



(12)发明专利

(10)授权公告号 CN 108417210 B

(45)授权公告日 2020.06.26

(21)申请号 201810022130.3

(22)申请日 2018.01.10

(65)同一申请的已公布的文献号

申请公布号 CN 108417210 A

(43)申请公布日 2018.08.17

(73)专利权人 苏州思必驰信息科技有限公司

地址 215123 江苏省苏州市苏州工业园区

星湖街328号创意产业园9-703

专利权人 上海交通大学

(72)发明人 俞凯 陈瑞年

(74)专利代理机构 北京商专永信知识产权代理

事务所(普通合伙) 11400

代理人 方挺 车江华

(51)Int.Cl.

G10L 15/22(2006.01)

(续)

(56)对比文件

CN 105654946 A,2016.06.08,

CN 106782560 A,2017.05.31,

CN 107491531 A,2017.12.19,

CN 101197083 A,2008.06.11,

US 9799327 B1,2017.10.24,

CN 107464559 A,2017.12.12,

CN 107391575 A,2017.11.24,

CN 107291795 A,2017.10.24,

CN 106126596 A,2016.11.16,

CN 103810999 A,2014.05.21,

CN 105244029 A,2016.01.13,

US 9779722 B2,2017.10.03,

US 8515745 B1,2013.08.20,

CN 104598441 A,2015.05.06,

CN 107544958 A,2018.01.05,

CN 106503146 A,2017.03.15,

CN 106383877 A,2017.02.08,

CN 107357789 A,2017.11.17,

CN 107180026 A,2017.09.19,

EP 2624149 A2,2013.08.07,

CN 107562715 A,2018.01.09,

Tomas Mikolov等.“Efficient Estimation of Word Representations in”.《arXiv preprint》.2013,

审查员 田静

权利要求书2页 说明书14页 附图6页

(54)发明名称

一种词嵌入语言模型训练方法、词语识别方法及系统

(57)摘要

本发明公开一种词嵌入语言模型训练方法,包括:确定语料库中的所有词语的属性以生成词表,所述属性包括所有词语的词性分类、所有词性分类的概率分布和所有词语在所属词性分类下的概率分布;生成所述词表中所有词语的词向量;生成对应于所述词表中所有词语的词性分类的词性分类向量;以所述词表中的词语的词向量和所述词表中的词语的词性分类向量为输入,以所述词表中的词语所属的词性分类的概率分布和所述词表中的词语在所属词性分类下的概率分布为输出进行训练,以得到所述词嵌入语言模型。本发明实施例中在进行语言模型,即便是遇

到OOV词语,也能通过该OOV词语的形态学信息和所述词性分类的句法级信息来进行准确的识别。



CN 108417210 B

[接上页]

(51) Int.Cl.

G06K 9/62(2006.01)

G10L 15/06(2013.01)

G10L 15/00(2013.01)

G10L 15/08(2006.01)

1. 一种词嵌入语言模型训练方法,包括:

确定语料库中的所有词语的属性以生成词表,所述属性包括所有词语的词性分类、所有词性分类的概率分布和所有词语在所属词性分类下的概率分布;

判断获取自所述词表的词语是否为低频词;

如果是,则将获取自所述词表的词语拆解成字,并对拆解得到的字进行编码以用于确定相应的词向量;

如果否,则提取所述获取自所述词表的词语的向量作为词向量;

生成对应于所述词表中所有词语的词性分类的词性分类向量;

以所述词表中的词语的词向量和所述词表中的词语的词性分类向量为输入,以所述词表中的词语所属的词性分类的概率分布和所述词表中的词语在所属词性分类下的概率分布为输出进行训练,以得到所述词嵌入语言模型。

2. 一种词语识别方法,所述方法采用了权利要求1中所述的词嵌入语言模型,所述方法包括:

生成待识别词语的词向量;

确定所述待识别词语的词性分类的词性分类向量;

将所述待识别词语的词向量和词性分类向量输入所述词嵌入语言模型,以获取所述待识别词语所属的词性分类的概率分布和所述待识别词语在所属词性分类下的概率分布;

当所述待识别词语属于用于训练所述词嵌入语言模型的词表时,所述生成所述待识别词语的词向量包括:

判断所述待识别词语是否为低频词;

如果是,则将所述待识别词语拆解成字,并对拆解得到的字进行编码以用于确定相应的词向量;

如果否,则提取所述待识别词语的向量作为词向量。

3. 根据权利要求2所述的方法,其中,当所述待识别词语不属于用于训练所述词嵌入语言模型的词表时,所述生成所述待识别词语的词向量包括:

确定所述待识别词语的属性以更新所述词表;

将所述待识别词语拆解成字,并对拆解得到的字进行编码以用于确定相应的词向量。

4. 一种词嵌入语言模型训练系统,包括:

词表生成程序模块,用于确定语料库中的所有词语的属性以生成词表,所述属性包括所有词语的词性分类、所有词性分类的概率分布和所有词语在所属词性分类下的概率分布;

频词判断程序单元,用于判断获取自所述词表的词语是否为低频词;

第一词向量生成程序单元,用于当判断获取自所述词表的词语为低频词时,将获取自所述词表的词语拆解成字,并对拆解得到的字进行编码以用于确定相应的词向量;

第二词向量生成程序单元,用于当判断获取自所述词表的词语为高频词时,提取所述获取自所述词表的词语的向量作为词向量;

词性分类向量生成程序模块,用于生成对应于所述词表中所有词语的词性分类的词性分类向量;

模型训练程序模块,用于以所述词表中的词语的词向量和所述词表中的词语的词性分

类向量为输入,以所述词表中的词语所属的词性分类的概率分布和所述词表中的词语在所属词性分类下的概率分布为输出进行训练,以得到所述词嵌入语言模型。

5. 一种词语识别系统,包括:

权利要求4中所述的词嵌入语言模型;

词向量生成程序模块,用于生成待识别词语的词向量;

词表生成程序模块,用于确定所述待识别词语的词性分类的词性分类向量;

词语识别程序模块,用于将所述待识别词语的词向量和词性分类向量输入所述词嵌入语言模型,以获取所述待识别词语所属的词性分类的概率分布和所述待识别词语在所属词性分类下的概率分布;

当所述待识别词语属于用于训练所述词嵌入语言模型的词表时,所述词向量生成程序模块包括:

频词判断程序单元,用于判断所述待识别词语是否为低频词;

第一词向量生成程序单元,用于当判断获取自所述词表的词语为低频词时,将所述待识别词语拆解成字,并对拆解得到的字进行编码以用于确定相应的词向量;

第二词向量生成程序单元,用于当判断获取自所述词表的词语为高频词时,提取所述待识别词语的向量作为词向量。

6. 根据权利要求5所述的系统,其中,当所述待识别词语不属于用于训练所述词嵌入语言模型的词表时,所述词向量生成程序模块包括:

词表更新程序单元,用于确定所述待识别词语的属性以更新所述词表;

词向量生成程序单元,用于将所述待识别词语拆解成字,并对拆解得到的字进行编码以用于确定相应的词向量。

7. 一种电子设备,其包括:至少一个处理器,以及与所述至少一个处理器通信连接的存储器,其中,所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行权利要求1-3中任意一项所述方法的步骤。

8. 一种存储介质,其上存储有计算机程序,其特征在于,该程序被处理器执行时实现权利要求1-3中任意一项所述方法的步骤。

一种词嵌入语言模型训练方法、词语识别方法及系统

技术领域

[0001] 本发明涉及人工智能技术领域,尤其涉及一种词嵌入语言模型训练方法、词语识别方法及系统。

背景技术

[0002] 目前应用于语音识别系统中的语言模型主要用于为语音识别出的词语及句子进行打分,并加上声学模型的打分,以得到识别出的最优结果。现有基于神经网络的语言模型训练速度较慢,并且训练时需要有固定已知的词表。传统的语言模型中,用于训练的词表中的每一个词用一个one-hot向量代表,例如,一个词表大小为1万(即,此表中有1万个词),那么表示一个词时采用一个一万维度的向量并且只有对应于该词的那一位为1。然后将该向量输入至神经网络之后与一个词嵌入矩阵相乘,转换成一个实数的向量,最终实现语言模型的训练;同样进行词语识别时也是将词语转化成实数向量进行识别。

