9A

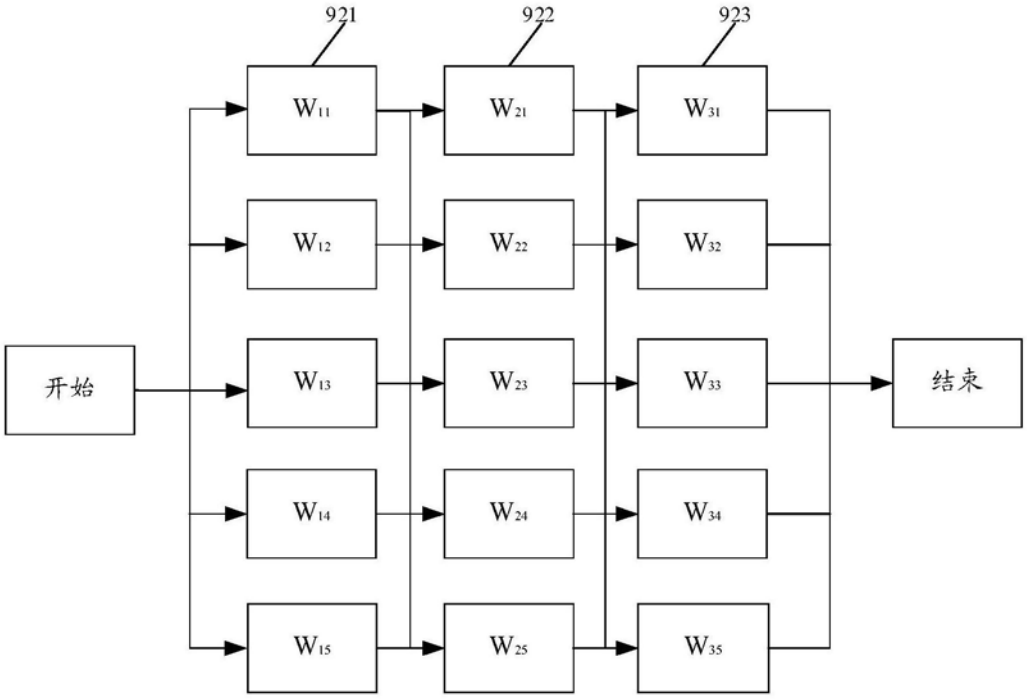

921
922
923
W11
W21
W31
W12
W22
W32
开始
W13
W23
W33
结束
W14
W24
W34
W15
W25
W35

图9B

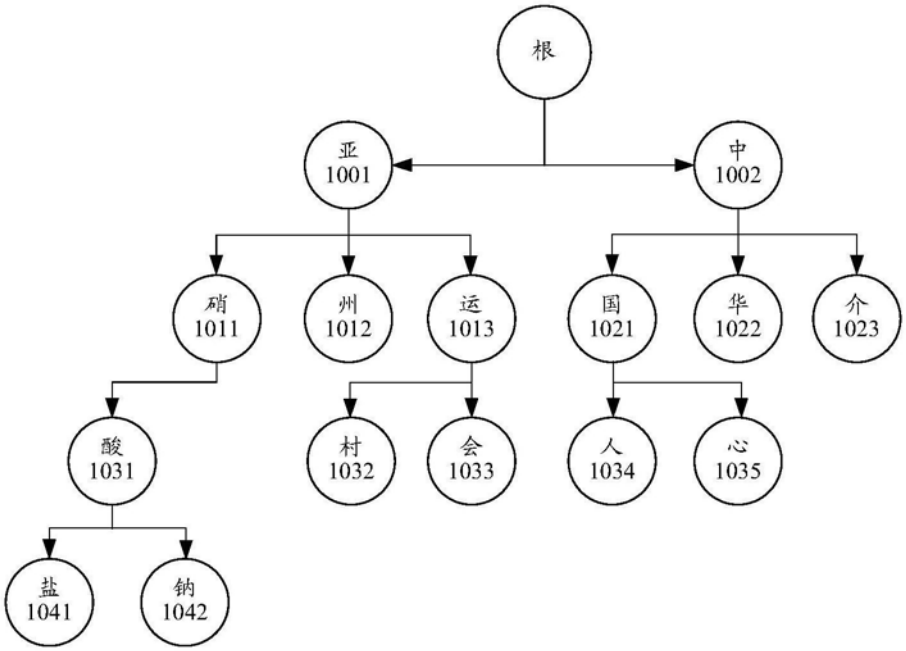

根
亚
1001
中
1002
硝
1011
州
1012
运
1013
国
1021
华
1022
介
1023
酸
1031
村
1032
会
1033
人
1034
心
1035
盐
1041
钠
1042

图10