(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 114911892 A
(43)申请公布日 2022.08.16

(21)申请号 202210118328.8
(22)申请日 2022.02.08
(30)优先权数据
17/169931 2021.02.08 US
(71)申请人 罗伯特·博世有限公司
地址 德国斯图加特
(72)发明人 L·博伊特索夫 J·柯尔特
(74)专利代理机构 中国专利代理(香港)有限公司 72001
专利代理师 任一方 刘春元
(51)Int.Cl.
G06F 16/31(2019.01)
G06F 16/33(2019.01)
G06N 3/02(2006.01)
G06N 3/04(2006.01)
G06N 3/08(2006.01)

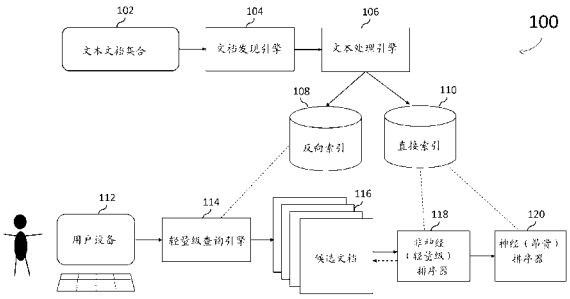
权利要求书3页 说明书16页 附图5页

(54)发明名称

用于搜索、检索和排序的交互层神经网络

(57)摘要

用于搜索、检索和排序的交互层神经网络。一种语言系统包括控制器。控制器可以被配置为接收查询和文档，将查询标记化为查询标记序列并将文档标记化为文档标记序列，为每个查询和文档标记生成标记对矩阵，为标记对矩阵中的每个条目检索由神经条件翻译概率网络产生的预计算的相似性得分，其中所述神经网络已经在使用成对查询和相应相关文档的语料库的排序任务中被训练，经由相应查询的每个相似性得分的和积聚合，产生每个文档相对于每个查询的排序得分；以及输出该文档和该文档的关联排序得分。



CN 114911892 A

1.一种语言系统,包括:

控制器,被配置为:

接收查询和文档;

将查询标记化为查询标记序列;

将文档中的每一个标记化为文档标记序列;

将查询标记转换为查询嵌入;

将文档标记转换为文档嵌入;

为每个查询和文档嵌入生成嵌入对矩阵;

经由神经条件翻译概率网络为嵌入对矩阵中的每个条目计算相似性得分,其中所述神经网络已经在使用成对查询和相应相关文档的语料库的排序任务中被训练;

经由相应查询的相似性得分的和积聚合产生每个文档相对于每个查询的排序得分;

根据文档的排序得分对文档定次序;和

输出所述文档和所述文档的关联排序得分。

2.根据权利要求1所述的语言系统,其中,所述神经网络已经被单语言训练。

3.根据权利要求1所述的语言系统,其中,所述查询嵌入或文档嵌入是使用Transformer模型产生的。

4.根据权利要求1所述的语言系统,其中,和积聚合是非上下文化的,并且被表述为

$$P(Q|D)=\prod_{q\in Q}\sum_{i=1}^{|D|}\frac{T(\mathrm{embed}_q(q)|\mathrm{embed}_d(d_i))}{|D|}$$

其中Q是查询,D是文档,q是查询标记,d是文档标记,$\boldsymbol{embed}_q\ (q)$是查询标记的查询嵌入,$\boldsymbol{embed}_d\ (d)$是文档标记$i$的文档嵌入,$T$是条件翻译,并且$P$是排序得分。

5.根据权利要求1所述的语言系统,其中,和积聚合是上下文化的,并且被表述为

$$P(Q|D)=\prod_{q\in Q}\sum_{i=1}^{|D|}\frac{T(\mathrm{embed}_q(q,d_i)|\mathrm{embed}_d(q,d_i))}{|D|}$$

其中Q是查询,D是文档,q是查询标记,d是文档标记,$\boldsymbol{embed}_q\ (q,d_i)$是查询标记的上下文化嵌入,$\boldsymbol{embed}_d\ (q,d_i)$是文档标记$d_1$的上下文化嵌入,$T$是条件翻译,并且$P$是排序得分。

6.根据权利要求1所述的语言系统,其中,所述控制器是多于一个的控制器。

7.根据权利要求1所述的语言系统,其中,所述查询是经由麦克风输入的。

8.根据权利要求7所述的语言系统,其中,所述输出是经由扬声器进行的。

9.一种语言系统,包括:

控制器,被配置为:

接收查询和文档;

将查询标记化为查询标记序列;

将文档标记化为文档标记序列;

为每个查询和文档标记生成标记对矩阵;

为标记对矩阵中的每个条目检索由神经条件翻译概率网络产生的预计算的相似性得

分，其中所述神经网络已经在使用成对查询和相应相关文档的语料库的排序任务中被训练；

经由相应查询的每个相似性得分的和积聚合，产生每个文档相对于每个查询的排序得分；和

输出所述文档和所述文档的关联排序得分。

10.根据权利要求9所述的语言系统，其中，和积聚合是

$$P(Q|D) = \prod_{q \in Q} \sum_{d \in D} T(q|d) P(d|D)$$

其中Q是查询，D是文档，q是查询标记，d是文档标记，$T(q|d)$是查询标记q与文档标记d对齐的预计算条件翻译概率，$P(Q|D)$是排序得分，并且$P(d|D)$是标记d在文档D中出现概率的最大似然估计。

11.根据权利要求9所述的语言系统，其中，所述神经网络是单语言训练的。

12.根据权利要求9所述的语言系统，其中，所述神经条件翻译概率网络将上下文无关嵌入用于查询标记或文档标记。

13.根据权利要求12所述的语言系统，其中，通过丢弃低于阈值的概率使密集查询文档标记翻译概率矩阵稀疏化，所述密集查询文档标记翻译概率矩阵由神经条件翻译概率网络产生。

14.根据权利要求12所述的语言系统，其中，所述查询经由麦克风输入，并且所述输出经由扬声器进行。

15.一种非暂时性计算机可读介质，包括用于神经条件翻译概率网络排序的指令，所述指令当由计算设备的处理器执行时，使得计算设备执行包括以下各项的操作：

接收查询和文档；

将查询标记化为查询标记序列；

对于每个文档将文档标记化为文档标记序列；

将查询标记转换为查询嵌入；

将文档标记转换为文档嵌入；

为每个查询和文档嵌入生成嵌入对矩阵；

经由神经条件翻译概率网络为嵌入对矩阵中的每个条目计算相似性得分，其中所述神经网络已经在使用成对查询和相应相关文档的语料库的排序任务中被训练；

经由相应查询的每个相似性得分的和积聚合，产生每个文档相对于每个查询的排序得分；和

根据文档的排序得分输出文档的次序。

16.根据权利要求15所述的介质，其中，所述神经条件翻译概率网络已经被单语言训练。

17.根据权利要求15所述的介质，其中，所述查询嵌入或文档嵌入是使用Transformer模型产生的。

18.根据权利要求15所述的介质，其中，和积聚合是非上下文化的，并且被表述为

$$P(Q|D)=\prod_{q\in Q}\sum_{i=1}^{|D|}\frac{T(\mathrm{embed}_q(q)|\mathrm{embed}_d(d_i))}{|D|}$$

其中Q是查询，D是文档，q是查询标记，d是文档标记，$embed_q(q)$是查询标记的查询嵌入，$embed_d(d)$是文档标记d_1的文档嵌入，T是条件翻译，并且P是排序得分。

19.根据权利要求15所述的介质，其中，和积聚合是上下文化的，并且被表述为

$$P(Q|D)=\prod_{q\in Q}\sum_{i=1}^{|D|}\frac{T(\mathrm{embed}_q(q,d_i)|\mathrm{embed}_d(q,d_i))}{|D|}$$

其中Q是查询，D是文档，q是查询标记，d是文档标记，$embed_q(q,d_i)$是查询标记的上下文化嵌入，$embed_d(q,d_i)$是文档标记d_1的上下文化嵌入，T是条件翻译，并且P是排序得分。

用于搜索、检索和排序的交互层神经网络

技术领域

[0001] 本申请通常涉及神经网络在多阶段检索系统中的使用。更具体地，本申请涉及用于信息检索系统中的文档检索和文档排序的神经网络的交互层的改进。

背景技术

[0002] 语言学是对语言的科学研究，并且涉及对语言形式、语言意义和语境中的语言的分析，以及对影响语言的社会、文化、历史和政治因素的分析。诸如互联网搜索系统的信息检索系统以大规模操作，从而索引数百万网页，使信息通过用户提交的查询而可搜索。这样的搜索系统的目标是快速标识与用户查询相关的那些文档，并返回相关的搜索结果。典型的文本检索系统依赖于简单的术语/标记匹配技术来生成候选的初始列表，该候选的初始列表可以使用学习的模型进一步重新排序。因此，查询和文档术语之间的不匹配对检索性能产生不利影响，这被称为词汇差距问题。在二十年前，Berger和Lafferty提出减小词汇差距，并且因此借助名为IBM模型1的词典翻译模型来改进检索效率。当应用于使用问题作为查询在英语问答(QA)档案中寻找答案以及跨语言检索时，IBM模型1具有很强的性能。在先前的工作中，模型1是在相似长度的问题文档对上训练的，这简化了在查询术语和相关文档中的术语之间寻找有用关联的任务。此外，已经主要使用产生非参数模型的传统期望最大化(EM)算法的变体在翻译任务中训练这些模型。

发明内容

[0003] 一种语言系统包括控制器。控制器可以被配置为接收查询和文档，将查询标记化为查询标记序列并将文档中的每一个标记化为文档标记序列，将查询标记转换为查询嵌入并将文档标记转换为文档嵌入，为每个查询和文档嵌入生成嵌入对矩阵，经由神经条件翻译概率网络为嵌入对矩阵中的每个条目计算相似性得分，其中所述神经网络已经在使用成对查询和相应相关文档的语料库的排序任务中被训练，经由相应查询的相似性得分的和积聚合产生每个文档相对于每个查询的排序得分，根据文档的排序得分对文档定次序；以及输出该文档和该文档的关联排序得分。

[0004] 一种语言系统包括控制器。控制器可以被配置为接收查询和文档，将查询标记化为查询标记序列并将文档标记化为文档标记序列，为每个查询和文档标记生成标记对矩阵，为标记对矩阵中的每个条目检索由神经条件翻译概率网络产生的预计算的相似性得分，其中所述神经网络已经在使用成对查询和相应相关文档的语料库的排序任务中被训练，经由相应查询的每个相似性得分的和积聚合，产生每个文档相对于每个查询的排序得分；以及输出该文档和该文档的关联排序得分。