[0003] 然而,发明人在实现本发明的过程中发现,用于训练语言模型的词表往往并不能够覆盖所有的词语,所以在使用传统语言模型时,如果碰到待识别词语是未录入词表的词语(例如,以后实际应用中所出现的词表之外的词out-of-vocabulary,OOV)时候,传统语言模型是无法对该词语进行正确可靠的识别的(因为,该待识别词语不存在于词表,所以根本就不存在对应于该待识别词语的向量,从而也就不能够与词嵌入向量相乘以得到相应于该OOV的实数向量),除非将该待识别词语添加至词表,并采用新的词表重新训练一个新的语言模型。

[0004] 为了解决传统语言模型中所存在的问题,目前最常用的就是用了一个特殊的标记<unk>代表所有超出词表的词。首先得到一个确定的词表,并加入一个特殊的<unk>字符。然后将训练集中所有OOV词全部替换为<unk>进行训练。在使用过程中,所有OOV词同样用<unk>代替。

[0005] 发明人在实现本发明的过程中发现,超出词表的词通常都是罕见词,因此这些词的训练数据本身就很少,现有技术中做法舍弃了这些OOV词所具有的语言学上的信息,因此,最后对这些词的识别结果也是十分不准确的。

发明内容

[0006] 本发明实施例提供一种在神经网络语言模型中添加新词的方法及系统,用于至少解决上述技术问题之一。

[0007] 第一方面,本发明实施例提供一种词嵌入语言模型训练方法,包括:

[0008] 确定语料库中的所有词语的属性以生成词表,所述属性包括所有词语的词性分类、所有词性分类的概率分布和所有词语在所属词性分类下的概率分布;

[0009] 生成所述词表中所有词语的词向量;

[0010] 生成对应于所述词表中所有词语的词性分类的词性分类向量;

[0011] 以所述词表中的词语的词向量和所述词表中的词语的词性分类向量为输入,以所

述词表中的词语所属的词性分类的概率分布和所述词表中的词语在所属词性分类下的概率分布为输出进行训练,以得到所述词嵌入语言模型。

[0012] 第二方面,本发明实施例还提供一种词语识别方法,所述方法采用了本发明上述实施例中的词嵌入语言模型,所述方法包括:

[0013] 生成待识别词语的词向量;

[0014] 确定所述待识别词语的词性分类的词性分类向量;

[0015] 将所述待识别词语的词向量和词性分类向量输入所述词嵌入语言模型,以获取所述待识别词语所属的词性分类的概率分布和所述待识别词语在所属词性分类下的概率分布。

[0016] 第三方面,本发明实施例还提供一种词嵌入语言模型训练系统,包括:

[0017] 词表生成程序模块,用于确定语料库中的所有词语的属性以生成词表,所述属性包括所有词语的词性分类、所有词性分类的概率分布和所有词语在所属词性分类下的概率分布;

[0018] 词向量生成程序模块,用于生成所述词表中所有词语的词向量;

[0019] 词性分类向量生成程序模块,用于生成对应于所述词表中所有词语的词性分类的词性分类向量;

[0020] 模型训练程序模块,用于以所述词表中的词语的词向量和所述词表中的词语的词性分类向量为输入,以所述词表中的词语所属的词性分类的概率分布和所述词表中的词语在所属词性分类下的概率分布为输出进行训练,以得到所述词嵌入语言模型。

[0021] 第四方面,本发明实施例还提供一种词语识别系统,包括:

[0022] 词嵌入语言模型;

[0023] 词向量生成程序模块,用于生成待识别词语的词向量;

[0024] 词表生成程序模块,用于确定所述待识别词语的词性分类的词性分类向量;

[0025] 词语识别程序模块,用于将所述待识别词语的词向量和词性分类向量输入所述词嵌入语言模型,以获取所述待识别词语所属的词性分类的概率分布和所述待识别词语在所属词性分类下的概率分布。

[0026] 第五方面,本发明实施例提供一种非易失性计算机可读存储介质,所述存储介质中存储有一个或多个包括执行指令的程序,所述执行指令能够被电子设备(包括但不限于计算机,服务器,或者网络设备等)读取并执行,以用于执行本发明上述任一项词嵌入语言模型训练方法和/或词语识别法。

[0027] 第六方面,提供一种电子设备,其包括:至少一个处理器,以及与所述至少一个处理器通信连接的存储器,其中,所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行本发明上述任一项词嵌入语言模型训练方法和/或词语识别法。

[0028] 第七方面,本发明实施例还提供一种计算机程序产品,所述计算机程序产品包括存储在非易失性计算机可读存储介质上的计算机程序,所述计算机程序包括程序指令,当所述程序指令被计算机执行时,使所述计算机执行上述任一项词嵌入语言模型训练方法和/或词语识别法。

[0029] 本发明实施例的有益效果在于:本发明实施例中在进行语言模型的训练时并非直

接将语料库中的词语拿来直接用于训练,而是首先确定了所有词语的属性,包括所有词语的词性分类、所有词性分类的概率分布和所有词语在所属词性分类下的概率分布;并且在进行语言模型训练时综合考虑了词语的形态学信息和句法级别信息,尤其是句法级别信息的引入,综合考虑了属于同一词性分类的词语的共性,从而使得训练得到的语言模型在实际应用中,即便是遇到OOV词语,也能通过该OOV词语的形态学信息和所述词性分类的句法级信息来进行准确的识别。

附图说明

[0030] 为了更清楚地说明本发明实施例的技术方案,下面将对实施例描述中所需要使用的附图作一简单地介绍,显而易见地,下面描述中的附图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的前提下,还可以根据这些附图获得其他的附图。

[0031] 图1为本发明的词嵌入语言模型训练方法的一实施例的流程图;

[0032] 图2为本发明的词嵌入语言模型训练方法的另一实施例的流程图;

[0033] 图3为本发明的词语识别方法的一实施例的流程图;

[0034] 图4为本发明的词语识别方法的另一实施例的流程图;

[0035] 图5为本发明的词语识别方法的又一实施例的流程图;

[0036] 图6为本发明的词嵌入语言模型的一实施例的结构示意图;

[0037] 图7为本发明的词嵌入语言模型训练系统的一实施例的原理框图;

[0038] 图8为本发明的词嵌入语言模型训练系统的另一实施例的原理框图;

[0039] 图9为本发明的词语识别系统的一实施例的原理框图;

[0040] 图10为本发明的词语识别系统的另一实施例的原理框图;

[0041] 图11为本发明的词语识别系统的又一实施例的原理框图;

[0042] 图12为本发明的电子设备的一实施例的结构示意图。

具体实施方式

[0043] 为使本发明实施例的目的、技术方案和优点更加清楚,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作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。

[0044] 需要说明的是,在不冲突的情况下,本申请中的实施例及实施例中的特征可以相互组合。

[0045] 本发明可以在由计算机执行的计算机可执行指令的一般上下文中描述,例如程序模块。一般地,程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、元件、数据结构等等。也可以在分布式计算环境中实践本发明,在这些分布式计算环境中,通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中,程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

[0046] 在本发明中,“模块”、“装置”、“系统”等等指应用于计算机的相关实体,如硬件、硬件和软件的组合、软件或执行中的软件等。详细地说,例如,元件可以、但不限于是在运行于处

理器的过程、处理器、对象、可执行元件、执行线程、程序和/或计算机。还有,运行于服务器上的应用程序或脚本程序、服务器都可以是元件。一个或多个元件可在执行的过程和/或线程中,并且元件可以在一台计算机上本地化和/或分布在两台或多台计算机之间,并可以由各种计算机可读介质运行。元件还可以根据具有一个或多个数据包的信号,例如,来自一个与本地系统、分布式系统中另一元件交互的,和/或在因特网的网络通过信号与其它系统交互的数据的信号通过本地和/或远程过程来进行通信。

[0047] 最后,还需要说明的是,在本文中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”,不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

[0048] 如图1所示,本发明的实施例提供一种词嵌入语言模型训练方法,该方法包括:

[0049] S11、确定语料库中的所有词语的属性以生成词表,所述属性包括所有词语的词性分类、所有词性分类的概率分布和所有词语在所属词性分类下的概率分布。

[0050] 本实施例中的词表中按照词性分类存储来自语料库中的所有的词语,词性分类可以包括名词、形容词、动词、副词等,通过统计确定所有词性分类所包含的词语占来自语料库中的所有词语的比例,即确定了所有词性分类的概率分布。

