(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 113934835 B
(45)授权公告日 2022.03.25

(21)申请号 202111538357.1
(22)申请日 2021.12.16
(65)同一申请的已公布的文献号
申请公布号 CN 113934835 A
(43)申请公布日 2022.01.14
(73)专利权人 之江实验室
地址 310023 浙江省杭州市余杭区文一西路1818号人工智能小镇10号楼
(72)发明人 李太豪 张晓宁 阮玉平 郑书凯
(74)专利代理机构 杭州浙科专利事务所(普通合伙) 33213
代理人 孙孟辉 杨小凡
(51)Int.Cl.
*G06F 16/332*(2019.01)
*G06F 16/33*(2019.01)
*G06F 16/335*(2019.01)

(56)对比文件
CN 112256860 A,2021.01.22
CN 113505198 A,2021.10.15
US 2021/0326371 A1,2021.10.21
吴俣 等.检索式聊天机器人技术综述.《计算机科学》.2021,第48卷(第12期),第278-285页.
谢琪 等.一种基于多粒度循环神经网络与词注意力的多轮对话回答选择方法.《小型微型计算机系统》.2021,第42卷(第12期),第2553-2560页.
Xuemiao Zhang等.Adaptively Multi-Objective Adversarial Training for Dialogue Generation.《Proceedings of the Twenty-Ninth Inter national Joint Conference on Ar tificial Intelligence》.2020,第2872-2878页.

审查员 李欢

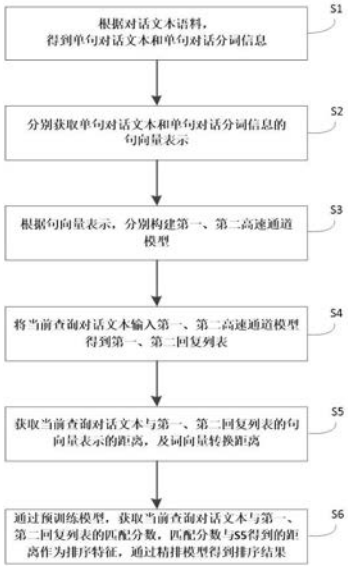
权利要求书3页 说明书11页 附图4页

(54)发明名称

# 结合关键词和语义理解表征的检索式回复对话方法及系统

(57)摘要

本发明公开了结合关键词和语义理解表征的检索式回复对话方法及系统,系统结合了两种层次粒度的向量表征,分别是词袋向量表征和语义理解表征,结合过程中不只考虑了对话中关键词的信息,还考虑了基于上下文的语义理解,极大地提升了检索式回复模型的性能。本发明中采取了中文预训练模型Bert网络模型获取句向量表征,不仅理解句意,并且排除了词向量加权引起的误差。该系统采取了Bert网络模型在自己的单轮对话上训练分类任务——对话是否匹配的任务,通过微调,学习到了Bert中线性层和激活函数的权重。该系统使用了精排模型LGMRanker,可以直接预测与query相关的回复相对顺序,返回一个排好序的列表回来。




CN 113934835 B

1.结合关键词和语义理解表征的检索式回复对话方法,其特征在于包括如下步骤:

S1,预处理对话文本语料,得到单句对话文本和单句对话分词信息;

S2,根据单句对话分词信息,通过训练好的词向量转换模型,获取单句对话向量表示;计算单句对话分词信息与所有单句对话文本的词频-逆向文件频率向量表示,根据单句对话向量表示与其对应的词频-逆向文件频率向量表示,得到该句对话的基于关键词表征学习的句向量表示;将单句对话文本,输入到预训练网络,得到基于语义理解网络学习的句向量表示;

S3,通过S2得到的基于关键词表征学习的句向量表示,构建分层的第一高速通道模型;通过S2得到的基于语义理解模型网络学习的句向量表示,构建分层的第二高速通道模型;

S4,针对当前的查询对话文本,基于S2获取当前查询对话文本的向量表示作为第一查询对话文本,将第一查询对话文本输入第一高速通道模型,检索出相似的对话文本,再将相似的对话文本对应的回复返回,得到第一回复列表;基于S2获取当前查询对话文本的向量表示作为第二查询对话文本,将第二查询对话文本输入第二高速通道模型,检索出相似的对话文本,再将相似的对话文本对应的回复返回,得到第二回复列表;

S5,将第一回复列表和第二回复列表,与当前查询对话文本建立一一对应的对话形式数据结构,根据该数据结构,通过S2分别计算出当前查询对话文本与第一回复列表和第二回复列表中每个回复的句向量表示,并计算当前查询对话文本的句向量表示分别与第一回复列表和第二回复列表中每个回复的句向量表示之间的距离;根据一一对应的对话形式数据结构,通过训练好的词向量转换模型,分别获取当前查询对话文本的向量表示分别与第一回复列表和第二回复列表中每个回复的向量表示,计算当前查询对话文本与每个回复之间词向量转换距离;