[0005] 一种非暂时性计算机可读介质，包括用于神经条件翻译概率网络排序的指令，该指令当由计算设备的处理器执行时，使得计算设备执行操作。操作包括接收查询和文档，将查询标记化为查询标记序列，并且对于每个文档将文档标记化为文档标记序列，将查询标记转换为查询嵌入并将文档标记转换为文档嵌入，为每个查询和文档嵌入生成嵌入对矩

阵，经由神经条件翻译概率网络为嵌入对矩阵中的每个条目计算相似性得分，其中所述神经网络已经在使用成对查询和相应相关文档的语料库的排序任务中被训练，经由相应查询的每个相似性得分的和积聚合，产生每个文档相对于每个查询的排序得分，以及根据文档的排序得分输出文档的次序。

附图说明

[0006] 图1是搜索/检索系统的框图。

[0007] 图2是现有技术排序神经网络的流程图。

[0008] 图3是具有神经条件翻译概率网络和和积聚合的排序神经网络的流程图。

[0009] 图4是神经条件翻译概率的网络稀疏化过程的流程图，其中使用了上下文无关的标记嵌入网络。

[0010] 图5是电子计算系统的框图。

具体实施方式

[0011] 根据需要，本文中描述了本公开的实施例。然而，将理解，公开的实施例仅仅是本发明的示例，本发明可以以各种形式和替代形式体现。各图不一定是按比例的；一些特征可以被放大或最小化以示出特定组件的细节。因此，本文中公开的具体结构和功能细节不应被解释为限制性的，而仅仅是作为用于教导本领域技术人员以各种方式采用实施例的代表性基础。

[0012] 术语“基本上”在本文中可以用来描述公开或要求保护的实施例。术语“基本上”可以修饰本公开中公开或要求保护的值或相对特性。在这样的情况下，“基本上”可以指定它修饰的值或相对特性在该值或相对特性的±0%、0.1%、0.5%、1%、2%、3%、4%、5%或10%之内。

[0013] 首先考虑词典翻译模型(例如IBM模型1)对英语文本检索的效用，特别是它的端到端训练的神经变体。然后使用IBM模型1的神经变体作为交互层，应用于以上下文无关或上下文化(contextualized)的方式产生的查询/文档嵌入序列。该设计神经排序系统的新方法对于有效性、效率和可解释性有益处，这是至关重要的。具体而言，本公开说明了将神经模型1层添加到来自基于Transformer(BERT)的上下文化嵌入的双向编码器表示的顶部(1)不降低准确性和/或效率；和(2)可以克服现有BERT模型对最大序列长度的限制。词典(即上下文无关)的神经模型1与基于BERT的排序模型相比不太有效，但它可以在控制器、处理器、CPU、GPU或其他计算或逻辑电路上高效运行(无需昂贵的索引时间预处理)。

[0014] 典型的文本检索系统依赖于简单的术语匹配技术来生成候选的初始列表，该候选的初始列表可以使用学习的模型进一步重新排序。因此，查询和文档术语之间的不匹配对检索性能产生不利影响，这被称为词汇差距问题。在二十年前，提出了一种方法来减小词汇差距，并且因此借助名为IBM模型1(也称为模型1)的词典翻译模型来改进检索效率。当应用于使用问题作为查询在英语问答(QA)档案中寻找答案以及跨语言检索时，模型1具有很强的性能。然而，人们对它在实际的单语英语查询上的有效性知之甚少，部分因为训练模型1需要先前不公开可用的大量的查询集。

[0015] 模型1可以在相似长度的问题文档对上训练，这简化了在查询术语和相关文档中的术语之间寻找有用关联的任务。然而，如果查询比对应的相关文档短得多，例如短两个数

量级，则考虑训练模型1。此外，已经主要使用产生非参数模型的传统期望最大化(EM)算法的变体在翻译任务中训练这些模型。

[0016] 本公开通过用神经网络对条件翻译概率进行参数化并在排序(而不是翻译)任务中对模型进行端到端学习而呈现了更好的选择。

[0017] 对两个最近的MS MARCO集合执行了词典翻译模型的进一步研究，所述两个最近的MS MARCO集合具有数十万个真实用户查询。具体而言，考虑排序模型的新奇类，其中可解释的神经模型1层被应用于标记嵌入神经网络的输出。该层产生在查询和文档BERT单词片段之间的所有逐对相似性$T(q|d)$，所述单词片段在没有任何学习的权重的情况下经由简单的和积公式进行组合：

$$P(Q|D) = \prod_{q \in Q} \sum_{d \in D} T(q|d) P(d|D)$$

其中Q是查询，D是文档，q是查询标记，d是文档标记，$T(q|d)$是查询标记q的条件翻译概率(与文档标记d对齐的概率)，$P(Q|D)$是排序得分(解释为在给定文档的情况下查询的条件概率)，并且$P(d|D)$是标记d在文档D中出现概率的最大似然估计。

[0018] 排序得分$P(Q|D)$是个体查询单词片段得分的积，这使得查明(pinpoint)具有最大贡献的单词片段变得容易。同样，对于每个查询单词片段，可能的是标识对其得分具有最高贡献的文档单词片段。这使得所公开的模型与先前的工作相比更可解释。

[0019] 结果得到的复合网络(包括标记嵌入)是使用排序目标端到端学习的。考虑两个场景：上下文无关的标记嵌入和BERT生成的上下文化的标记嵌入。注意，本公开中呈现的方法是通用的，并且也可以应用于其他嵌入网络。

[0020] 该工作的发现可以总结如下：

首先，在BERT顶部添加可解释的神经模型1层使得与不容易可解释的vanilla BERT排序器相比在准确性和效率上几乎没有损失成为需要。事实上，对于长文档，基于BERT的可解释神经模型1可能胜过应用于截断文档的基线模型，因此克服了现有预处理Transformer模型对最大序列长度的限制。

[0021] 非参数模型1与BM25得分的融合可以胜过基线融合模型，但是增益非常小(≈3%)。在这种情况下，上下文无关的神经模型1可以比其非参数变体有效得多。

[0022] 上下文无关的模型1可以稀疏化，导出为非参数格式，并在CPU上执行，与在GPU上运行基于BERT的排序器相比速度快10^3倍以上。因此，它可以改进候选生成阶段，而不需要诸如doc2query之类的昂贵的索引时间预计算方法。

[0023] 统计翻译方法在文本检索中的使用被成功地应用于将问题用作查询在QA档案中寻找答案。在这种情况下，该模型在与用户提供的答案成对的大的单语问题集合上、即在平行的单语语料库上训练。模型1对于跨语言检索也很有用。

[0024] 模型1是非参数翻译模型，其中词位(lexeme)(或标记)的上下文无关翻译概率使用来自并行语料库的期望最大化(EM)算法来学习。改进非参数统计学习模型性能的通用方法在于使用神经网络对相应概率参数化。该思想在语言处理中的早期成功实现是用于语音识别的混合HMM DNN/RNN(隐马尔可夫模型深度神经网络/递归神经网络)系统。更具体地，本公开使用神经模型1作为最后的网络层，其基于LSTM-CRF(长短期记忆-条件随机场)和

CEDR(文档排序的上下文化嵌入)架构来计算查询文档似然性。

[0025] 考虑跨语言检索的上下文相关词典神经翻译模型。该方法在词典翻译任务中使用来自双语平行语料库的上下文相关翻译概率。给定文档,最高翻译概率连同相应的标记被提前预计算并存储在索引中。人们将需要在相似长度的对齐语句上训练模型。然而,在单语检索的情况下,本公开没有如此细粒度的训练数据,因为查询仅与长得多的相关文档成对。

[0026] 神经排序模型是近年来的热门话题。然而,在BERT之前的早期方法的成功是有争议的。随着采用以自我监督方式预处理的大神经模型,尤其是在引入Transformer模型和发布BERT之后,情形改变。人们认为基于BERT的模型显著胜过所有其他方法。BERT及其变体也在MS MARCO排行榜上占据主导地位。

[0027] 在Transformer之前,LSTM(长短期记忆)和其他递归神经网络是序列数据分类和翻译的主要方法。Transformer对序列长度没有固有的限制。然而,最近提出的Transformer模型采用了关注机制,其中每个序列位置可以关注先前层中的所有位置。因为自关注复杂度相对于序列长度是二次的,所以Transformer模型(包括BERT)仅支持有限长度的输入。存在多个建议来缓解该约束,例如,Conformer-kernel排序模型使用了所谓的可分离关注。这条研究路线是该工作的补充。

[0028] 为了用现有的预训练模型处理较长的文档,系统将必须将文档拆分成几个块,分别处理每个块,并通过例如计算最大值或加权预测得分来聚合结果。这样的模型不能在完整的文档上进行端到端训练。此外,训练过程必须假设相关文档中的每个块也是相关的,这不总是准确的。为了改进简单的聚合方法,引入了使用三个更简单模型的几个文档块的组合输出:KNRM(基于核的神经排序模型)、PACRR(PACRR:用于相关性匹配的位置感知神经IR模型)和DRMM(深度相关性匹配模型)。然而,没有一个聚合器模型是可解释的。

[0029] 统计模型的解释和说明已经成为繁忙的研究领域。然而,绝大多数方法依赖于训练单独的解释模型或利用显著性/注意力图。这是有问题的,因为由无关模型提供的解释难以验证和信任。此外,显著性和注意力图揭示了模型正在处理哪些数据部分,而不是模型如何处理它们。一种方法提倡其计算通过设计是透明的网络,而不是构建不可靠的事后解释。当完全透明的网络不可行时,仍然存在最后一层可解释性的益处。在文本检索中,考虑使用基于核的公式来计算通过上下文化嵌入的软匹配计数。因为每对查询文档标记产生对应于不同阈值的几个软匹配值,所以以可解释的方式聚合这些值是有问题的。尽管该方法确实提供了对模型决策的见解,但是聚合公式相当复杂。它本质上是两层神经网络,其中在第一层之后有非线性(对数)激活函数。相比之下,使用可解释的模型1层为每对查询和文档子词标记产生相似性得分。这些得分使用明确的和积公式进行组合,该明确的和积公式没有任何学习的权重(参见等式1)。