[0051] 然后再进一步统计属于每个词性分类中的词语占该词性分类下所有词语的比例,即确定所有词语在所述词性分类下的概率分布。

[0052] S12、生成所述词表中所有词语的词向量。通过获取词语的词向量得到了词语的形态学信息。

[0053] S13、生成对应于所述词表中所有词语的词性分类的词性分类向量;通过确定词性分类向量的方式能够确定属于相应词性分类的词语的句法级别信息(即,语义信息),使得同属于一个词性分类的所有词语共享同一词性分类向量。

[0054] S14、以所述词表中的词语的词向量和所述词表中的词语的词性分类向量为输入,以所述词表中的词语所属的词性分类的概率分布和所述词表中的词语在所属词性分类下的概率分布为输出进行训练,以得到所述词嵌入语言模型。

[0055] 直接将这两个向量拼接起来当做输入,在输出层,采用一种因式分解的归一化指数函数来计算每个词的概率。首先计算每个语义分类的概率分布,再计算这些词性下词语的概率分布。最后将这个两个概率分布相乘就能得到所需的概率分布。

[0056] 本发明实施例中在进行语言模型的训练时并非直接将语料库中的词语拿来直接用于训练,而是首先确定了所有词语的属性,包括所有词语的词性分类、所有词性分类的概率分布和所有词语在所属词性分类下的概率分布;并且在进行语言模型训练时综合考虑了词语的形态学信息和句法级别信息,尤其是句法级别信息的引入,利用的词语在语义学上的一些共性,通过参数共享的方法,让OOV词也能利用已在词表中的词的参数,综合考虑了属于同一词性分类的词语的共性,从而使得训练得到的语言模型在实际应用中,即便是遇到OOV词语,也能通过该OOV词语的形态学信息和所述词性分类的句法级信息来进行准确的识别。

[0057] 此外,因为我们引入了额外信息的缘故(语义分类、形态学分解),使得我们的方法可以直接对未见过的新词进行建模,而且建模更为准确。然而在传统方法中,对新词建模需要收集新的数据然后重新训练模型,这非常耗时。因此在实际使用过程中,我们的模型可以大大节省加入新词的时间。这种方法可以进一步整合进语音识别系统中,来实现快速词表更新,同时在低频词的识别正确率上也有提升。

[0058] 如图2所示,在一些实施例中,所述生成所述词表中所有词语的词向量包括:

[0059] S21、判断获取自所述词表的词语是否为低频词;

[0060] S22、如果是,则将获取自所述词表的词语拆解成字,并对拆解得到的字进行编码以用于确定相应的词向量;

[0061] S23、如果不是,则提取所述获取自所述词表的词语的向量作为词向量。

[0062] 本发明实施例中,针对高频词和低频词采用不同的处理方法:对于高频词,每个词有其独立的词向量(例如,可以采用one-hot向量);针对低频词,首先将其在形态学上进行拆解(对于中文即拆解为字),然后再进一步采用一些序列编码的方法,将其转换为一个定长的向量,常用的编码方法有字符级固定大小常规遗忘编码(FOFE,Fixed-size Ordinally Forgetting Encoding)、直接将字向量相加、采用循环或者卷积神经网络进行编码等等。本发明实施例中,一个词语不单单看成一个one-hot向量,而是从句法层面进行分类(pos tag,part-of-speech tag)进行分类(即,one-hot的句法层面的表示),再从形态学层面,将一个进行FOFE code的编码。

[0063] 之所以对高频词和低频词进行区分,是因为不是所有词的含义都能用它的字来很好地表示,因此本发明实施例避免了上述情况下对语言模型性能的影响。

[0064] 此外,在传统语言模型中需要对每个词设置一组参数。而由于在本发明实施例的方法中,低频词都被拆解成了字(形态学分解),因此我们只需对所有的高频词和字设置参数,所带来的效果就是所需的参数量大大减少(通常可以减少80%左右),参数量少了带来的好处就是,本发明实施例所得到的词嵌入语言模型可以嵌入到一些更小型的设备(如手机)中去。

[0065] 还有词语的形态学分解一开始也可以用音素分解,但由于某些同音词的含义差距很大,因此音素分解的方法效果并不是很好,而通过采用本发明实施例的方法就克服了这些问题。

[0066] 在一些实施例中,所述以所述词表中的词语的词向量和所述词表中的词语的词性分类向量为输入,以所述词表中的词语所属的词性分类的概率分布和所述词表中的词语在所属词性分类下的概率分布为输出进行训练,以得到所述词嵌入语言模型包括:

[0067] 将所述词表中的词语的词向量和所述词表中的词语的词性分类向量输入至长短时记忆网络;

[0068] 将所述长短时记忆网络的输出输入至词性分类器以得到所述词表中的词语所属的词性分类的概率分布;

[0069] 将所述长短时记忆网络的输出输入至词分类器以得到所述词表中的词语在所属的词性分类下的概率分布。

[0070] 训练得到的词嵌入语言模型包括长短时记忆网络、词性分类器和词分类器。

[0071] 如图3所示,本发明实施例还提供一种词语识别方法,所述方法采用了本发明实施

例中所述的词嵌入语言模型,所述方法包括:

[0072] S31、生成待识别词语的词向量;

[0073] S32、确定所述待识别词语的词性分类的词性分类向量;

[0074] S33、将所述待识别词语的词向量和词性分类向量输入所述词嵌入语言模型,以获取所述待识别词语所属的词性分类的概率分布和所述待识别词语在所属词性分类下的概率分布。

[0075] 本发明实施例中所采用的语言模型在进行训练时并非直接将语料库中的词语拿来直接用于训练,而是首先确定了所有词语的属性,包括所有词语的词性分类、所有词性分类的概率分布和所有词语在所属词性分类下的概率分布;并且在进行语言模型训练时综合考虑了词语的形态学信息和句法级别信息,尤其是句法级别信息的引入,利用的词语在语义学上的一些共性,通过参数共享的方法,让OOV词也能利用已在词表中的词的参数,综合考虑了属于同一词性分类的词语的共性,从而使得训练得到的语言模型在实际应用中,即便是遇到OOV词语,也能通过该OOV词语的形态学信息和所述词性分类的句法级信息来进行准确的识别。

[0076] 此外,因为我们引入了额外信息的缘故(语义分类、形态学分解),使得我们的方法可以直接对未见过的新词进行建模,而且建模更为准确。然而在传统方法中,对新词建模需要收集新的数据然后重新训练模型,这非常耗时。因此在实际使用过程中,我们的模型可以大大节省加入新词的时间。这种方法可以进一步整合进语音识别系统中,来实现快速词表更新,同时在低频词的识别正确率上也有提升。

[0077] 如图4所示,在一些实施例中,当所述待识别词语属于用于训练所述词嵌入语言模型的词表时,所述生成所述待识别词语的词向量包括:

[0078] S41、判断所述待识别词语是否为低频词;

[0079] S42、如果是,则将所述待识别词语拆解成字,并对拆解得到的字进行编码以用于确定相应的词向量;

[0080] S43、如果不是,则提取所述待识别词语的向量作为词向量。

[0081] 本发明实施例中,针对高频词和低频词采用不同的处理方法:对于高频词,每个词有其独立的词向量(例如,可以采用one-hot向量);针对低频词,首先将其在形态学上进行拆解(对于中文即拆解为字),然后再进一步采用一些序列编码的方法,将其转换为一个定长的向量,常用的编码方法有字符级固定大小常规遗忘编码(FOFE,Fixed-size Ordinally Forgetting Encoding)、直接将字向量相加、采用循环或者卷积神经网络进行编码等等。本发明实施例中,一个词语不单单看成一个one-hot向量,而是从句法层面进行分类(pos tag,part-of-speech tag)进行分类(即,one-hot的句法层面的表示),再从形态学层面,将一个进行FOFE code的编码。

[0082] 之所以对高频词和低频词进行区分,是因为不是所有词的含义都能用它的字来很好地表示,因此本发明实施例避免了上述情况下对语言模型性能的影响。

[0083] 此外,在传统语言模型中需要对每个词设置一组参数。而由于在本发明实施例的方法中,低频词都被拆解成了字(形态学分解),因此我们只需对所有的高频词和字设置参数,所带来的效果就是所需的参数量大大减少(通常可以减少80%左右),参数量少了带来的好处就是,本发明实施例所得到的词嵌入语言模型可以嵌入到一些更小型的设备(如手