S6,通过训练好的预训练网络计算上下文是否匹配,将当前查询对话文本分别与第一回复列表和第二回复列表的每个回复,输入预训练网络中,得到匹配分数;将匹配分数与其对应的所述当前查询对话文本与第一回复列表和第二回复列表中每个回复的句向量表示之间的距离、词向量转换距离作为排序特征,输入精排模型中,获取精排后的回复结果顺序,选取回复结果顺序中最优回复作为当前查询对话文本的回复。

2.根据权利要求1所述的结合关键词和语义理解表征的检索式回复对话方法,其特征在于根据采集的对话文本语料,预处理得到单句对话文本和单句对话分词信息,将单句对话文本和单句对话分词信息输入词向量转换模型进行训练,得到训练好的词向量转换模型。

3.根据权利要求1或2所述的结合关键词和语义理解表征的检索式回复对话方法,其特征在于所述预处理,是对采集的对话文本语料进行拆分,得到单轮对话文本,单轮对话文本是以相邻的两句对话作为单轮的对话,对相邻的两句对话进行拆分,得到单句对话分词信息。

4.根据权利要求1所述的结合关键词和语义理解表征的检索式回复对话方法,其特征在于所述S2中的词频-逆向文件频率向量表示,是根据词频TF(x)与逆向文件频率IDF(x)的乘积确定,TF(x)根据词x在该句中出现的次数与该句所有的词数的比值确定,IDF(x)根据对话的所有数量与包含词x的所有对话数量的比值确定。

5.根据权利要求1所述的结合关键词和语义理解表征的检索式回复对话方法,其特征

在于所述S2中，根据单句对话向量表示与其对应的词频-逆向文件频率向量表示，得到该句对话的基于关键词表征学习的句向量表示，包括如下步骤：

S2_1，将单句对话向量表示与其相应的词频-逆向文件频率向量表示相乘，得到矩阵；

S2_2，对矩阵的每行进行加权求和；

S2_3，对每个求和后的数值，分别除以单句对话的向量数，得到句向量表示。

6.根据权利要求1所述的结合关键词和语义理解表征的检索式回复对话方法，其特征在于所述S3中，将句向量表示输入高速通道模型，建立第一高速通道模型和/或第二高速通道模型，高速通道模型将向量构建成一张相互联通的图，并基于该图搜索某个顶点的K个最近邻。

7.根据权利要求1所述的结合关键词和语义理解表征的检索式回复对话方法，其特征在于所述预训练网络采用语言表征网络，将单句对话文本调整为语言表征网络的输入格式，当没有答复文本时，学习第一个文本的向量表示，当有答复文本时，学习第一个文本和答复文本整体文本的向量表示；语言表征网络输出整体句向量表示。

8.根据权利要求3所述的结合关键词和语义理解表征的检索式回复对话方法，其特征在于所述S6中预训练网络通过如下步骤进行训练：

S6_11，根据S2处理好的单轮对话文本，随机抽取部分对话文本作为正样本并标记1，随机抽取部分对话文本，并将其随机组合作为负样本并标记0，将正样本和负样本打乱；

S6_12，正负对话样本集合处理成预训练网络的输入格式，标记为0或1，0表示负样本，说明两句对话不匹配，1表示正样本，说明两句对话是上下文匹配的；

S6_13，将调整格式后的对话文本输入预训练网络进行训练。

9.根据权利要求1所述的结合关键词和语义理解表征的检索式回复对话方法，其特征在于所述S6中的精排包括如下步骤：

S6_21，根据S2处理好的单轮对话文本，随机抽取部分对话文本作为正样本并标记1，随机抽取部分对话，并将其随机组合作为负样本并标记0，将正样本和负样本打乱；

S6_22，通过前后对话文本获取分词信息，依据S5至S6中计算方式，计算排序特征；

S6_23，将S6_22获取的特征与S6_21获取的对应的标记，输入精排模型进行训练；

S6_24，通过训练好的精排模型，对当前查询对话文本获取检索结果排序，选取第一个回复作为当前查询对话文本的回复。

10.根据权利要求1所述的结合关键词和语义理解表征的检索式回复对话方法的系统，包括：对话语料采集模块、对话语料预处理模块、词向量转换模型训练模块、高速通道模型建立模块、精排模型训练模块，其特征在于：

所述对话语料采集模块，用于采集对话系统所需的对话语料以及匹配标注；

所述对话语料预处理模块，用于处理成单轮对话文本以及对话分词信息，将所有样本按比例分配训练集、测试集以及验证集；

所述词向量转换模型训练模块，用于获取训练集所有的对话语料之后，训练词向量转换模型；

所述高速通道模型建立模块，用于建立两种表征的高速通道模型，一种是基于关键词表征获取句向量表示后，根据训练集的对话样本，在词向量转换模型基础上获取每句的关键词向量表示，利用检索工具，初始化第一高速通道模型；另一种是基于语义理解表征获取

句向量表示后，根据训练集的对话样本，利用检索工具，初始化第二高速通道模型；

所述精排模型训练模块，根据训练集的单轮对话文本，基于各种不同的距离度量特征，训练精排模型；基于当前查询对话文本检索召回、排序，根据当前查询对话文本进行检索召回，依据召回的结果，进行精排，最后返回精排结果中的最优回复。