[0030] 通过将一些计算推迟到索引时间来加速查询时间排序的方法可以划分为两组。第一组包括生成部分解开的重新呈现,其可以在查询时快速组合。这些方法使很少到无性能降级成为需要。第二组方法产生或增强了独立的查询和文档表示,其可以通过计算内积进行比较。尽管代价是昂贵的索引处理和效率上的一些损失,表示——密集的或稀疏的——被示出改进了第一阶段的检索。

[0031] 在稀疏表示中,DeepCT(深度上下文化术语加权)使用BERT为文档和查询术语二者生成重要性权重,doc2query学习用最可能的查询术语来扩充文档,而Epic模型将文档扩展

与术语重新加权相组合。这些模型与vanilla BERT排序器相比不太有效，并且需要昂贵的索引时间处理。凭借术语扩展和重新加权模型生成的数据的稀疏性，它可以存储在传统的倒排文件中，以改进第一检索阶段的性能。尽管难以解释模型的扩展和重新加权决策，因为第一检索阶段通常采用诸如BM25的透明的排序公式，但可以很容易确定哪些术语匹配对查询文档得分最有贡献。这使得扩展和重新加权模型部分可解释。

[0032] 转到标记嵌入、Vanilla BERT和CEDR，假设输入文本被拆分成名为标记的小块文本。该过程名为标记化。标记可以是完整的英语单词、单词片段或词位(词元)。文档d的长度以标记的数量来度量，并且用$|d|$标示。因为神经网络不能直接对文本数据进行操作，所以标记序列$t_1t_2 \ldots t_n$首先通过嵌入子网络被转换成d维向量的序列$w_1w_2 \ldots w_n$。这些向量也被称为嵌入。最初，嵌入子网络是上下文无关的，即每个标记总是映射到相同向量。最近证明了上下文相关的、即上下文化的嵌入的优越性，其以自监督的方式在大语料库上产生多层双向LSTM预处理。这些不久被大的预训练Transformer超越。

[0033] 本公开包括两个类型的嵌入：vanilla上下文无关嵌入和基于BERT的上下文化嵌入。关于BERT架构，要考虑的几个重要事项是：

上下文化标记嵌入是最后一层隐藏状态的值；

BERT对单词片段而不是完整的单词进行操作；

词汇表具有接近30K个标记，并且包括两个特殊符号[CLS](聚合器)和[SEP](分隔符)；和

[CLS]总是在每个标记序列之前，并且它的嵌入被用作分类和排序任务的序列表示。

[0034] “vanilla”BERT排序器使用单个全连接层作为预测头，其将[CLS]向量转换为标量。它基于以下标记序列做出预测：$[\mathrm{CLS}]\ q\ [\mathrm{SEP}]\ d\ [\mathrm{SEP}]$，其中$q$是查询，并且$d = t_1t_2 \ldots t_n$是文档。长文档和查询需要被截断，使得标记的总体数量不超过512。要克服该限制，考虑一个方法，该方法：

将较长的文档d拆分成几个块：$d = d_1d_2 \ldots d_m$；

生成m个标记序列$[\mathrm{CLS}]\ q\ [\mathrm{SEP}]\ d_i\ [\mathrm{SEP}]$；和

用BERT处理每个序列，为常规标记以及为[CLS]生成上下文化嵌入。

[0035] 此结果是m个[CLS]-向量cls_i和n个上下文化向量$w_1w_2 \ldots w_n$：每个文档标记t_i一个。存在几个方法将这些上下文化向量组合。

[0036] 一个方法是通过对平均[CLS]标记做出预测来扩展vanilla BERT排序器：$\frac{1}{m}\sum_{i=1}^{m} cls_i$。

[0037] 第二个方法是在以下神经架构中使用上下文化嵌入作为上下文无关嵌入的直接替换：KNRM、PACRR和DRMM。

[0038] 在引入CEDR架构的第三个方法中，[CLS]标记/向量以特定于模型的方式被附加地并入KNRM、PACCR和DRMM中，这进一步提升了性能。

[0039] 非参数和神经模型1。包括将检索重铸为翻译问题的建议，其具有学习软匹配函数的目标，该软匹配函数为相关但不同的术语(例如同义词)分配非零权重。为此，考虑采用简

单的模型1,它是独立于术语且上下文无关的模型。

[0040] 让$P(D|q)$标示文档D与查询q相关的概率。

[0041] 使用贝叶斯规则,$P(D|q)$方便地改写为$P(D|q) \propto P(q|D)p(D)$。此外,假设文档出现概率$p(D)$具有均匀的先验,则相关性概率与$P(q|D)$成比例,$P(q|D)$进而可以作为文档到查询的翻译来计算。

[0042] 让$T(q|d)$是查询标记q是文档标记d的翻译的概率,并且$P(d|D)$是标记d由文档D"生成"的概率。然后,查询Q是文档D的翻译的概率可以如下计算为个体查询术语似然性的积:

$$P(Q|D) = \prod_{q \in Q} P(q|D)$$
$$P(q|D) = \sum_{d \in D} T(q|d)P(d|D) \quad (1)。$$

[0043] 使模型1有效需要多个技术,所述多个技术在非参数变量和神经变量之间有很大不同。对于非参数模型1,$P(q|D)$——查询术语q的似然性——使用参数λ与集合概率$P(q|C)$线性组合:

$$P(q|D) = (1-\lambda)\left[\sum_{d \in D} T(q|d)P(d|D)\right] + \lambda P(q|C), \quad (2)。$$

[0044] 集合概率$P(q|C)$和文档内术语概率$P(d|D)$二者是最大似然估计,即归一化出现频率。对于词汇外术语q,$P(q|C)$设置为小数字(例如10^{-9})。翻译概率$T(q|d)$使用在MGIZA中实现的EM算法来计算。MGIZA建模虚假插入(即从空单词翻译而来),但这可以被丢弃。接下来考虑几个附加的措施来改进模型1有效性:

通过将文档和段落拆分成其长度与查询长度相当的连续小块来创建并行语料库;

条件翻译概率$T(q|d)$是从对称化语料库学习的;

丢弃在经验发现的阈值约为10^{-3}以下的所有翻译概率$T(q|a)$,并且最多保留10^6个最频繁的标记;和

将自翻译概率$T(t|t)$设置为经验发现的正值,并重新缩放概率$T(t_0|t)$,使得$\sum_{t_0} T(t_0|t) = 1$。

[0045] 在神经翻译模型的情况下,考虑更简单的等式1,其不平滑翻译和集合概率。翻译概率$T(q|d)$由简单的神经网络计算,它的输入是标记q和d的上下文无关或上下文化的嵌入。对于上下文无关嵌入,$P(d|D)$等于文档中术语d的出现次数除以$|D|$。然而,标记嵌入的上下文化产生了本质上不重复的标记。因此,我们针对基于BERT的嵌入设置了$P(d|D) = 1/|D|$。

$$P(Q|D) = \prod_{q \in Q} \sum_{i=1}^{|D|} \frac{T(\text{embed}_q(q)|\text{embed}_d(d_i))}{|D|} \quad (3)$$

[0046] 其中Q是查询，D是文档，q是查询标记，d是文档标记，$\mathrm{embed}_q(q)$是查询标记的嵌入(向量)，$\mathrm{embed}_d(d_i)$是文档标记d_i的嵌入(向量)，T是条件翻译概率，并且$P(Q|D)$是排序得分，其被解释为给定文档D的查询Q的条件概率。

[0047] 同样，在上下文化嵌入设置中，

$$P(Q|D) = \prod_{q \in Q} \sum_{i=1}^{|D|} \frac{T(\mathrm{embed}_q(q,d_i)|\mathrm{embed}_d(q,d_i))}{|D|} \tag{4}$$

其中Q是查询，D是文档，q是查询标记，d是文档标记，$\mathrm{embed}_q(q,d_i)$是查询标记的上下文化嵌入，$\mathrm{embed}_d(q,d_i)$是文档标记d_i的上下文化嵌入，T是条件翻译，并且P是排序得分。

[0048] 注意，上下文无关的神经模型1可以被稀疏化并导出为非参数格式，这准许在CPU上极其高效的执行。这是通过预计算所有词汇标记对的$T(t_1|t_2)$并丢弃所有小于阈值的值来完成的。

[0049] 通过简单且高效的神经网络进一步计算等式3中的$T(q|d)$。网络“消耗”标记q和d的上下文无关或上下文化嵌入，并产生在范围$[0,1]$中的值。为了并入自翻译概率——这对于上下文无关模型的良好收敛性至关重要——我们设置$T(t|t) = p_{self}$，并将所有其他概率乘以$1-p_{self}$。然而，缩放条件概率以确保$\forall t_2 \sum_{t_1} T(t_1|t_2) = 1$是不实际的。因此，$T(t_1|t_2)$是相似性函数，但不是真实的概率分布。

[0050] 可以示出，使用小的自翻译概率$p_{self} = 0.05$极大改进了上下文无关模型的收敛性。在这种情况下，对条件概率进行适当缩放，使得对于任何标记t_2，我们具有$\sum_{t_1} T(t_1|t_2) = 1$，对于上下文无关的嵌入是低效的，并且甚至对于上下文化的嵌入是无知觉的(因为相同标记的嵌入向量在不同的上下文中是不同的)。因此，不对条件概率归一化，而是将神经网络的输出$T(t_1|t_2)$约束在范围$[0,1]$中。此外，设置$T(t|t) = p_{self}$并将所有其他概率乘以$1 - p_{self}$。

[0051] 考虑了为$T(t_1|t_2)$设计神经参数化的几个方法。考虑使用$\mathrm{embed}_q(t_1)$和$\mathrm{embed}_d(t_2)$分别标示查询和文档标记的嵌入。也许，最简单的方法是学习查询和文档的单独嵌入集，并将概率计算为缩放的余弦相似度：$T(t_1|t_2) = 0.5\{\cos(\mathrm{embed}_q(t_1), \mathrm{embed}_d(t_2)) + 1\}$。然而，该神经网络不具有足够的表达能力，并且结果得到的上下文无关模型1劣于经由EM学习的非参数模型1。

[0052] 可以示出，良好性能的关键因素是嵌入与其哈达玛积的连结，我们认为这有助于下面的层发现更好的交互特征。然后让它通过一个或多个全连接的线性层，其中RELU后面是sigmoid：

$$T(q|d) = \sigma(F_3(\text{relu}(F_2(\text{relu}(F_1([x_q, x_d, x_q \circ x_d])))))$$
$$x_q = P_q(\tanh(\text{layer-norm}(\text{embed}_q(q))))$$
$$x_d = P_d(\tanh(\text{layer-norm}(\text{embed}_d(d)))),$$