机)中去。

[0084] 如图5所示,在一些实施例中,当所述待识别词语不属于用于训练所述词嵌入语言模型的词表时,所述生成所述待识别词语的词向量包括:

[0085] S51、确定所述待识别词语的属性以更新所述词表;

[0086] S52、将所述待识别词语拆解成字,并对拆解得到的字进行编码以用于确定相应的词向量。

[0087] 本实施例中实现了在词嵌入语言模型中快速增加新词的方法。这种方法可以进一步整合进语音识别系统中,来实现快速词表更新,同时在低频词的识别正确率上也有提升。

[0088] 以下通过对比传统LSTM(Long Short-Term Memory)语言模型对OOV单词的处理与本发明实施例的技术方案来进一步描述本发明的实施例。

[0089] LSTM语言模型介绍:

[0090] 深度学习方法在语言模型中得到了广泛的应用,取得了很大的成功。长短期记忆(LSTM)网络是一种特别适用于序列的递归神经网络(RNN)体系结构。设 V 是词汇集,在每个时间戳 t 处,输入单词 w_t 由单热(one-hot)向量 e_t 表示,然后单词嵌入可以被获得为 x_t :

$$[0091] \quad x_t = E_i e_t \quad (1)$$

[0092] 其中 $E_i \in \mathbb{R}^{m \times |V|}$,是输入单词嵌入矩阵, m 表示输入单词嵌入的灰度。具体地说,LSTM的一个步骤将 x_t, h_{t-1}, c_{t-1} 作为输入,并产生 h_t, c_t 。计算细节在本文中省略。下一个单词的概率分布是在输出层上通过对隐含层进行仿射变换然后是softmax函数来计算的:

$$[0093] \quad P(w_{t+1} = j | w_{1:t}) = \frac{\exp(h_t^T E_o^j + b^j)}{\sum_{k \in V} \exp(h_t^T E_o^k + b^k)} \quad (2)$$

[0094] 其中, E_o^j 是 $E_o \in \mathbb{R}^{m \times |V|}$ 的第 j 列,也被称为输出嵌入,而 b^j 是偏置项。我们发现,输出层的偏置项起着重要的作用,与单词的频率高度相关。

[0095] 由于大部分计算成本取决于输出层,所以提出了因式分解的softmax输出层来提高语言模型的速度。这种方法是基于假设可以将单词映射到类。设 S 是类集。不同于等式(2),因式分解输出层的下一个单词的概率分布计算如下:

$$[0096] \quad P(w_{t+1} = j | w_{1:t}) = P(s_{t+1} = s_j | h_t) P(w_{t+1} = j | s_j, h_t) \quad (3)$$

$$[0097] \quad P(s_{t+1} = s_j | h_t) = \frac{\exp(h_t^T E_c^{s_j} + b_c^{s_j})}{\sum_{n \in S} \exp(h_t^T E_c^n + b_c^n)} \quad (4)$$

$$[0098] \quad P(w_{t+1} = j | s_j, h_t) = \frac{\exp(h_t^T E_o^j + b^j)}{\sum_{k \in V_{s_j}} \exp(h_t^T E_o^k + b^k)} \quad (5)$$

[0099] 其中, s_j 表示单词 w_{t+1} 的类, V_{s_j} 是属于类 s_j 的所有单词的集合。这里,单词的概率计算分为两个阶段:我们首先估计类的概率分布,然后从期望的类中计算特定单词的概率。其实一个单词可以属于多个类。但在本文中,每个单词映射到一个不同的类,即所有的类是互斥的。常用的词类是基于频率的类或者是从数据驱动的方法获得的类。

[0100] OOV单词处理

[0101] 如前所述,在经典的LSTM语言模型中已经使用了两种方法来处理OOV单词问题:

[0102] 1. 一类特殊的<UNK>用来取代所有OOV单词,使用另一种称为调整复杂度的测量:

$$[0103] \quad P(w_{\text{OOV}}) = \frac{P(\langle \text{unk} \rangle)}{|\mathcal{V}_{\text{OOV}}|} \quad (6)$$

[0104] 其中 \mathcal{V}_{OOV} 是所有OOV单词的词汇集合。实验中我们把这种方法称为“unk”。

[0105] 2. 用更新的词汇再训练模型。由于OOV单词在训练集中没有或很少有正例，训练后其概率将被分配到一个很小的值。这种方法可以类似于在n元语言模型中使用的平滑方法。我们在实验中将这种方法称为“再训练”。

[0106] 这两种传统方法都有其不足之处在于：在unk LSTM语言模型中，由于OOV单词的频率与训练数据和测试数据不匹配，因此误判OOV单词的概率。此外，这种方法忽略了OOV单词的语言信息。再训练的LSTM语言模型的主要问题是耗时。

[0107] 在传统的LSTM语言模型中，每个单词的单词嵌入是独立的，这就产生了两个问题。首先，新词不能用训练单词嵌入。第二，由于缺乏训练数据，这很少见。结构化单词嵌入的动机是使用参数共享来解决这两个问题。与数据驱动的方法不同，参数共享方法必须基于明确的规则。通过使用句法和形态规则，我们可以很容易地找到OOV单词的共享参数，并在我们的模型中构建自己的结构化单词嵌入。

[0108] 形态句法结构化嵌入：

[0109] 在句法层面上，每个单词都被分配给一个词类 (POS) 类。同一POS (part-of-speech) 类中的所有单词共享相同的POS类嵌入，称为句法嵌入。词类是一种具有类似语法特征的单词。因此，我们假设句法嵌入代表了单词的基本句法功能。

[0110] 对于每个单词，我们使用几个例句来标记它的POS标签，并选择最常见的标签作为词类 (POS标签也可从字典中获得)。词表内 (IV) 单词的例句从训练集中选择。对于OOV单词，例句可以由其他数据源组成或选自其他数据源，例如网络数据。与数据驱动的方法不同，基于POS标签的句法嵌入可以使用规则轻松地OOV单词生成。

[0111] 字符 (或子单词) 表示在许多NLP (Natural Language Processing) 任务中被广泛使用，作为提高低频单词的性能的一个附加特征，特别是在形态丰富的语言中。但对于高频单词来说，改进是有限的。本文为了进一步捕捉低频单词的语义，建立了形态嵌入。这是基于低频单词的数据稀疏性在字符级较不严重的假设。对于高频单词，保留单词嵌入。因此，混合嵌入，即，低频单词的形态嵌入和高频单词的单词嵌入应该处于同一维度。

[0112] 在以前的文献中，单词嵌入与子单词级特征相结合以获得所有单词的增强嵌入。相反，本文提出的低频单词的形态嵌入只依赖于字符级特征。因此，它有能力对OOV单词进行建模。

[0113] 所提出的形态嵌入利用字符级固定大小常规遗忘编码 (FOFE) 字符信息。在我们的模型中，所有的低频单词由字符序列 $e_1:T$ 表示，其中， e_t 是时间戳 t 处字符的单热 (one-hot) 表示。FOFE基于一个简单的递归公式 ($z_0=0$) 对整个序列进行编码：

$$[0114] \quad z_t = \alpha z_{t-1} + e_t \quad (1 \leq t \leq T) \quad (7)$$

[0115] 其中 $0 < \alpha < 1$ 是控制历史对最终时间步长影响的常数遗忘因子。另外，前馈神经网络 (FNN, Feedword Natural Network) 被用来将字符级的FOFE编码转换成最终的形态嵌入。

[0116] 将结构化嵌入与LSTM语言模型结合在一起：

[0117] 如图6所示，为本发明一实施例中的词嵌入语言模型的结构示意图。

[0118] 在输入层，输入单词的结构化嵌入是通过将其句法嵌入与单词嵌入 (针对高频单

词)或形态嵌入(针对低频单词)连结而获得的。

[0119] 在输出层,使用因式分解softmax结构是很容易的。公式(4)中的输出类嵌入矩阵 E_c 被句法嵌入代替,并且公式(5)中的输出嵌入矩阵 E_o 被单词和形态嵌入代替。