# 结合关键词和语义理解表征的检索式回复对话方法及系统

## 技术领域

[0001] 本发明涉及人工智能检索式回复对话领域，尤其是涉及一种结合关键词和语义理解表征的检索式回复对话方法及系统。

## 背景技术

[0002] 当前，对话系统在各个领域越来越引起人们的重视，主要是通过对话的形式让机器理解并处理人类语言的系统，其核心是模仿及抽象人与人之间沟通的方式，将对话抽象成可以建模的对话过程。而对话建模不是一个简单的任务，这是一个涉及理解、生成、交互等多个方向技术的综合实体。其对话场景的复杂性，比如客服、语音助手、闲聊等，也造就了对话系统的复杂性。

[0003] 检索式对话是一种经典的解决方案，把一个对话问题抽象为一个搜索问题，早期的对话系统都是采用这种解决方案来实现的，直到现在，工业界首先会采用检索式对话实现一些简单的对话任务。

[0004] 检索式回复模型的核心是其所使用的语义匹配算法。在目前的技术中，获取对话文本的语义表征时，往往采取RNN-based模型获取，当文本过长时，往往捕捉不到关键信息，且无法过滤掉自身冗余的信息，检索的相关回复质量不高。而单纯的使用关键词表征检索相关匹配回复，在语义上达不到流畅自然的回复效果。

## 发明内容

[0005] 为解决现有技术的不足，基于关键词和语义理解两种不同粒度的表征，检索匹配回复，实现有效丰富检索回复的内容和质量的目的，本发明采用如下的技术方案：

[0006] 结合关键词和语义理解表征的检索式回复对话方法，包括如下步骤：

[0007] S1，根据对话文本语料，得到单句对话文本和单句对话分词信息；

[0008] 采集中文对话文本语料[ $utterance_1, utterance_2, \ldots, utterance_m$ ]，拆分为单句对话文本[[$utterance_1$],[$utterance_2$],...,[$utterance_m$ ]]和分词信息，用于训练word2vec模型；

[0009] 预处理对话文本，分别将对话文本语料[$utterance_1, utterance_2, \ldots, utterance_m$]处理成单轮对话文本，获取单句对话分词信息；单轮对话文本，是以相邻的两句对话[ $utterance_i, utterance_j$ ]作为单轮的对话，其中i和j的关系满足i+1 = j，使用结巴分词将单轮对话的两句对话分词$[t_1, t_2, \cdots, t_d]$。

[0010] S2，分别获取单句对话文本和单句对话分词信息的句向量表示；

[0011] 单句对话分词信息 $[t_1, t_2, \cdots, t_d]$通过训练好的word2vec模型，获取其向量表示$[w_1, w_2, \cdots, w_d]$；

[0012] 计算单句对话分词信息$[t_1, t_2, \cdots, t_d$ ]与所有单句对话文本的TF-IDF向量表示$[b_1, b_2, \cdots, b_d]$ ；

[0013] 根据单句对话的向量表示$[w_1,w_2,\cdots,w_d]$与其对应的TF-IDF向量表示$[b_1,b_2,\cdots,b_d]$，得到该句对话的句向量表示$\mathrm{sen}=(s_1,s_2,\cdots,s_{embed_size})$，emded_size表示word2vec模型的词向量维度。

[0014] S3,根据句向量表示,分别构建第一、第二高速通道模型；

[0015] 通过S2得到的句向量表示$\mathrm{sen}=(s_1,s_2,\cdots,s_{embed_size})$，基于关键词表征学习的句向量表示，构建分层的高速通道模型HNSW_1；通过S2得到的句向量表示$\mathrm{h}=(h_1,h_2,\cdots,h_n)$，构建分层的高速通道模型HNSW_2；获取的单句对话[[$utterance_1$],[$utterance_2$],...,[$utterance_m$]],分别将每句对话$utterance_i$按照S2步骤计算每句的句向量表示$sen_i$,拼接在一起形成corpus_embedding=[[$sen_1$],[$sen_2$],...,[$sen_m$]]。

[0016] 将单句对话文本,输入到输出宽度为n的中文预训练Bert(Bidirectional Encoder Representation from Transformers--基于transformer的双向编码器)网络,得到当前对话整体隐藏的语义表征$\mathrm{h}=(h_1,h_2,\cdots,h_n)$，即基于语义理解模型Bert网络学习的query文本句向量表示；

[0017] S4,将当前查询对话文本输入第一、第二高速通道模型,得到第一、第二回复列表；

[0018] 针对当前的查询对话文本query，基于S2获取query的向量表示：$\mathrm{query\_1}=[q_1,q_2,\cdots,q_n]$；

[0019] 将获取的query文本的向量表示$\mathrm{query_1}=[q_1,q_2,\cdots,q_n]$，输入到已经建立好的HNSW_1检索模型,检索出相似的对话文本,再将相似的对话文本对应的回复返回,得到检索出的top-K回复列表$\mathrm{res_1}=[r_1,r_2,\cdots,r_K]$；