其中P_q、P_d和F_i是全连接的线性投影层；$[x, y]$是向量连结；并且层范数是层归一化。$x \circ y$是向量x和y的哈达玛积。

	文档	段落
	综合统计	
文档的#	3.2M	8.8M
文档词元的#	476.7	30.6
查询词元的#	3.2	3.5
	查询的#	
训练/融合	10K	20K
训练/建模	357K	788.7K
开发	2500	20K
测试	2693	3000
TREC 2019	100	100
TREC 2020	100	100

表1 MS MARCO数据集细节

[0053] 数据集：当用MS MARCO集合实验时，它包括用于段落和文档检索任务的数据。每个MS MARCO集合具有大量的真实用户查询(见表1)。在这方面，不存在其他集合可与MS MARCO相当。大的查询集是从搜索引擎必应的日志采样的。在这种情况下，数据集创建者确保所有的查询都可以使用短的文本片段来回答。这些查询仅被稀疏地判断(每个查询大约一个相关段落)。稀疏判断是二元的：相关文档具有等级一，并且所有其他文档具有等级零。

[0054] 除了具有稀疏判断的大的查询集之外，还使用了来自TREC 2019/2020深度学习轨迹的两个评估集。这些查询集非常小，但是它们已经由NIST评估员分别针对文档和段落检索任务进行了彻底判断。TREC NIST判断范围从零(不相关)到三(完全相关)。

[0055] 公开可用的训练和验证集被随机拆分成以下子集：训练线性融合模型的小训练集(训练/融合)、训练神经模型和非参数模型1的大集合(训练/建模)、开发集(开发)和最多包含3K查询的测试集(测试)。表1中总结了详细的数据集统计。注意，训练子集是从原始训练集获得的，而新的开发和测试集是从原始开发集获得的。

[0056] 接下来是使用Spacy 2.2.3处理集合，从文本提取标记和词元(词位)。频繁出现的标记和词元是使用Indri的停止词列表过滤掉的，该Indri的停止词列表被扩展为包括一些缩写，诸如“n't”和“ll”。使用Lucene 7.6对词元进行索引。子单词标记、即BERT单词片段是使用HuggingFace Transformers库生成的。停止词列表不适用于BERT单词片段。

[0057] 基本设置。在配备有六核(12线程)i7-6800K 3.4 Ghz CPU、125 GB内存和四个

GeForce GTX 1080 TI GPU的Linux服务器上进行的实验。它使用了用Java实现的文本检索框架FlexNeuART。它使用具有BM25得分器的Lucene 7.6生成了初始候选列表，其可以使用传统的或神经重排序器来进一步重新排序。包括非参数模型1的传统的重排序器也是用Java实现的。它们以多线程模式(12个线程)运行并充分利用了CPU。神经排序器是使用PyTorch 1.4和Apache Thrift实现的。神经排序器作为独立的单线程服务器操作。

[0058] 排序速度是以每一个文档/段落的总体CPU/GPU吞吐量而不是时延来度量的。排序的准确性是使用TREC主办方提供的标准实用工具trec_eval来度量的。使用具有阈值0.05的双侧t检验来计算统计显著性。

[0059] 所有排序模型都应用于由调整的BM25得分器生成的候选列表。基于BERT的模型对具有最高BM25得分的100个条目进行重新排序:使用更大的候选池损害效率和准确性二者。包括神经上下文无关模型1的所有其他模型都对1000个条目重新排序:进一步增加候选数量不改进准确性。

[0060] 训练模型。使用逐对的裕度损失来训练神经模型。训练对是通过将已知的相关文档与从Lucene返回的前500个候选的集合中选择的20个负示例相组合而获得的。在每个时期，每个查询随机采样一个正和一个负示例。基于BERT的模型首先使用掩蔽语言建模和下一语句预测目标进行目标语料库预处理。然后，在一个时期内训练他们排序任务。使用经由梯度累积模拟的批量大小16。上下文无关模型1使用批量大小32从头开始训练32个时期。非参数模型1利用MGIZA训练五个时期。进一步增加时期的数量不显著改进结果。

[0061] 使用小的权重衰减(10^{-7})和预热时间表，其中，在10-20%的步骤内，学习速率从零线性增长，直到它达到基本学习速率。优化器包括AdamW。对于基于BERT的模型，使用不同的基本速率，全连接预测头($2 \cdot 10^{-4}$)和主Transformer层($2 \cdot 10^{-5}$)。对于上下文无关模型1，基本速率为$3 \cdot 10^{-3}$，其在每个时期后衰减0.9。这对于所有参数都相同。通过预计算所有逐对翻译概率并丢弃小于10^{-4}的概率，训练的神经模型1被导出为非参数格式。导出花费三分钟，并且导出的模型使用与非参数模型1相同的Java代码执行。

[0062] 针对五个种子(从零到四)训练和评估每个神经模型和稀疏化模型1。为此，计算每个查询的值以及特定于查询的种子值和平均值。所有的超参数都在开发集上进行调整。

[0063] 因为上下文无关的模型1排序器本身不强大，所以在融合模式下评估它们。首先，在训练/建模上训练模型1。然后将模型的得分与BM25得分进行线性组合。使用来自RankLib的坐标上升算法在训练/融合子集上计算最优权重。为了改进该线性融合的有效性，使用模型1，日志得分通过查询标记的数量来归一化。进而，BM25得分通过查询术语IDF值的总和进行归一化。对于不同的标记化方法，也使用BM25得分的融合，这也是经由RankLib在训练/融合上训练的。

[0064] 模型概述。比较几个模型(见表2)。首先，使用为词元化文本计算的BM25得分，此后称为BM25(词元)。其次，评估上下文无关模型1的几个变体。非参数模型1针对原始标记和BERT单词片段二者进行训练:相应的模型被标示为模型1(tok)和模型1(b/tok)。神经上下文无关模型1——标示为NN-模型1——仅与BERT单词片段一起使用。该模型被稀疏化并导出为非参数格式，其在CPU上高效运行。将其标示为NN-模型1-exp。注意，上下文无关的模型1 排序器本身不强大，因此，我们通过将它们的得分与BM25(词元)相组合，以融合模式对它们进行评估。

[0065] 至关重要的是,所有上下文无关的模型都经由自翻译概率或经由具有标记集合概率的显式平滑来并入精确的术语匹配信号(见等式2)。因此,这些模型不仅应该与BM25进行比较,而且还应该与并入原始标记或BERT单词片段的BM25得分的融合模型进行比较。将这些模型分别标示为BM25(词元)+ BM25(tok)和BM25(词元)+ BM25(b/tok)。

[0066] 上下文化模型1将神经模型1层应用于由BERT产生的上下文化嵌入。将该模型标示为BERT-模型1。由于现有预训练的Transformer模型的限制,因此长文档需要被拆分成块,每个块都被分别处理、即上下文化。这是在BERT-模型1(完整)、BERT-vanilla(完整)和BERT-CEDR模型中完成的。另一种方法是对截断的文档文本做出预测。这是在BERT-模型1(短)和BERT-vanilla(短)中完成的。在段落检索任务中,所有段落都是短的,并且不需要截断或分块。对于基于BERT的模型,使用基本模型、即12层Transformer模型,因为它比24层BERT-large实用,并且在MS MARCO数据上的性能与BERT-large相当。

[0067] 使用双侧t检验测试了几个假设:

	文档				段落			
	测试	TREC 2019	TREC 2020	排序速度	测试	TREC 2019	TREC 2020	排序速度
	MRR	NDCG@10	NDCG@10	ms/1K	MRR	NDCG@10	NDCG@10	ms/1K
BM25(词元)	0.270	0.544	0.524	3	0.256	0.522	0.516	1
BM25(词元)+BM25(tok)	0.274	0.544	0.523	57	0.265	0.517	0.521	1
BM25(词元)+模型1(tok)	0.283*	0.548	0.535	18	0.274*	0.522	0.567*	3
BM25(词元)+BM25(b/tok)	0.283	0.528	0.537	44	0.270	0.518	0.525	2
BM25(词元)+模型1(b/tok)	0.284	0.557	0.525	53	0.271	0.517	0.509	19
BM25(词元)+NN-模型1-exp	0.307*	0.568	0.545	20	0.298*	0.541*	0.581*	6
BM25(词元)+NN-模型1	0.311*	0.566	0.541	3030	0.300*	0.549*	0.587*	313
BERT-vanilla(短)	0.387	0.655	0.623	37200	0.426	0.686	0.684	10100
BERT-vanilla(完整)	0.376#	0.667	0.631	82200				
BERT-模型1(短)	0.384	0.657	0.631	36600	0.426	0.685	0.682	11700
BERT-模型1(完整)	0.391#	0.666	0.637*	83900				
BERT-CEDR-KRNM	0.387	0.665	0.649*	84600	0.421*	0.682	0.675	10600
BERT-CEDR-DRMM	0.377*	0.667	0.636	114000	0.425	0.688	0.685	20100
BERT-CEDR-PACRR	0.392	0.670	0.652*	81500	0.425	0.690	0.684	10500

表2

[0068] 表2说明评估结果:b/tok标示BERT单词片段,词元标示文本词元,并且tok标示原始标记。NN-模型1和NN-模型1-exp是上下文无关的神经模型1及其在CPU上运行的导出变体。两者都仅使用b/tok。统计显著性用*和#标示。

[0069] 上下文无关的神经模型1与相应的融合基线相同,例如,将BM25(词元)+模型1(tok)与BM25(词元)+ BM25(tok)进行比较;

BERT-模型1(完整)与基线BERT-vanilla(短)相同;

BERT-CEDR模型与基线BERT-vanilla(短)相同;

BERT-vanilla(完整)与基线BERT-vanilla(短)相同;和

BERT-模型1(完整)与基线BERT-模型1(短)相同。

[0070] 这些测试的主要目的是评估与在截断文档上运行的模型相比，特殊聚合层(包括神经模型1)是否可以更准确。在表2中，统计显著性由特殊符号指示：最后两个假设使用#；所有其他假设使用*。