[0120] 一旦训练完成,OOV单词的句法和形态嵌入就可以随时获得。为了计算OOV单词的概率,我们需要重建公式(5)中的输出层参数: E_o, b 。 E_o 中的所有嵌入和偏置项 b 被保留给IV单词,并且 E_o 中的OOV单词的嵌入被其形态嵌入填充。在实验中,我们发现偏置项与词频高度相关,这意味着词频越高,偏置值越大。在本文中,我们使用OOV单词的偏置项作为一个经验的小恒定值。

[0121] 通过利用结构化单词嵌入,可以将OOV单词结合到LSTM语言模型中而不需要再训练。正如我们前面提到的,在训练过程中通过共享所提出的模型中的参数,也可以减轻OOV单词的数据稀疏性。

[0122] 本发明实施例中所提出的结构化单词嵌入语言模型还实现了参数压缩。在LSTM语言模型中,低频单词的单词嵌入占据了很大一部分模型参数但是是训练不够的。通过用字符表示代替低频单词,可以大大减少参数的数量。

[0123] 设 V 是所有单词的词汇集, H 是隐藏层大小。在LSTM语言模型中,单词嵌入的参数数量是 $2 \times |V| \times H$,然而,在结构化嵌入的LSTM语言模型中,参数总数为 $(|V_h| + |V_{char}| + |S|) \times H^3$,其中, V_h 表示高频单词, V_{char} 表示字符集, S 表示POS标签组。实验表明,当 $|V| = 60000, |V_h| = 8000, |V_{char}| = 5000, |S| = 32$ 时,可以减少近90%的参数。

[0124] 发明人在为了验证本发明的方法及系统能够达到预期的效果,基于中文短信息服务(SMS, short message service)数据集进行试验评估了本发明实施例中的词嵌入语言模型(后续统称结构化单词嵌入LSTM语言模型)。

[0125] 表1-数据集信息

数据集		#tokens	$ V_s $ $ V_d $	OOV rate on V_s
sms	训练	1.8M	33K 39K	0.8%
	验证	100K		1.0%
	测试	1.5K		0.8%
sms-30m	训练	5.6M	57K 67K	0.7%
	验证	166K		0.7%
	测试	112K		0.6%

[0126] 表1给出了数据集的详细信息。两个不同大小的词汇表用于每个数据集。完整的词汇集 V_f 涵盖出现在语料库中的所有词汇。小词汇集 V_s 是 V_f 的一个子集。这里词汇内(IV)被定义为 V_s 中的单词,词汇外(OOV)意味着 V_f 而不是 V_s 中的单词。sms-30m数据集也作为训练集和普通话自发对话测试集(约25小时,3K话语)用于ASR(自动语音识别)重新执行任务。

[0127] 1.用小词汇集 V_s 来训练LSTM语言模型,并且所有OOV单词被视为一个单独的<UNK>符号,称为“UNK”。

[0128] 2.用全部词汇集 V_f 再训练LSTM语言模型,称为“再训练”。

[0130] 对于具有结构化单词嵌入的LSTM语言模型,在训练阶段使用小词汇集 V_s ,在测试阶段将模型词汇更新为 V_f 。

[0131] 为了与所提出的模型的大小保持一致,LSTM基线的输入嵌入尺寸被设置为600,并且输出嵌入尺寸被设置为300。在具有结构化嵌入的LSTM语言模型中,句法嵌入尺寸被设置到300。我们使用1层5000-300FNN进行FOFE编码,其中5000是字符集 V_c 的大小,300是形态嵌入的维度。将FOFE的 α 设置为0.7,将新词的偏置项设置为0,将这两个经验参数在有效集中进行微调。最频繁的8192个单词被选为高频单词,其他单词在我们的模型中都被视为低频单词。所有模型都使用相同的超参数随机梯度下降(SGD)进行训练。

[0132] 复杂度评估

[0133] 混淆度评估结果见表2。特别是,对于“unk”LSTM,OOV单词的PPL计算由等式(6)代替。结果表明,所提出的结构嵌入(SE)方法与unk LSTM具有相似的性能。然而,再训练的LSTM表现更差。为了进一步调查,我们分别对每个模型分别进行词汇内(IV)和词汇外(OOV)单词的PPL计算。实验结果如表3所示。unk LSTM以牺牲OOV单词为代价,在IV单词中表现最好,因为它的OOV单词的PPL非常高。相对于unk LSTM,再训练的LSTM大大提高了OOV单词的PPL,并且在IV单词中有所降低。我们的方法进一步提高了在OOV单词中的PPL,在IV单词中具有类似的性能。

[0134] 表2-不同OOV结合方法之间的复杂度比较

数据集	OOV 结合:	#params	PPL
[0135] sms	unk	30M	117
	重训练	36M	121
	SE	4M	114
sms-30m	unk	51M	185
	重训练	60M	193
	SE	4M	185

[0136] 表3-词表内单词和词表外单词的复杂度分解

数据集	OOV 结合:	PPL	
		IV	OOV
[0137] sms	unk	109	4809646
	重训练	116	836125
	SE	110	548383
sms-30m	unk	175	10747568
	重训练	185	2764168
	SE	179	1023620

[0138] 快速的词汇更新在ASR

[0139] 在自动语音识别(ASR)系统中,回退n元模型被用作生成网格的语言模型,从中生成n-best列表。然后可以使用神经网络语言模型重新调整n-best列表以获得更好的性能。

通常，n元和神经网络语言模型共享相同的词汇。因此，当词汇更新时，n元和神经网络语言模型都需要再训练。与神经网络语言模型相比，n元语言模型的训练时间可以忽略不计。

[0140] 这个实验分为两个阶段。在第一阶段中，分别使用小词汇表示为unk LSTM和LSTM与SE来训练LSTM语言模型和具有结构化嵌入(SE, Structure embedding)的LSTM语言模型。用Vs训练的n元语言模型被用来生成n-best列表。然后我们使用unk LSTM进行n-best列表重打分。在第二阶段，词汇Vs扩大为更大的词汇量Vf。由于词汇改变了，我们需要再训练unk LSTM和n元模型。但是具有SE的LSTM的词汇没有重新训练就重建了。之后，再训练的LSTM和具有SE的LSTM被用来重新定义由新的n元模型生成的n-best列表。

[0141] 表4-词表内句子和词表外句子的字符错误率比较和分解

OOV 结合:	数据集		
	ALL	IVS	OOVS
[0142] unk	11.76	10.88	39.80
重训练	11.38	10.91	26.20
SE	11.18	10.74	25.07

[0143] 实验结果如表4所示。受益于词汇的扩展，重新训练的LSTM在所有句子中获得绝对0.38%的CER改进。所提出的具有结构化嵌入的LSTM模型(具有SE的LSTM)达到最佳性能。为了研究从所提出的模型中可以得到什么样的句子获得最大的收益，我们根据所有词是否出现在Vs中将重打分句子分成两类，称为词汇内句子(IVS)和词汇外句子(OOVS)。如表4所示，用Vs训练的unk LSTM对于词汇外句子有更高的CER，因为由Vs构建的n元不能产生这些OOV单词。通过扩大词汇量，再训练的LSTM对于词汇外的句子得到了CER的显著改善。与再训练的LSTM相比，所提出的模型在IV和OOV句子上都优于CER。而且，对于OOV句子的CER的改进(绝对为1.13%)显著地高于对IV句子的CER改进(绝对为0.13%)，这意味着具有SE的LSTM具有更好的为OOV单词建模的能力。请注意，通过利用所提出的结构化单词嵌入的LSTM语言模型，可以节省传统方法中模型再训练的时间，并取得较好的性能。

[0144] 需要说明的是，对于前述的各方法实施例，为了简单描述，故将其都表述为一系列的动作合并，但是本领域技术人员应该知悉，本发明并不受所描述的动作顺序的限制，因为依据本发明，某些步骤可以采用其他顺序或者同时进行。其次，本领域技术人员也应该知悉，说明书中所描述的实施例均属于优选实施例，所涉及的动作和模块并不一定是本发明所必须的。

[0145] 在上述实施例中，对各个实施例的描述都各有侧重，某个实施例中未详述的部分，可以参见其他实施例的相关描述。

[0146] 如图7所示，本发明的实施例还提供一种词嵌入语言模型训练系统700，包括：

[0147] 词表生成程序模块710，用于确定语料库中的所有词语的属性以生成词表，所述属性包括所有词语的词性分类、所有词性分类的概率分布和所有词语在所属词性分类下的概率分布；