[0020] 针对当前的查询对话文本query，基于S2获取query的向量表示：$\mathrm{query\_2}=[g_1,g_2,\cdots,g_n]$；

[0021] 将获取的query文本的向量表示$\mathrm{query_2}=[g_1,g_2,\cdots,g_n]$，输入到已经建立好的HNSW_2检索模型,检索出相似的对话文本,再将相似的对话文本对应的回复返回,得到检索出的top-K的回复列表$\mathrm{res_2}=[a_1,a_2,\cdots,a_K]$。

[0022] S5,获取当前查询对话文本与第一、第二回复列表的句向量表示的距离,及词向量转换距离；

[0023] 将返回的回复列表res_1和res_2,与query建立一一对应的对话形式dataframe：

[0024] $[[\mathrm{query},r_1],[\mathrm{query},r_2],\cdots,[\mathrm{query},r_K],[\mathrm{query},a_1],[\mathrm{query},a_2],\ldots[\mathrm{query},a_K]]$；

[0025] 根据dataframe,通过S2分别计算出query与res_1和res_2中每个回复$r_i$和$a_i$的句向量表示,并计算query的句向量表示分别与res_1和res_2中每个回复$r_i$和$a_i$的句向量表示之间的最长公共字串的长度lcs、consine距离和bm25相关值；

[0026] 根据dataframe,通过训练好的word2vec模型,分别获取query的向量表示分别与res_1和res_2中每个回复$r_i$和$a_i$的向量表示,计算query与每个回复$r_i$和$a_i$之间的余弦距离word2vec_cosine,皮尔逊距离word2vec_pearson和词移距离word2vec_wmd。

[0027] S6,通过预训练模型,获取当前查询对话文本与第一、第二回复列表的匹配分数,匹配分数与S5得到的距离作为排序特征,通过精排模型得到排序结果；

[0028] 通过已经预训练好的中文网络模型Bert可以计算上下文是否匹配,将query分别

与res_1和res_2的每个回复 $r_i$ 和 $a_i$，输入到训练好的预训练网络Bert中，得到最后的匹配分数match_score具体的表示为

[0029] $\text{Score} = [score_1, score_2, \cdots, score_K, score_{K+1}, \ldots, score_{2K}]$ ；

[0030] 基于获取的各种距离作为下一步精排的特征，即将匹配分数与其对应的所述最长公共字串的长度lcs、consine距离、bm25相关值、word2vec距离作为排序特征：$[lcs, cos, bm25, word2vec_{cos}, word2vec_{pearson}, word2vec_{wmd}, match\_score]$ ，输入到训练好的精排模型LGBMRanker中，获取精排后的回复结果顺序res，选取res的最优回复作为当前query的回复。

[0031] 进一步地，所述S2中的TF-IDF=TF(x)*IDF(x)，

[0032] $\text{TF(x)} = \frac{\text{词x在该句中出现的次数}}{\text{该句所有的词数}}$ ，$\text{IDF(x)} = \frac{\text{对话的所有数量}}{\text{包含词x的所有对话数量}}$ 。

[0033] 进一步地，所述S2中，将单句对话的向量表示 $[w_1, w_2, \cdots, w_d]$ 与其对应的TF-IDF数值 $[b_1, b_2, \cdots, b_d]$ ，加权求和取平均，得到句向量表示，包括如下步骤：

[0034] S2_1，将单句对话向量表示与其相应的TF-IDF向量表示 $[b_1, b_2, \cdots, b_d]$ 相乘，得到矩阵；

[0035] S2_2，对矩阵的每行进行加权求和；

[0036] S2_3，对每个求和后的数值，分别除以d，得到一个embbed_size*1维度的句向量表示 $\text{sen} = (s_1, s_2, \cdots, s_{embed\_size})$ ，emded_size表示word2vec模型的词向量维度。

[0037] 进一步地，所述S3中，将句向量表示输入Faiss的HNSW接口模型，建立HNSW_1和/或HNSW_2，HNSW将D维空间中所有的向量构建成一张相互联通的图，并基于该图搜索某个顶点的K个最近邻。

[0038] 进一步地，所述S6的预训练网络采用输出宽度为n的Bert网络，包括如下步骤：

[0039] 将单句对话文本，调整为Bert网络的输入格式[CLS]chat[SEP]response[SEP]或[CLS]chat[SEP]，当没有答复文本response时，学习第一个文本chat的向量表示，当有答复文本response时，学习chat和response整体文本的向量表示；

[0040] Bert网络输出整体句向量表示sequence_output。

[0041] 进一步地，所述S6中的距离包括最长公共字串的长度lcs、consine距离和bm25相关值的计算，计算过程包括如下步骤：

[0042] 最长公共子串的长度lcs：有两个字符串(可能包含空格)，找出其中最长的公共连续子串，并输出其长度；

[0043] consine距离：

[0044] $\text{cosine}(\text{query}, \text{r}) = \frac{\text{query} \cdot \text{r}}{|\text{query}| \cdot |\text{r}|} = \frac{\sum x_i y_i}{\sum x_i \sum y_i}$ ，其中 $\text{query} = [x_1, \ldots, x_l]$ ，$\text{r} = [y_1, \ldots, y_l]$ 表示res_1的每个回复，其是S2计算的句向量表示；