[0071] 结果讨论。结果总结在表2中。首先注意，与测试集相比，TREC 2019/2020测试集上的结果中不太存在一致性。在这种情况下，一些统计上显著的差异(在测试中)在TREC 2019/2020上“消失”。TREC 2019/2020查询集非常小，并且更可能(与测试相比)获得虚假结果。此外，融合模型BM25(词元)+模型1(b/tok)要么比基线模型BM25(词元)+ BM25(b/tok)差，要么差异不显著。BM25(词元)+ 模型1(tok)大多比相应的基线好，但增益非常小。相比之下，神经模型1与BERT单词片段的BM25得分的融合在所有查询集上都更准确。在测试集上，它比BM25(词元)好15-17%。这些差异在测试集上以及TREC 2019/2020测试集上二者针对段落检索任务都是显著的。神经模型1的稀疏化不导致准确性方面的明显下降。在这种情况下，稀疏化模型——在CPU上执行——比基于BERT的排序器(在GPU上运行)快10^3倍以上。对于大的候选集，模型1得分的计算可以进一步改进。因此，BM25(词元)+NN-模型1-exp在第一检索阶段可能是有用的。

[0072] 最后，将基于BERT的神经模型1的准确性与BERT-CEDR和BERT-vanilla模型进行比较。通过在测试集(用于文档检索任务)上将BERT-vanilla(短)与BERT-模型1(短)进行比较，以及将BERT-vanilla(长)与BERT-模型1(长)进行比较，可以示出神经模型1层实际上没有效率或准确性损失。在这种情况下，在测试集上，BERT-模型1(完整)——其在完整文档上操作——比BERT-模型1(短)和BERT-vanilla(短)二者更准确大约1%，然而，后者的差异在统计上不显著。在这种情况下，对于BERT-CEDR-PACRR也是如此，它也比BERT-vanilla好1%。这没有使结果无效，该结果示出，将BERT-CEDR-PACRR与vanilla BERT排序器的“完整”变体比较，BERT-CEDR-PACRR比BERT-vanilla更好，vanilla BERT排序器对平均的[CLS]嵌入做出预测。然而，在我们的实验中，该模型明显比更简单的BERT-vanilla(短)模型——其在截断的文本上操作——差(差4.2%)，并且差异在统计上是显著的。据信，获得更多关于聚合层有效性的结论性证据需要不同的数据集，其中文档和段落的相关性更难从截断的文档进行预测。

[0073] 在利用词典翻译模型——特别是将神经模型1和标记嵌入网络(上下文化和上下文无关二者)组合——的实验中。发现后者有益于效率和可解释性。此外，神经模型1是可以被稀疏化并导出为“非神经”格式的唯一神经模型，它可以在CPU上高效运行(比GPU上的BERT快$10^3\times$)，而无需昂贵的索引时间预计算。该方法的有效性可以进一步改进，例如，通过设计更好的条件翻译概率的参数化。可解释的神经模型1层对于长文档的有效排序也可能是有用的，但是结论性实验需要具有与MS MARCO不同的特性的数据集。可以示出，即使当查询和文档具有相差很大的长度时，非参数模型1也可以经由EM进行训练，但是通过训练上下文无关的神经模型1并将其导出为非参数格式，获得更好的结果。

[0074] 本公开提出了一种利用特殊的神经网络层、即词典翻译模型(例如，IBM模型1)来对搜索结果进行排序的机制。首先使用上下文无关的或上下文的神经网络(例如BERT)对查询和文档进行编码。因此，查询和文档标记由向量序列表示。然后，特殊的词典神经翻译层摄取这些向量，并产生像查询术语概率的相似性得分的矩阵。然后，使用简单的和积公式，

以可解释的方式组合这些得分。当使用上下文化网络时，高效的执行通常需要特殊的加速器，诸如图形处理单元(GPU)或张量处理单元(TPU)。然而，对于上下文无关的编码网络，该机制依赖于查询文档标记概率矩阵的预计算和稀疏化。这准许在较低性能的系统、处理器或控制器上高效执行。

[0075] 典型的文本检索系统依赖于简单的术语/标记匹配技术来生成候选的初始列表，该候选的初始列表可以使用学习的模型进一步重新排序。因此，查询和文档术语之间的不匹配对检索性能产生不利影响，这被称为词汇差距问题。为了减小词汇差距并且改进检索效率，引入了名为IBM模型1的词典翻译模型。当应用于使用问题作为查询在英语问答(QA)档案中寻找答案以及跨语言检索时，IBM模型1具有很强的性能。模型1是在基本上相似长度的问题文档对上训练的，这简化了在查询术语和相关文档中的术语之间寻找有用关联的任务。已经主要使用产生非参数模型的传统期望最大化(EM)算法的变体在翻译任务中训练这些模型。本公开的关键步骤包括用神经网络对条件翻译概率进行参数化并在排序(而不是翻译)任务中对模型进行端到端学习。

[0076] 以下以简化的形式描述概念，所述概念不是穷尽的，并且不旨在限制本公开的范围，而是呈现概念的基本描述，所述概念贯穿本公开被更详细地讨论。

[0077] 本公开提供了一种多阶段检索系统，其可以用于多种域，包括但不限于公共和专用网络(例如，万维网)上的移动和桌面搜索。在这样的系统中，其中文档通过一系列“漏斗”流动，所述一系列“漏斗”使用越来越复杂和准确的排序组件丢弃没有希望的候选文档。查询和文档二者都是可以用标记序列表示的文本碎片。标记可以是完整的单词、词位(词元)、单词片段或多单词短语。将文本碎片拆分为标记称为标记化或进行标记化。

[0078] 在检索系统中，神经排序模型可以应用于查询和文档，以产生相关性得分的估计。这些模型更复杂和昂贵，并且因此，它们用在检索流水线的后期阶段。尽管用神经排序模型计算的这些估计可能非常准确，但它们难以解释。本公开呈现了可解释的神经网络层——神经词典翻译模型(例如，IBM模型1)，其可以应用于最先进的神经网络，诸如BERT。连同效率和有效性的益处。

[0079] 特别地，上下文无关的词典翻译模型可以被完全预计算(诸如对于所有查询和文档标记对)、稀疏化，并在较低性能的系统、处理器或控制器(例如，诸如GPU和TPU的不需要昂贵加速器的CPU)上高效执行。尽管这样的模型具有用于最终排序的较差准确性，但它仍然可以改进较早阶段检索。

[0080] 图1是搜索检索系统100的框图。这是简化的描述并且不包括所有的实现细节。搜索检索系统100旨在简单地访问文档集合102，文档集合102可以使用以多种格式(例如，文本、富文本格式(rtf)、微软Word、PDF等)的多种介质(例如，RAM、磁、光和固态存储设备)来存储。它也可以以不同的形式组织：存储为文件系统中的文件、超链接文档的集合或关系数据库表中的内部字段。集合可以全部或部分处理。特别地，文档发现引擎104(例如，网络爬虫或数据库查询应用)首先用于发现值得包含在索引中的文档。文本处理引擎106标识文档格式，为它们分配(通常)唯一的标识符，并提取文本数据。文档然后被存储在(1)准许使用其标识符快速检索文档文本的直接(或正向)索引110和(2)准许快速检索包含一个或多个标记的文档的反向(或逆向)索引108中。

[0081] 当用户使用用户设备(例如，台式计算机、膝上型计算机、智能电话等)112提交查

询(即,寻找索引文档的搜索请求)时,其采用多阶段检索流水线。尽管图像示出了键盘和显示器,但是对用户设备112的输入可以包括触摸屏、键盘、麦克风、相机、扫描仪、图像文件或其他文件或输入设备。此外,到用户设备112的输出可以包括显示器、投影仪、扬声器或文件或其他输出设备200。第一阶段使用轻量级查询引擎114,该轻量级查询引擎114获取相对大的候选文档集116。为此,它依赖于简单的标记/术语匹配技术,并采用先前创建的反向索引108。后续阶段采用零个或多个非神经轻量级排序组件118来细化原始候选集。这些重排序器可以非常高效,并且它们在CPU上运行,即它们不需要使用诸如GPU或TPU的神经网络昂贵的加速器。并且,由于神经网络排序器120通常在计算上昂贵,因此它们仅在检索流水线的最新阶段使用。

[0082] 图2是排序神经网络200的流程图。考虑给查询Q 202和文档D 204对分配相关性得分的问题。查询202和文档204二者都是可以由标记序列表示的文本碎片。标记可以是完整的单词、词位(词元)、单词片段或多单词短语。查询标记用q标示(可能带有索引),并且文档标记用d标示(可能带有索引)。文档D的长度以标记的数量来度量,并且用$|D|$标示。

[0083] 因为神经网络不能直接对文本数据进行操作,所以标记序列$t_1t_2 \ldots t_n$首先通过嵌入神经网络被转换成d维嵌入向量的序列$w_1w_2 \ldots w_n$,该嵌入神经网络由一个或多个组件组成。

[0084] 典型地,这样的网络首先产生用于查询标记的上下文无关嵌入206和用于文档标记的上下文无关嵌入208。用于相同查询(或文档标记)的上下文无关嵌入总是相同的,不管它们的位置和周围的标记如何。然而,它们在查询和文档之间可能是不同的。

[0085] 在产生上下文无关的嵌入206和208之后,诸如Transformer模型(例如,BERT)的可选神经网络210产生上下文化查询标记嵌入212和上下文化文档标记嵌入216。这些嵌入依赖于上下文,并且永远不相同,除非我们在相同的上下文中具有精确的标记。

[0086] 除了上下文化的标记嵌入之外,网络210可以产生一个或多个(池化)嵌入214,其表示文档的较大部分。例如,对于512个标记的每个序列,BERT为所谓的类标记[CLS]产生嵌入向量,它表示整个序列。

[0087] 当不使用可选的上下文化网络210时,212与206相同并且216与208相同。

[0088] 最后,嵌入212、214和216(或者在其中可选的神经上下文化的层210不可用的瞬间,嵌入206和208)被馈送到交互神经网络,该交互神经网络产生最终的查询文档排序得分220。交互神经网络218可以是多层交互神经网络218。

[0089] 训练:训练神经网络的一个方法涉及编译正文档和负文档的对(或列表),它们分别与给定的查询相关和不相关,并且强制相关文档的排序得分高于不相关文档的得分。这是通过将得分(或它们的差异)馈送到名为损失函数的特殊比较函数中,并使用各种优化方法使损失最小化来完成的。当非相关文档比相关文档排序高时,规定损失函数是高的。