[0148] 词向量生成程序模块720，用于生成所述词表中所有词语的词向量；

[0149] 词性分类向量生成程序模块730，用于生成对应于所述词表中所有词语的词性分类的词性分类向量；

[0150] 模型训练程序模块740,用于以所述词表中的词语的词向量和所述词表中的词语的词性分类向量为输入,以所述词表中的词语所属的词性分类的概率分布和所述词表中的词语在所属词性分类下的概率分布为输出进行训练,以得到所述词嵌入语言模型。

[0151] 如图8所示,在一些实施例中,所述词向量生成程序模块720包括:

[0152] 频词判断程序单元721,用于判断获取自所述词表的词语是否为低频词;

[0153] 第一词向量生成程序单元722,用于当判断获取自所述词表的词语为低频词时,将获取自所述词表的词语拆解成字,并对拆解得到的字进行编码以用于确定相应的词向量;

[0154] 第二词向量生成程序单元723,用于当判断获取自所述词表的词语为高频词时,提取所述获取自所述词表的词语的向量作为词向量。

[0155] 如图9所示,本发明实施例还提供一种词语识别系统900,包括:

[0156] 本发明上述实施例中所述的词嵌入语言模型910;

[0157] 词向量生成程序模块920,用于生成待识别词语的词向量;

[0158] 词表生成程序模块930,用于确定所述待识别词语的词性分类的词性分类向量;

[0159] 词语识别程序模块940,用于将所述待识别词语的词向量和词性分类向量输入所述词嵌入语言模型,以获取所述待识别词语所属的词性分类的概率分布和所述待识别词语在所属词性分类下的概率分布。

[0160] 如图10所示,在一些实施例中,当所述待识别词语属于用于训练所述词嵌入语言模型的词表时,所述词向量生成程序模块920包括:

[0161] 频词判断程序单元921,用于判断所述待识别词语是否为低频词;

[0162] 第一词向量生成程序单元922,用于当判断获取自所述词表的词语为低频词时,将所述待识别词语拆解成字,并对拆解得到的字进行编码以用于确定相应的词向量;

[0163] 第二词向量生成程序单元923,用于当判断获取自所述词表的词语为高频词时,提取所述待识别词语的向量作为词向量。

[0164] 如图11所示,在一些实施例中,当所述待识别词语不属于用于训练所述词嵌入语言模型的词表时,所述词向量生成程序模块920包括:

[0165] 词表更新程序单元921',用于确定所述待识别词语的属性以更新所述词表;

[0166] 词向量生成程序单元922',用于将所述待识别词语拆解成字,并对拆解得到的字进行编码以用于确定相应的词向量。

[0167] 在一些实施例中,本发明实施例提供一种非易失性计算机可读存储介质,所述存储介质中存储有一个或多个包括执行指令的程序,所述执行指令能够被电子设备(包括但不限于计算机,服务器,或者网络设备等)读取并执行,以用于执行本发明上述任一项词嵌入语言模型训练方法和/或词语识别方法。

[0168] 在一些实施例中,本发明实施例还提供一种计算机程序产品,所述计算机程序产品包括存储在非易失性计算机可读存储介质上的计算机程序,所述计算机程序包括程序指令,当所述程序指令被计算机执行时,使所述计算机执行上述任一项词嵌入语言模型训练方法和/或词语识别方法。

[0169] 在一些实施例中,本发明实施例还提供一种电子设备,其包括:至少一个处理器,以及与所述至少一个处理器通信连接的存储器,其中,所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能

够执行词嵌入语言模型训练方法和/或词语识别方法的步骤。

[0170] 在一些实施例中,本发明实施例还提供一种存储介质,其上存储有计算机程序,其特征在于,该程序被处理器执行词嵌入语言模型训练方法和/或词语识别方法的步骤。

[0171] 上述本发明实施例的实现构建语言模型的系统 and/或输入法系统可用于执行本发明实施例的实现构建语言模型的方法和/或输入法,并相应的达到上述本发明实施例的实现构建语言模型的方法和/或输入法所达到的技术效果,这里不再赘述。

[0172] 本发明实施例中可以通过硬件处理器(hardware processor)来实现相关功能模块。

[0173] 图12是本申请另一实施例提供的词嵌入语言模型训练方法和/或词语识别方法的电子设备的硬件结构示意图,如图12所示,该设备包括:

[0174] 一个或多个处理器1210以及存储器1220,图12中以一个处理器1210为例。

[0175] 执行实现词嵌入语言模型训练方法和/或词语识别方法的设备还可以包括:输入装置1230和输出装置1240。

[0176] 处理器1210、存储器1220、输入装置1230和输出装置1240可以通过总线或者其他方式连接,图12中以通过总线连接为例。

[0177] 存储器1220作为一种非易失性计算机可读存储介质,可用于存储非易失性软件程序、非易失性计算机可执行程序以及模块,如本申请实施例中的实现词嵌入语言模型训练方法和/或词语识别方法对应的程序指令/模块。处理器1210通过运行存储在存储器1220中的非易失性软件程序、指令以及模块,从而执行服务器的各种功能应用以及数据处理,即实现上述方法实施例实现词嵌入语言模型训练方法和/或词语识别方法。

[0178] 存储器1220可以包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需要的应用程序;存储数据区可存储根据实现词嵌入语言模型训练装置和/或词语识别装置的使用所创建的数据等。此外,存储器1220可以包括高速随机存取存储器,还可以包括非易失性存储器,例如至少一个磁盘存储器件、闪存器件、或其他非易失性固态存储器件。在一些实施例中,存储器1220可选包括相对于处理器1210远程设置的存储器,这些远程存储器可以通过网络连接至词嵌入语言模型训练装置和/或词语识别装置。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

[0179] 输入装置1230可接收输入的数字或字符信息,以及产生与词嵌入语言模型训练装置和/或词语识别装置的用户设置以及功能控制有关的信号。输出装置1240可包括显示屏等显示设备。

[0180] 所述一个或者多个模块存储在所述存储器1220中,当被所述一个或者多个处理器1210执行时,执行上述任意方法实施例中的词嵌入语言模型训练方法和/或词语识别方法。

[0181] 上述产品可执行本申请实施例所提供的方法,具备执行方法相应的功能模块和有益效果。未在本实施例中详尽描述的技术细节,可参见本申请实施例所提供的方法。

[0182] 本申请实施例的电子设备以多种形式存在,包括但不限于:

[0183] (1) 移动通信设备:这类设备的特点是具备移动通信功能,并且以提供话音、数据通信为主要目标。这类终端包括:智能手机(例如iPhone)、多媒体手机、功能性手机,以及低端手机等。

[0184] (2) 超移动个人计算机设备:这类设备属于个人计算机的范畴,有计算和处理功

能,一般也具备移动上网特性。这类终端包括:PDA、MID和UMPC设备等,例如iPad。

[0185] (3) 便携式娱乐设备:这类设备可以显示和播放多媒体内容。该类设备包括:音频、视频播放器(例如iPod),掌上游戏机,电子书,以及智能玩具和便携式车载导航设备。

[0186] (4) 服务器:提供计算服务的设备,服务器的构成包括处理器、硬盘、内存、系统总线等,服务器和通用的计算机架构类似,但是由于需要提供高可靠的服务,因此在处理能力、稳定性、可靠性、安全性、可扩展性、可管理性等方面要求较高。

[0187] (5) 其他具有数据交互功能的电子装置。

[0188] 以上所描述的装置实施例仅仅是示意性的,其中所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。

[0189] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到各实施方式可借助软件加通用硬件平台的方式来实现,当然也可以通过硬件。基于这样的理解,上述技术方案本质上或者说对相关技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品可以存储在计算机可读存储介质中,如ROM/RAM、磁碟、光盘等,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行各个实施例或者实施例的某些部分所述的方法。

[0190] 最后应说明的是:以上实施例仅用以说明本申请的技术方案,而非对其限制;尽管参照前述实施例对本申请进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围。