[0045] bm25相关值，用于评价搜索词和文本之间相关性，对每个词与文档的相似度IDF×R求和，公式为：

[0046] $\text{bm25}(\text{query}, \text{r}) = \sum W_i \text{R}(\text{query}_i, \text{r})$；

[0047] 其中 $W_i = \frac{N - n(query_i) + 0.5}{n(query_i) + 0.5}$ ，$\text{R}(\text{query}_i, \text{r}) = \frac{f_i \cdot (k_1 + 1)}{f_i + K} \cdot \frac{qf_i \cdot (k_2 + 1)}{qf_i + k_2}$ ，

$K = k_1 \cdot (1 - b + b \cdot \frac{dl}{avgdl})$ ,b,k1,k2都是自定义参数,一般k1=2,k2=1,b=0.75,dl是文档长度,avgdl是平均文档长度,$f_i$是词在文档中的出现的次数,$qf_i$是词在query中出现的次数,N是所有的文档数量,$n(query_i)$是包含词$query_i$的文档数量;

[0048] res_2的每个回复a公式与r相似,其是S2步骤计算的句向量表示。

[0049] 进一步地,所述S6中的word2vec距离包括余弦距离、皮尔逊距离和词移距离的计算,计算过程包括如下步骤:

[0050] 余弦距离:$\mathrm{cosine}(query,r) = \frac{query \cdot r}{|query| \cdot |r|} = \frac{\sum x_i y_i}{\sum x_i \sum y_i}$ ,其中 $query = [x_1,\ldots,x_l]$ , $r = [y_1,\ldots,y_l]$ 表示res_1的每个回复,其是S2至S5步骤计算的句向量表示;

[0051] 皮尔逊距离:$\mathrm{pearson}(query,r) = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sum(X-\mu_X)^2 \sum(Y-\mu_Y)^2}$ ,其中$X$ 和$Y$ 分别是基于S2至S5获取的句向量表示,E表示序列的期望,μ表示序列的均值;

[0052] 词移距离:是度量两个文本之间距离的一种方式(方法),用于判断两个文本之间的相似度,WMD是通过将一个文本中包含的词语“移动”(travel)到另一个文本中的词语,这个“移动”过程产生的距离总和的最小值作为词移距离,首先根据S2至S5步骤计算的句向量表示,然后按照以下方式计算出两个文本向量之间的词移距离:

[0053] $\min \sum T_{ij} ||x_i - y_j||$;

[0054] 使得 $\sum_j T_{ij} = d_i$ 且 $\sum_i T_{ij} = d'_j$;

[0055] 其中$T_{ij} \geq 0$表示query中第i个词到回复第j个词移动的权重, $d_i$表示query中第i个词在query所有词的权重, $d'_i$表示回复第i个词在回复中所有词的权重;

[0056] res_2的每个回复a公式与r相似,其是S2步骤计算的句向量表示。

[0057] 进一步地,所述S6中预训练网络的训练,包括如下步骤:

[0058] S6_11,根据S2处理好的单轮对话样本$[chat_1, chat_2]$,随机抽取部分对话文本作为正样本即 $[chat_1, chat_2, 1]$,随机抽取部分对话文本,并将其随机组合作为负样本$[chat_1, chat_3, 0]$,将正样本和负样本打乱之后,保存;

[0059] S6_12,根据S6_11获取的正负对话样本集合处理成与训练网络Bert的输入格式:[CLS] $chat_1$[SEP] $chat_2$[SEP],label为0或1,0表示负样本,说明两句对话不匹配,1表示正样本,说明两句对话是上下文匹配的;

[0060] S6_13,将调整格式后的对话文本[CLS] $chat_1$[SEP] $chat_2$[SEP]输入预训练网络Bert,进行有监督的训练,并保存模型。

[0061] 进一步地,所述S6中的精排包括如下步骤:

[0062] S6_21,根据S2处理好的单轮对话文本 $[chat_1, chat_2]$,随机抽取部分对话文本作为正样本即 $[chat_1, chat_2, 1]$,随机抽取部分对话,并将其随机组合作为负样本$[chat_1, chat_3, 0]$,将正样本和负样本打乱;

[0063] S6_22,通过前后对话文本获取分词信息 $chat_1 = [c_1, c_2, \cdots, c_d]$ 和 $chat_2 = [l_1, l_2, , \cdots, l_d]$,依据S5至S6中计算方式,计算排序特征:

$[lcs, cos, bm25, word2vec_{cos}, word2vec_{pearson}, word2vec_{wmd}, match\_score]$;

[0064] S6_23，将S6_22获取的特征与S6_21获取的对应的label，输入精排模型LGBMranker进行有监督的训练，得到已经训练好的精排模型LGBMRanker；

[0065] S6_24，通过训练好的精排模型LGBMRanker，对query获取检索结果排序，输出一个排好序的检索结果列表，选取第一个回复作为query的回复。其输入就是将S6_21的正负样本按照S6_22所计算的特征形式输入到LGBMRanker模型训练。