[0090] 本公开中呈现的关键方面是,与使用并行语料库在翻译任务中训练神经词典翻译模型的系统不同,该方法允许系统在单语言排序任务中训练模型。平行语料库是不同语言中的成对短文档(大部分是单语句文档)集。

[0091] 特殊神经网络层:图3是神经排序器架构300的流程图,其接收查询302和文档304,生成上下文无关的嵌入306和308,所述嵌入可选地通过上下文化神经网络310传递。嵌入

312和314然后由聚合组件316和318处理，以产生最终的排序得分320。

[0092] 本公开中呈现的另一个关键方面是用部分可解释的聚合组件316和318替换图2的交互层218。

[0093] 首先，神经网络316为所有查询和文档标记对产生像概率的翻译得分$T(q|d)$。如下面解释的，经由神经网络计算$T(q|d)$。然后，使用简单的和积公式318来组合这些值，该和积公式318将所有单个标记的得分相乘：

$$P(Q|D) = \prod_{q\in Q} \sum_{d\in D} T(q|d)P(d|D) \quad (5)。$$

[0094] 在公式(5)中，$P(d|D)$标示文档嵌入是由文档“生成”的概率。对于上下文无关的嵌入，它是最大似然估计，即术语出现的归一化频率。在上下文化嵌入的情况下，它被设置为$1/|D|$。

[0095] 转到使用神经网络计算词典翻译概率$T(q|d)$。本公开的另一个关键步骤包括用神经网络对条件翻译概率进行参数化。它如下进行：

首先，系统使查询和文档标记化。

[0096] 接下来，系统采用图3的嵌入网络来生成查询标记的嵌入312(标示为w_q)和文档标记的嵌入314(标示为w_d)。该方法对上下文无关嵌入和上下文化嵌入的神经网络二者有效。

[0097] 最后，使用查询和文档标记的嵌入，系统使用前馈神经网络来计算翻译概率$T(q|d)$。存在多个方式来进行此。在一个实现中，系统如下进行：

1. $T(q|d) = \sigma(F_3(\text{relu}(F_2(\text{relu}(F_1([w_q, w_d, w_q \circ w_d])))))$.
2. $w_q = P_q(\tanh(\text{层范数}(\text{embed}_q(q))))$.
3. $w_d = P_d(\tanh(\text{层范数}(\text{embed}_q(d))))$.
4. F_l, P_d, P_q 是全连接层。
5. $\sigma(x)$ 是sigmoid激活函数。
6. $\tanh(x)$ 是tanh激活函数。
7. 层范数是层归一化。
8. $[x, y]$ 标示向量连结。
9. $x \circ y$ 标示向量之间的哈达玛积。

[0098] 这是基于在步骤406和408中计算的标记嵌入，经由神经条件翻译概率网络计算给定文档标记的查询标记的条件翻译概率的方法。

[0099] 现在考虑查询文档标记概率矩阵的预计算和稀疏化(神经导出)。当图3的排序系统将神经词典翻译层与上下文化嵌入相组合时，如本公开中所述，总体系统具有最先进的性能。然而，它在计算上可能也很昂贵。使用上下文无关的嵌入导致性能方面的显著下降。然而，结果得到的模型可以(1)稀疏化和预计算，以及(2)与简单的TF*IDF模型(诸如BM25)组合，并在CPU上高效执行以改进图1的早前阶段排序组件118。

[0100] 图4是模型预计算和稀疏化400的流程图，其中使用了上下文无关的标记嵌入网络。首先，处理器或控制器为查询402和文档标记404的所有频繁组合生成嵌入406和408。然后，控制器或处理器使用上下文无关的模型$T(q|d)$(例如，使用神经网络计算词典翻译概

率)来计算所有可能的逐对翻译概率/得分。

[0101] 在步骤410中,处理器或控制器基于在步骤406和408中计算的标记嵌入,经由神经条件翻译概率的网络计算给定文档标记的查询标记的条件翻译概率。

[0102] 步骤410的计算结果是可以在步骤412中存储的密集查询文档标记翻译概率矩阵。例如,在步骤412中,来自GPU的步骤410的输出被存储在CPU存储器中。

[0103] 稀疏化步骤416丢弃低于经验选择的阈值的所有得分。如本公开中所述,这准许在CPU上高效计算等式(1)。阈值的示例包括一个标记与另一个标记匹配的为千分之一(0.1%或0.001)、0.2%、0.5%、0.75%、1%或2%的概率。结果是稀疏查询文档标记翻译概率矩阵418。

[0104] 示例机器架构和机器可读介质。图5是适合用于实现本文公开的系统或执行本文公开的方法的电子计算系统的框图。图5的机器被示出为独立的设备,其适合用于以上概念的实现。对于上述服务器方面,可以使用在数据中心、云架构的一部分等中操作的多个这样的机器。在服务器方面,并非所有图示的功能和设备都被利用。例如,当用户用来与服务器和/或云架构交互的系统、设备等可以具有屏幕、触摸屏输入等时,服务器通常没有屏幕、触摸屏、相机等,并且通常通过具有适当输入和输出方面的连接系统与用户交互。因此,下面的架构应该被视为包含多个类型的设备和机器,并且取决于其形状因素和用途,各种方面可能存在于或可能不存在于任何特定的设备或机器中(例如,服务器鲜少具有相机,而可穿戴设备鲜少包括磁盘)。然而,图5的示例解释适合于允许本领域技术人员确定如何利用硬件和软件的适当组合来实现先前描述的实施例,并对所使用的特定设备、机器等的图示实施例进行适当修改。

[0105] 虽然仅图示了单个机器,但是术语"机器"也应当被理解为包括单独或联合执行指令集(或多个集)以执行本文讨论的方法中的任何一个或多个的任何机器集合。

[0106] 机器500的示例包括至少一个处理器502(例如,控制器、微控制器、中央处理单元(CPU)、图形处理单元(GPU)、张量处理单元(TPU)、高级处理单元(APU)或其组合)、一个或多个存储器、诸如主存储器504、静态存储器506或其他类型的存储器,它们经由链路508彼此通信。链路508可以是总线或其他类型的连接信道。机器500可以包括进一步可选的方面,诸如包括任何类型的显示器的图形显示单元510。机器500还可以包括其他可选方面,诸如字母数字输入设备512(例如,键盘、触摸屏等)、用户界面(UI)导航设备514(例如,鼠标、轨迹球、触摸设备等)、存储单元516(例如,(一个或多个)磁盘驱动器或其他存储设备)、信号生成设备518(例如,扬声器)、(一个或多个)传感器521(例如,全球定位传感器、(一个或多个)加速度计、(一个或多个)麦克风、(一个或多个)相机等)、输出控制器528(例如,有线或无线连接,以连接一个或多个其他设备和/或与一个或多个其他设备通信,所述一个或多个其他设备诸如是通用串行总线(USB)、近场通信(NFC)、红外(IR)、串行/并行总线等)以及用于连接到一个或多个网络526和/或通过一个或多个网络526通信的网络接口设备520(例如,有线和/或无线)。

[0107] 各种存储器(即,504、506和/或(一个或多个)处理器502的存储器)和/或存储单元516可以存储指令和数据结构(例如,软件)524的一个或多个集,所述指令和数据结构(例如,软件)524的一个或多个集体现本文描述的方法或功能中的任何一个或多个或者被本文描述的方法或功能中的任何一个或多个利用。这些指令当由(一个或多个)处理器502执行时,引起各种操作来实现所公开的实施例。

[0108] 如本文使用的，术语“机器存储介质”、“设备存储介质”、“计算机存储介质”意指相同的东西，并且可以在本公开中可互换使用。术语指代存储可执行指令和/或数据的单个或多个存储设备和/或介质(例如，集中式或分布式数据库，和/或相关联的高速缓存和服务器)。因此，术语应被理解为包括诸如固态存储器的存储设备，以及包括处理器内部或外部的存储器的光学和磁性介质。机器存储介质、计算机存储介质和/或设备存储介质的具体示例包括非易失性存储器，包括例如半导体存储器设备，例如可擦除可编程只读存储器(EPROM)、电可擦除可编程只读存储器(EEPROM)、FPGA和闪存设备；磁盘，诸如内部硬盘和可移动磁盘；磁光盘；以及CD-ROM和DVD-ROM盘。术语机器存储介质、计算机存储介质和设备存储介质具体并且明确地排除载波、调制数据信号和其他这样的暂时介质，其中至少一些包含在下面讨论的术语“信号介质”中。

[0109] 术语“机器可读介质”、“计算机可读介质”和“设备可读介质”意指相同的东西，并且可以在本公开中可互换使用。术语被定义为包括机器存储介质和信号介质二者。因此，术语包括存储设备/介质和载波/调制数据信号二者。

[0110] 体现本文描述的算法和/或方法的程序代码能够以多种不同的形式作为程序产品单独或共同分发。程序代码可以使用其上具有计算机可读程序指令的计算机可读存储介质来分发，用于使得处理器实行一个或多个实施例的方面。固有地非暂时性的计算机可读存储介质可以包括以用于存储信息的任何方法或技术实现的易失性和非易失性以及可移动和不可移动的有形介质，诸如计算机可读指令、数据结构、程序模块或其他数据。计算机可读存储介质可以进一步包括RAM、ROM、可擦除可编程只读存储器(EPROM)、电可擦除可编程只读存储器(EEPROM)、闪存或其他固态存储器技术、便携式光盘只读存储器(CD-ROM)或其他光存储装置、盒式磁带、磁带、磁盘存储装置或其他磁存储设备，或者可以用于存储所期望信息并且可以由计算机读取的任何其他介质。计算机可读程序指令可以从计算机可读存储介质下载到计算机、另一类型的可编程数据处理装置或另一设备，或者经由网络下载到外部计算机或外部存储设备。

[0111] 存储在计算机可读介质中的计算机可读程序指令可以用于指导计算机、其他类型的可编程数据处理装置或其他设备以特定方式运行，使得存储在计算机可读介质中的指令产生包括实现流程图或图表中指定的功能、动作和/或操作的指令的制品。在某些替代实施例中，流程图和图表中指定的功能、动作和/或操作可以与一个或多个实施例一致地被重新定次序、串行处理和/或并发处理。此外，流程图和/或图表中的任何一个可以包括比与一个或多个实施例一致的那些图示的更多或更少的节点或块。