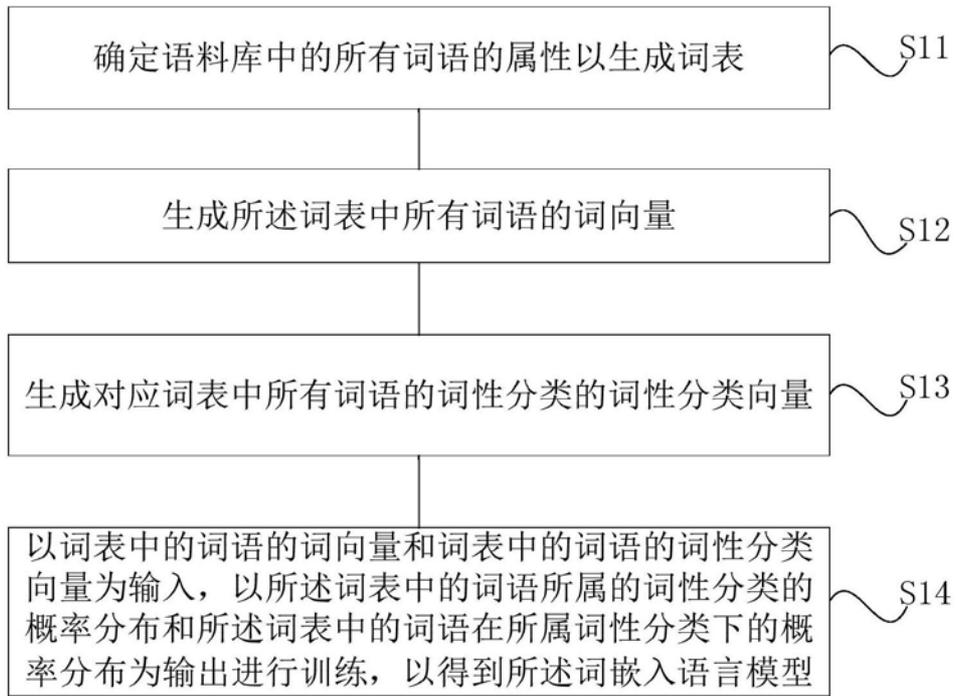


图1

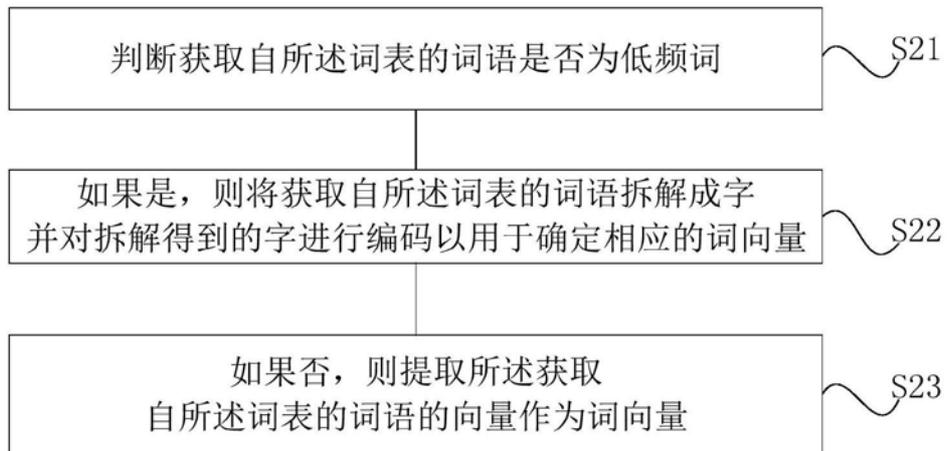


图2

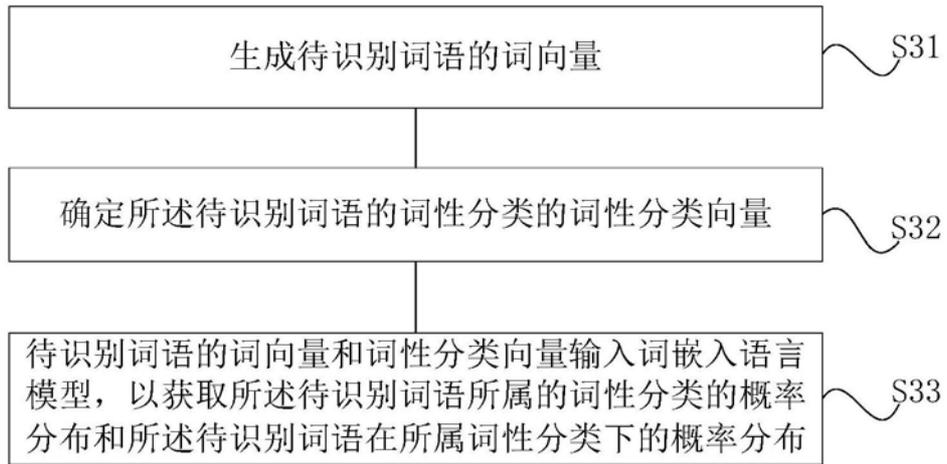


图3

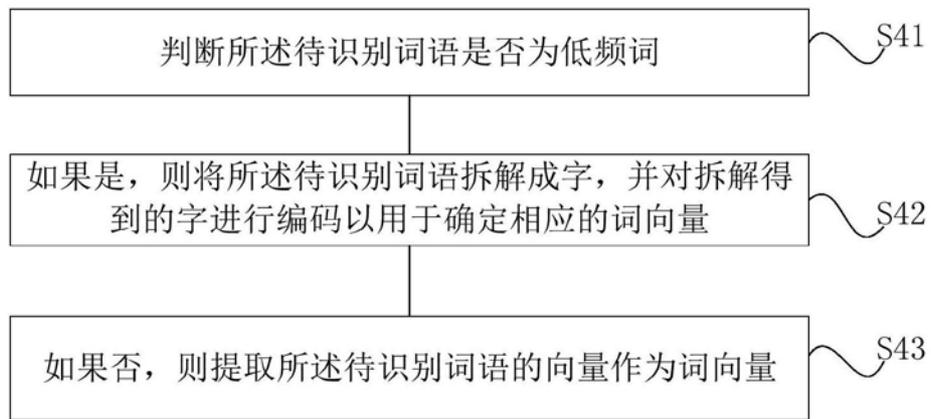


图4

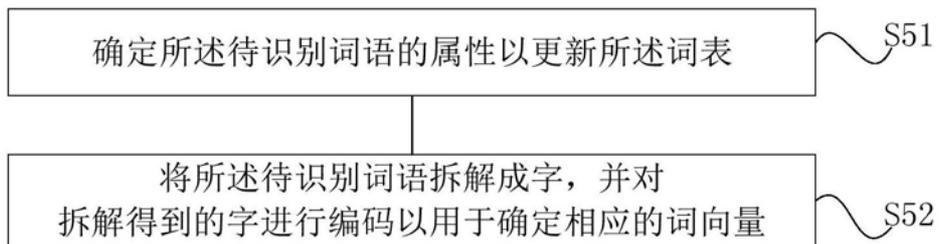


图5

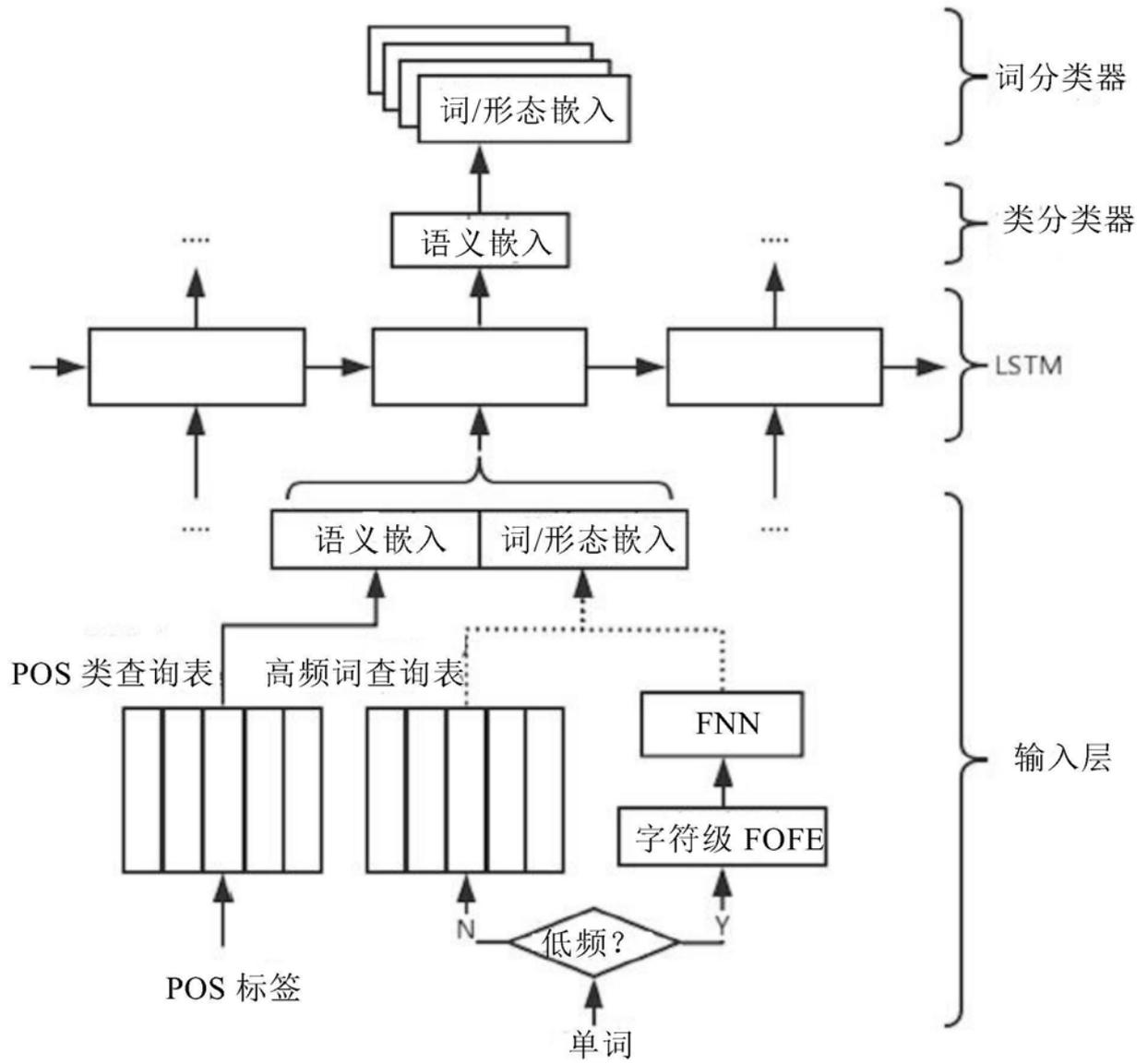


图6