[0066] 结合关键词和语义理解表征的检索式回复对话系统，包括：对话语料采集模块、对话语料预处理模块、训练word2vec模型、HNSW模型、训练精排模型；

[0067] 所述对话语料采集模块，用于采集对话系统所需的对话语料以及匹配标注；

[0068] 所述对话语料预处理模块，用于处理成单轮对话文本以及对话分词信息，将所有样本按比例分配训练集、测试集以及验证集；

[0069] 所述word2vec模型，获取训练集所有的对话语料之后，训练word2vec模型；

[0070] 所述HNSW模型，建立了两种表征的HNSW模型，一种是基于关键词表征获取句向量表示后，根据训练集的对话样本，在word2vec模型基础上获取每句的关键词向量表示，利用Faiss检索开源工具，初始化HNSW_1模型；另一种是基于语义理解表征获取句向量表示后，根据训练集的对话样本，利用Faiss检索开源工具，初始化HNSW_2模型；

[0071] 训练精排模型，根据训练集的单轮对话样本，基于各种不同的距离度量特征，训练精排模型；基于查询对话文本query检索召回、排序，根据当前的query进行检索召回，依据召回的结果，进行精排，最后返回精排结果的最优回复。

[0072] 本发明的优势和有益效果在于：

[0073] 1、本发明结合了两种不同粒度的向量表征形式，基于关键词表征和语义理解表征的两种不同方式检索相关回复，使检索回复在关键词上具有相关性，对话的主题意识不会脱离，在语义理解上具有很近的相似匹配，提高了检索式回复的流畅性和自然性，能够提高最终检索式回复对话质量；

[0074] 2、本发明中采取了中文预训练模型Bert网络模型获取句向量表征，优点在于不仅理解句意，并且排除了词向量加权引起的误差；

[0075] 3、本发明采取了BertForSequenceClassification网络模型在自己的单轮对话上训练分类任务——对话是否匹配的任务，通过微调，学习到了BertForSequenceClassification中线性层+激活函数的权重，可以做到后续对话的分类任务；

[0076] 4、本发明后续的精排过程中采取了各种不同的距离作为特征，分别是$[lcs, cos, bm25, word2vec_{cos}, word2vec_{pearson}, word2vec_{wmd}, match_score]$，输入到模型LGMRanker中，直接预测检索出的相关回复的相对顺序，返回一个排好序的列表。

## 附图说明

[0077] 图1是本发明的方法流程图。

[0078] 图2是本发明中HNSW构建的向量联通结构示意图。

[0079] 图3是本发明中Bert网络的结构示意图。

[0080] 图4是本发明中Encoder结构示意图。

[0081] 图5是本发明的系统结构示意图。

[0082] 图6是本发明另一种结合关键词和语义理解表征的检索式回复对话装置的结构图。

## 具体实施方式

[0083] 以下结合附图对本发明的具体实施方式进行详细说明。应当理解的是，此处所描述的具体实施方式仅用于说明和解释本发明，并不用于限制本发明。

[0084] 如图1所示，一种结合关键词和语义理解表征的检索式回复对话方法，包括如下步骤：

[0085] S1、根据对话文本语料，得到单句对话文本和单句对话分词信息；

[0086] 采集中文对话文本语料[$utterance_1, utterance_2, \ldots, utterance_m$]，分别拆开获取所有的单句对话文本[[ $utterance_1$],[$utterance_2$],...,[$utterance_m$]]以及分词信息后，训练word2vec模型，并保存word2vec模型；

[0087] 预处理对话文本，分别将对话预料[$utterance_1, utterance_2, \ldots, utterance_m$]处理成单轮对话文本，即以相邻的两句对话作为单轮的对话[ $utterance_i, utterance_j$]，其中i和j的关系满足i+1 = j，使用结巴分词将单轮对话的两句对话分词，获取每句对话分词信息 $[t_1, t_2, \cdots, t_d]$；

[0088] 比如text = "我来到北京清华大学"

[0089] [ $t_1, t_2, \cdots, t_d$] = jieba.cut(text, cut_all=True)= [我，来到，北京，清华，清华大学，华大，大学]。

[0090] S2、分别获取单句对话文本和单句对话分词信息的句向量表示；

[0091] 获取的每句对话分词信息 $[t_1, t_2, \cdots, t_d]$，通过S1训练的word2vec模型获取每句对话分词信息的向量表示 $[w_1, w_2, \cdots, w_d]$，i∈d；

[0092] $W_i$=w2v_model.wv.get_vector( $t_i$)；

[0093] 计算每句对话分词信息 $(t_1, t_2, \cdots, t_d)$与所有单句对话文本的TF-IDF的数值 $(b_1, b_2, \cdots, b_d)$，其中TF-IDF =TF(x)*IDF(x)，TF(x) =$\frac{\text{词x在该句中出现的次数}}{\text{该句所有的词数}}$，IDF(x)= $\frac{\text{对话的所有数量}}{\text{包含词x的所有对话数量}}$。