[0112] 虽然已经通过各种实施例的描述说明了本发明的全部，并且已经相当详细地描述了这些实施例，但是申请人不意图将所附权利要求书的范围限制或以任何方式限定到这样的细节。本领域的技术人员将容易想到附加的优点和修改。因此，本发明在其更宽泛的方面不限于特定的细节、代表性的装置和方法以及所示和所述的说明性示例。因此，在不脱离总的发明构思的精神或范围的情况下，可以做出从这样的细节的偏离。

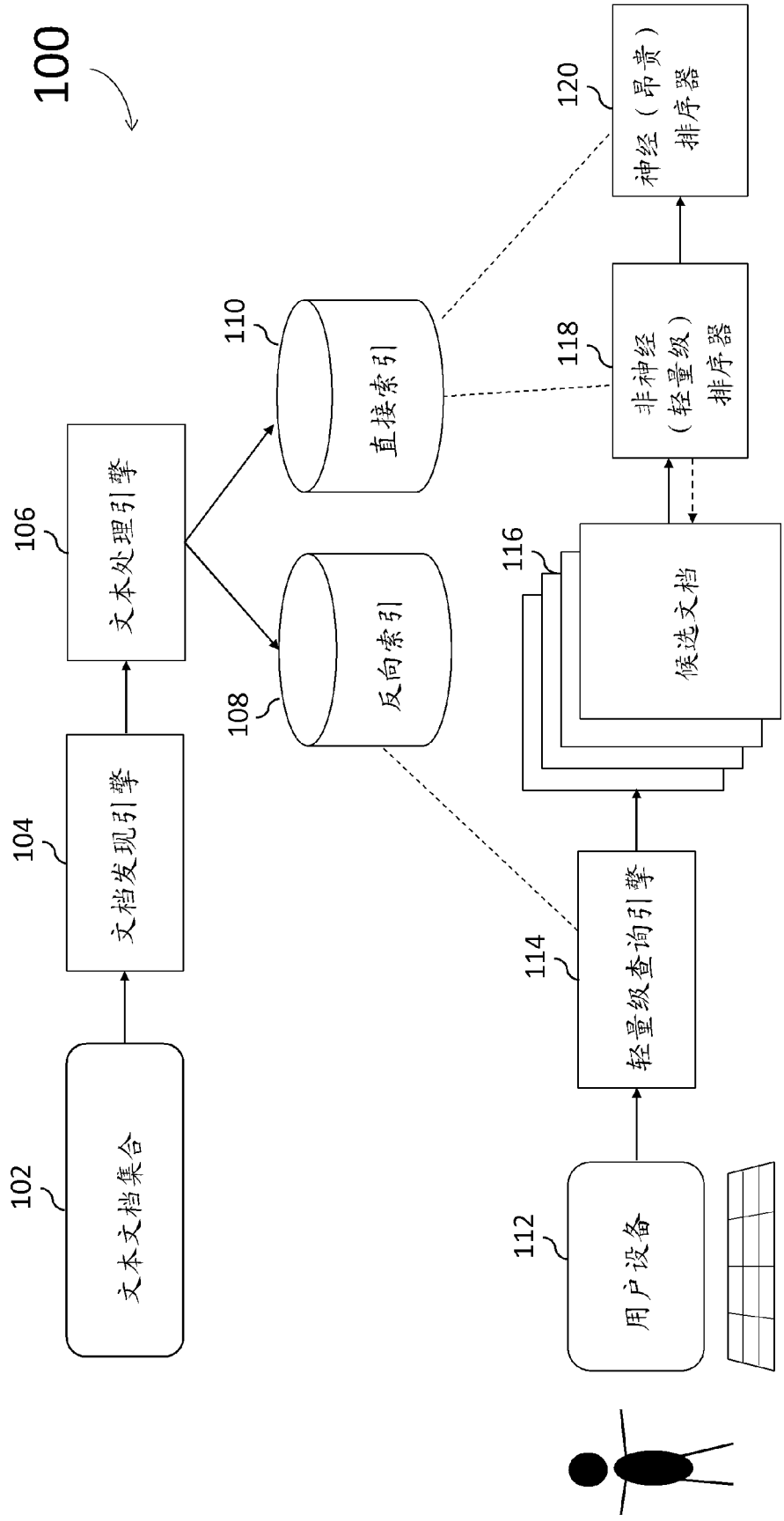

100
102
文本文档集合
104
文档发现引擎
106
文本处理引擎
108
反向索引
110
直接索引
112
用户设备
114
轻量级查询引擎
116
候选文档
118
非神经（轻量级）排序器
120
神经（昂贵）排序器

图 1

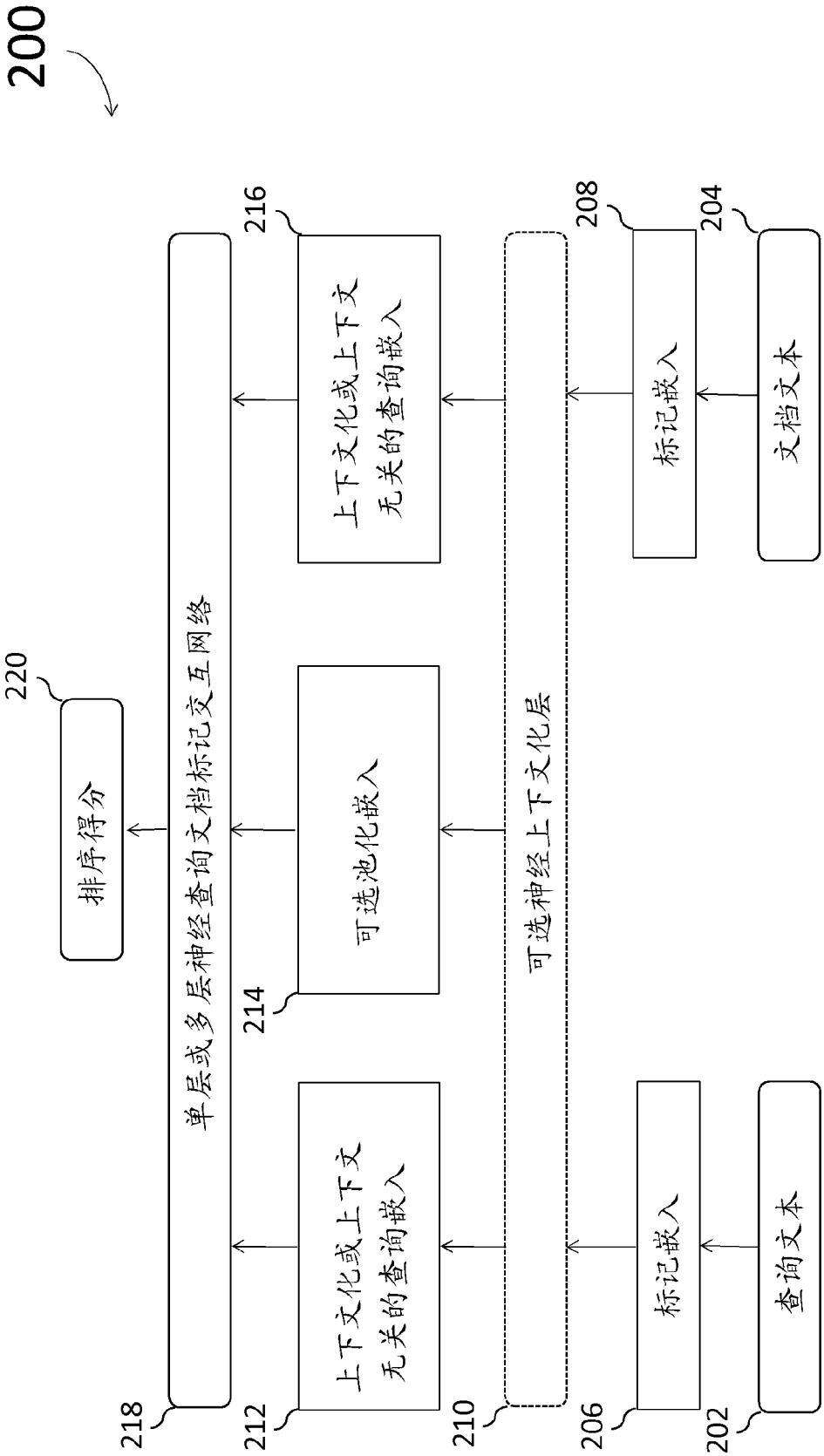

200
220
排序得分
218
单层或多层神经查询文档标记交互网络
212
上下文化或上下文无关的查询嵌入
214
可选池化嵌入
216
上下文化或上下文无关的查询嵌入
210
可选神经上下文化层
206
标记嵌入
208
标记嵌入
202
查询文本
204
文档文本