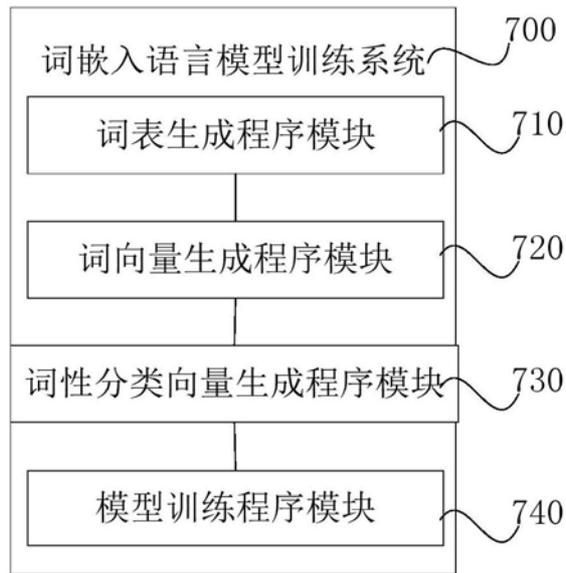


图7

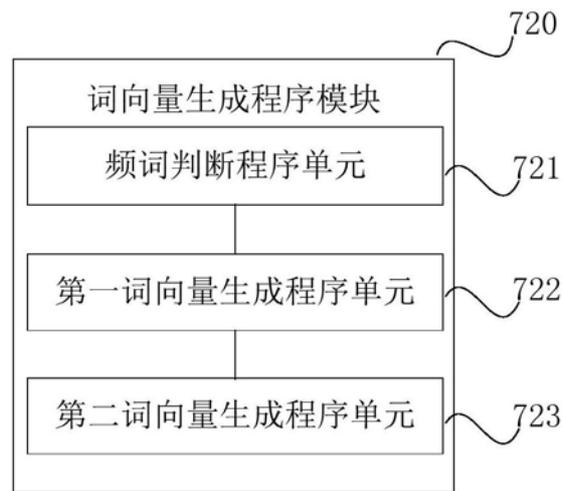


图8

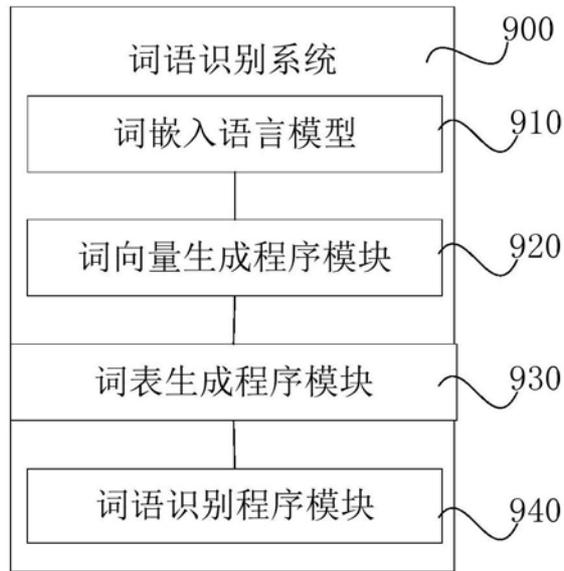


图9

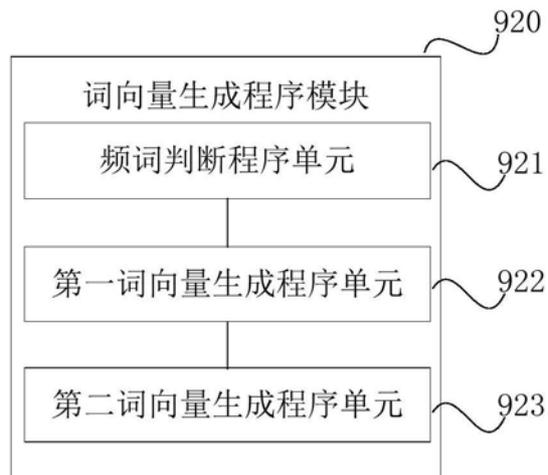


图10

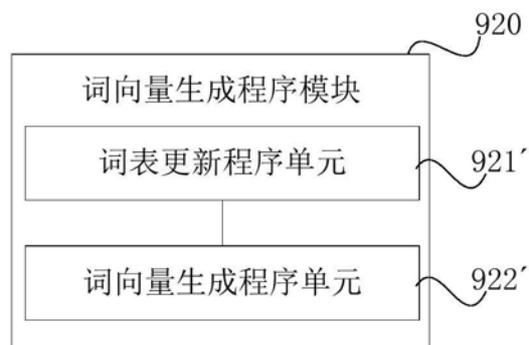


图11

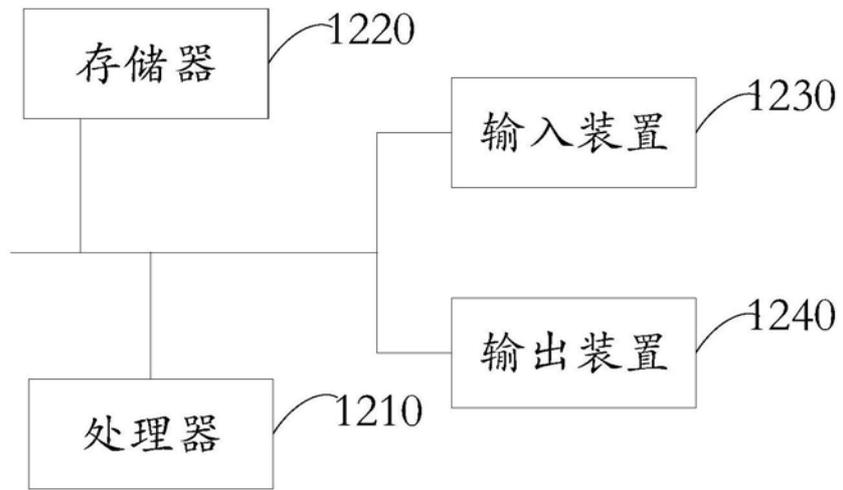


图12