[0094] 每句对话的句向量表示，通过该句对话的向量表示 $(w_1, w_2, \cdots, w_d)$和其相应的TF-IDF $(b_1, b_2, \cdots, b_d)$加权求和取平均( $\frac{\sum b_i w_i}{d}$)而得，具体地表示为：

[0095] $\text{sen} = (s_1, s_2, \cdots, s_{embed\_size})$；

[0096] 计算过程可表示为：

[0097] 每句对话的句向量表示为，相应的TF-IDF向量表示为，两者相乘之后得，然后按行求每行得加和可得，再对每个求和后得数值分别除以d，可得最后的表示结果sen，是一个embbed_size*1维度的向量；

[0098] 其中 $W_i$表示第i个分词的向量表示，其维度为embed_size，所以每句对话的句向量表示其实为矩阵； $b_i$表示第i个分词的TF-IDF数值，所以每句对话分词对应的TF-IDF是一维向量。emded_size为word2vec词向量维度。

[0099] S3、根据句向量表示，分别构建第一、第二高速通道模型；

[0100] 依据获取的每句对话的向量表示 $\mathrm{Sen}=(s_1,s_2,\cdots,s_{embed_size})$后，输入Faiss的HNSW接口模型，建立HNSW_1模型，并保存HNSW_1模型，即基于关键词表征学习句向量表示，建立HNSW_1模型；

[0101] S1获取的单句对话[[ $utterance_1$],[$utterance_2$],...,[$utterance_m$]]按照S2的步骤计算每句的句向量表示corpus_embedding=[[ $sen_1$],[$sen_2$],...,[$sen_m$]]，输入Faiss的HNSW接口模型：

[0102] dim = embed_size

[0103] index = faiss.IndexHNSWFlat(dim, m, measure) # build the index

[0104] index.add(corpus_embedding) # add vectors to the index

[0105] HNSW(Hierarchical Navigable Small World--分层的高速通道)是把D维空间中所有的向量构建成一张相互联通的图，并基于这张图搜索某个顶点的K个最近邻，如图2所示；

[0106] 第0层中包含图中所有节点；

[0107] 向上节点数依次减少，遵循指数衰减概率分布；

[0108] 建图时新加入的节点由指数衰减概率函数得出该点最高投影到第几层；

[0109] 从最高的投影层向下的层中该点均存在；

[0110] 搜索时从上向下依次查询；

[0111] 初始化模型时，需要的是将所有单轮对话表示为学习的向量表示后，按照列表的形式加入模型中，会自动按照图的形式建图。

[0112] 将每句对话文本，输入到输出宽度为n的中文预训练Bert(Bidirectional Encoder Representation from Transformers--基于transformer的双向编码器)网络，如图3、4所示，得到当前对话整体隐藏的语义表征 $\mathrm{h}=(h_1,h_2,\cdots,h_n)$，即基于语义理解模型Bert学习的query文本句向量表示。

[0113] Bert(Bidirectional Encoder Representation from Transformers)是一个预训练的语言表征模型。其输入格式表示为[CLS]chat[SEP]response[SEP]或[CLS]chat[SEP]，第二个答复文本response如果没有的话，那就是在学习第一个文本的向量表示，如果有response的话，那就是在学习chat和response整体文本的向量表示；

[0114] 针对Bert模型的输出sequence_output，一般情况下，使用sequence_output表示整体句向量表示。

[0115] 通过S2获取的每句对话的向量表示 $\mathrm{h}=(h_1,h_2,\cdots,h_n)$，输入Faiss的HNSW接口模型，建立HNSW_2模型，并保存HNSW_2模型；

[0116] S1获取的单句对话[[$utterance_1$],[$utterance_2$],...,[$utterance_m$]]按照S2步骤计算每句的句向量表示corpus_embedding=[[ $sen_1$],[$sen_2$],...,[$sen_m$]]，

[0117] 计算过程是分别将每个按照S2步骤计算句向量表示为，拼接在一起就形成了corpus_embedding。

[0118] 输入Faiss的HNSW接口模型：

[0119] dim = hiddien_size

[0120] index = faiss.IndexHNSWFlat(dim, m, measure) # build the index

[0121] index.add(corpus_embedding) # add vectors to the index。

[0122] S4、将当前查询对话文本输入第一、第二高速通道模型，得到第一、第二回复列表；

[0123] 针对当前的query对话文本，基于S2至S5的计算获取query文本的向量表示，具体的表示为：

[0124] $query\_1 = [q_1, q_2, \cdots, q_n]$；

[0125] 将获取的query文本的向量表示 $query\_1 = [q_1, q_2, \cdots, q_n]$，输入到已经建立好的HNSW_1检索模型，检索出top-K的回复列表 $res\_1 = [r_1, r_2, \cdots, r_K]$；

[0126] 建立好的HNSW_1模型为index_1，输入S9学习到的query的向量表示，检索出与query相似的对话文本，然后将相似的对话文本对应的回复返回作为此时的检索结果，即res_1。

[0127] 比如：query：我下周要去爬山

[0128] 检索出和query相似的文本有：1)我下星期要去爬山，2)我下星期登山等。

[0129] 然后将S2处理成的单轮的对话[$utterance_i, utterance_j$]中找到相似文本对应的下一句对话作为回复返回

[0130] 1)和谁2)哪个山...