图 2

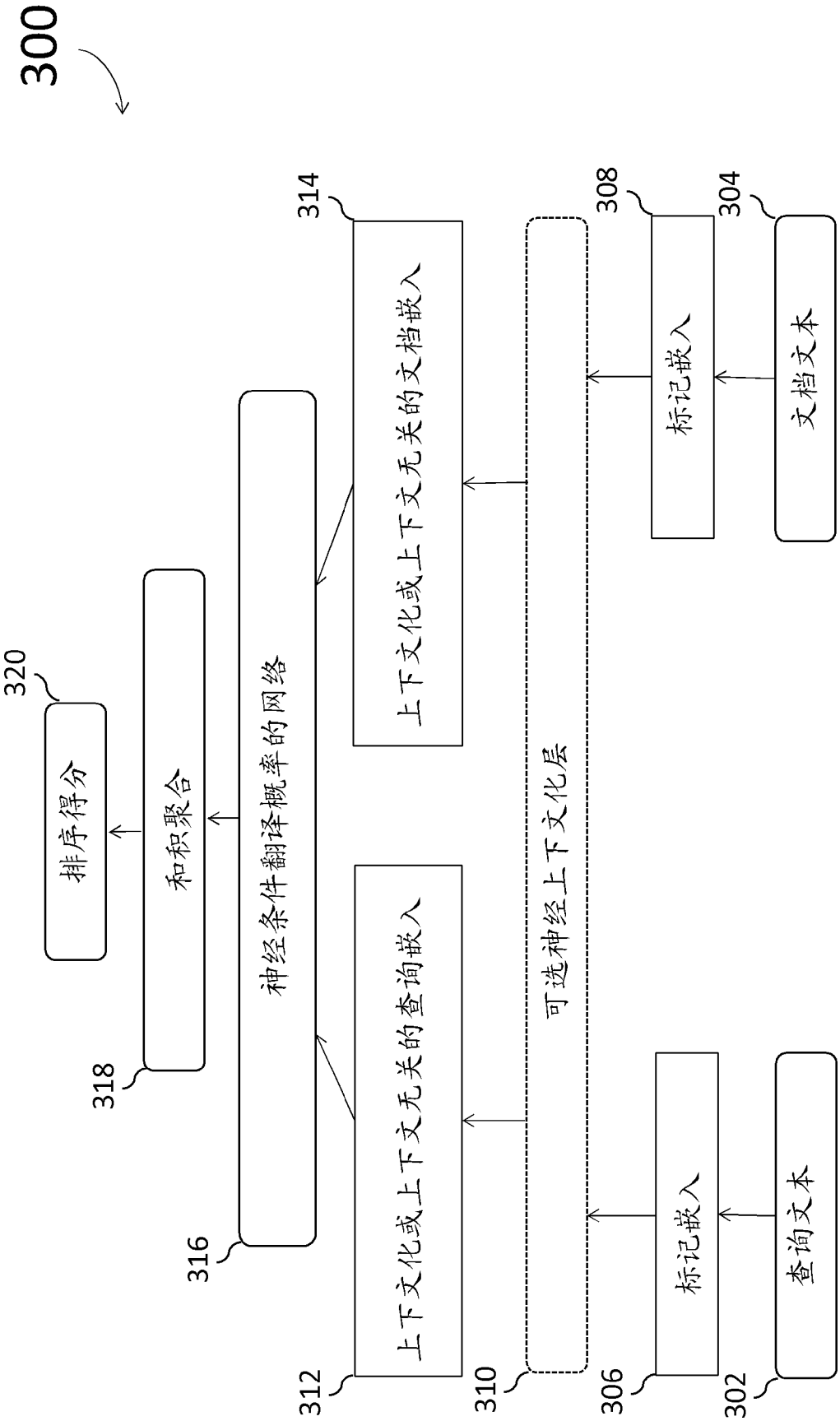

300
320
排序得分
318
和积聚合
316
神经条件翻译概率的网络
312
上下文化或上下文无关的查询嵌入
310
可选神经上下文化层
306
标记嵌入
302
查询文本
314
上下文化或上下文无关的文档嵌入
308
标记嵌入
304
文档文本

图 3

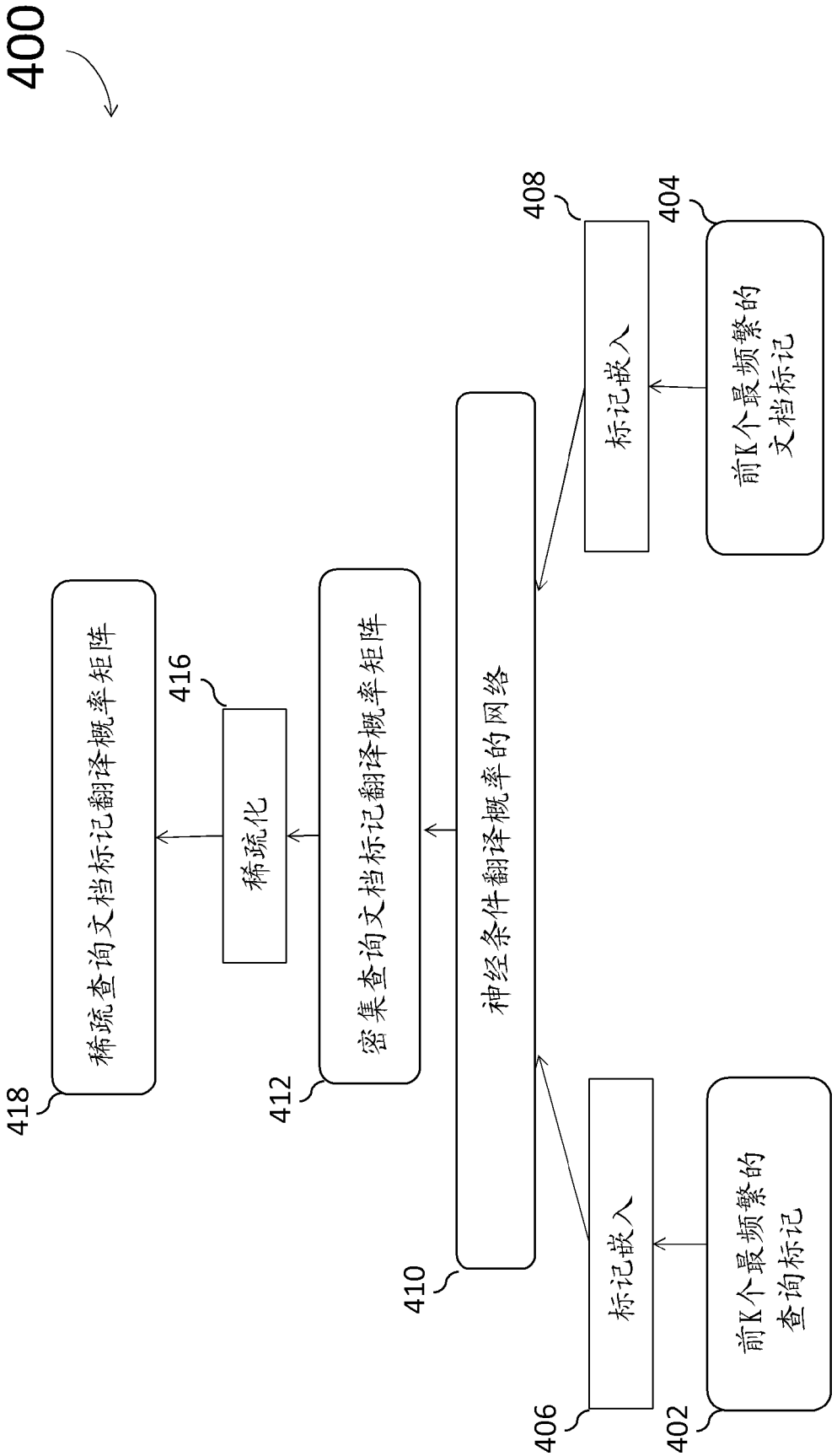

400
稀疏查询文档标记翻译概率矩阵
418
稀疏化
416
密集查询文档标记翻译概率矩阵
412
神经条件翻译概率的网络
410
标记嵌入
408
前K个最频繁的
文档标记
404
标记嵌入
406
前K个最频繁的
查询标记
402

图 4

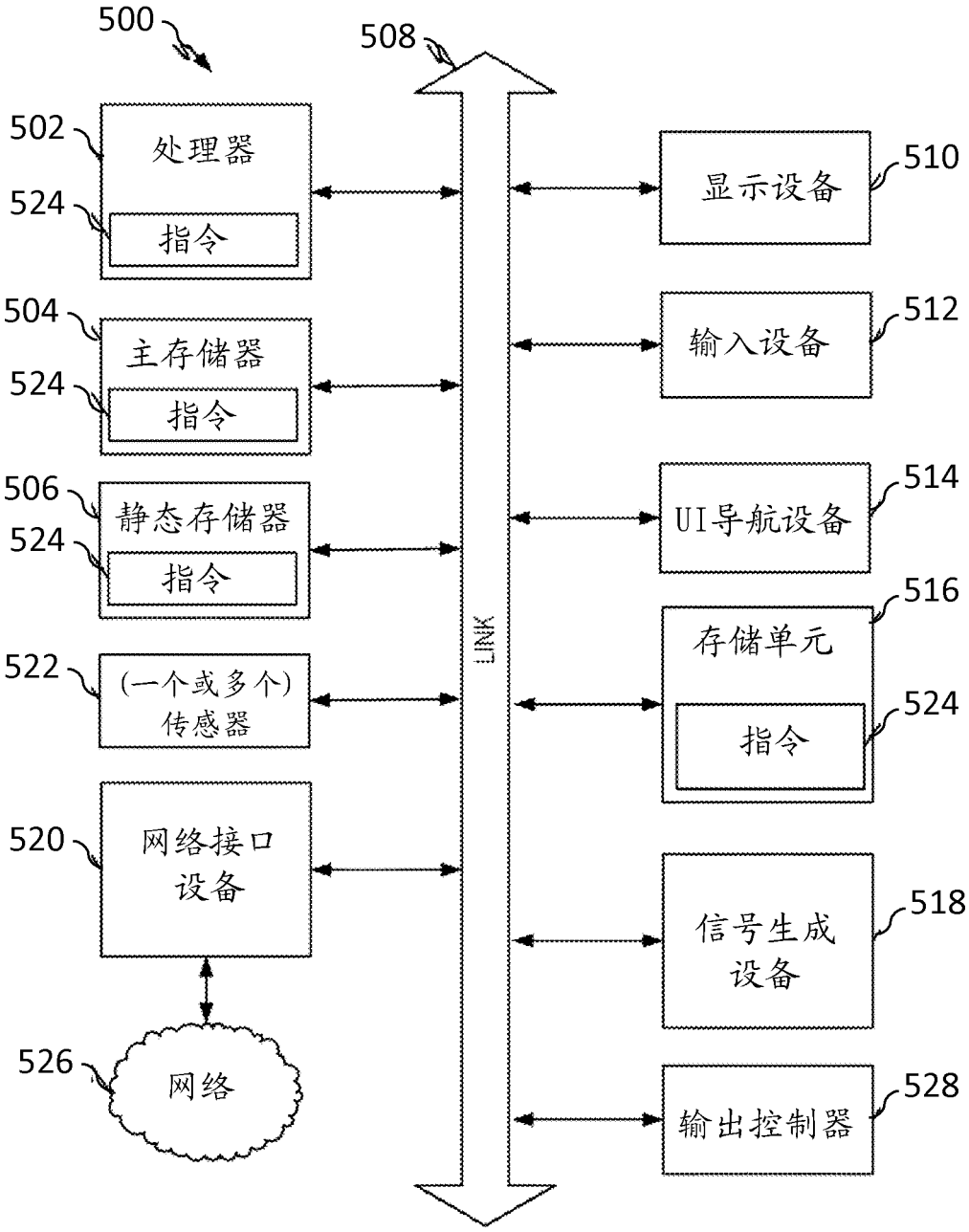

500
508
502
处理器
524
指令
504
主存储器
524
指令
506
静态存储器
524
指令
522
(一个或多个)
传感器
520
网络接口
设备
526
网络
LINK
显示设备
510
输入设备
512
UI导航设备
514
516
存储单元
524
指令
信号生成
设备
518
输出控制器
528

图 5