[0131] 针对当前的query对话文本，计算获取query文本的向量表示，具体表示为：

[0132] $query\_2 = [g_1, g_2, \cdots, g_n]$；

[0133] 将获取的query文本的向量表示 $query\_2 = [g_1, g_2, \cdots, g_n]$，输入到已经建立好的HNSW_2检索模型，检索出top-K的回复列表 $res\_2 = [a_1, a_2, \cdots, a_K]$；

[0134] 建立好的HNSW_2模型为index_2，输入S12学习到的query向量表示，检索出与query相似的对话文本，然后将相似的对话文本对应的回复返回作为此时的检索结果，即res_1。

[0135] 比如：query：我晚上睡不着

[0136] 检索出和query相似的文本有：1)我最近天天睡不着 2)我晚上睡的不舒服等

[0137] 然后将S2处理成的单轮的对话[$utterance_i, utterance_j$]中找到相似文本对应的下一句对话作为回复返回

[0138] 1)怎么了2)陪你...。

[0139] S5、获取当前查询对话文本与第一、第二回复列表的句向量表示的距离，及词向量转换距离；

[0140] 对返回的回复列表res_1和res_2，与当前的query文本建立一一对应的对话形式dataframe，具体地：

[0141] $[[query, r_1], [query, r_2], \cdots, [query, r_K], [query, a_1], [query, a_2], \ldots [query, a_K]]$；

[0142] 比如：[‘我下周要去爬山’，‘和谁’]这样的文本；

[0143] 针对获取的dataframe，按照S2步骤分别计算出query与 $r_i$和 $a_i$的句向量表示后，计算query与每个回复$r_i$和 $a_i$之间的最长公共字串的长度lcs、consine距离和bm25相关值；

[0144] 最长公共子串的长度lcs：有两个字符串(可能包含空格)，请找出其中最长的公共连续子串，输出其长度。例如：输入：“我下周要去爬山”和“我下星期要去登山”，则lcs的结

果就是5。

[0145] consine距离：$\mathrm{cosine}(\mathrm{query},\mathrm{r})=\frac{\mathrm{query}\cdot\mathrm{r}}{|\mathrm{query}|\cdot|\mathrm{r}|}=\frac{\sum x_i y_i}{\sum x_i \sum y_i}$，其中 $\mathrm{query}=[x_1,\ldots,x_l]$，$\mathrm{r}=[y_1,\ldots,y_l]$，其是前面S2至S5步骤计算的句向量表示；

[0146] bm25是一种用来评价搜索词和文本之间相关性的算法。

[0147] 简而言之就是对每个词与文档的相似度IDF×R的求和。公式为：

[0148] $$\mathrm{bm25}(\mathrm{query},\mathrm{r})=\sum W_i R(\mathrm{query}_i,\mathrm{r})$$

[0149] 其中 $W_i=\frac{N-n(query_i)+0.5}{n(query_i)+0.5}$，$\mathrm{R}(\mathrm{query}_i,\mathrm{r})=\frac{f_i\cdot(k_1+1)}{f_i+K}\cdot\frac{qf_i\cdot(k_2+1)}{qf_i+k_2}$；

[0150] 其中$K=k_1\cdot(1-b+b\cdot\frac{dl}{avgdl})$，b,k1,k2都是自己设置的参数，一般k1=2,k2=1,b=0.75。dl是文档长度，avgdl是平均文档长度，$f_i$是词在文档中的出现的次数，$qf_i$是词在query中出现的次数，N是所有的文档数量，$n(query_i)$是包含词$query_i$的文档数量。

[0151] 针对获取的dataframe，基于S1训练的word2vec模型，分别获取query的向量表示和每个回复 $r_i$和 $a_i$的向量表示，进而计算query与每个回复 $r_i$和$a_i$之间的余弦距离word2vec_cosine，皮尔逊距离word2vec_pearson和词移距离word2vec_wmd；

[0152] pearson距离：$\mathrm{pearson}(\mathrm{query},\mathrm{r})=\frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sum(X-\mu_X)^2\sum(Y-\mu_Y)^2}$，其中 $X$和 $Y$分别是基于步骤S2获取的句向量表示，E表示序列的期望，μ表示序列的均值；

[0153] 词移距离(WMD)是度量两个文本之间距离的一种方式(方法)，用于判断两个文本之间的相似度。WMD是通过将一个文本中包含的词语“移动”(travel)到另一个文本中的词语，这个“移动”过程产生的距离总和的最小值作为词移距离。首先按照S2步骤计算的句向量表示，然后按照以下方式计算出两个文本向量之间的词移距离：

[0154] $$\min\sum T_{ij}||x_i-y_j||;$$

[0155] 使得$\sum_j T_{ij}=d_i$ 且$\sum_i T_{ij}=d_j'$；

[0156] 其中$T_{ij}\geq 0$表示query文本中第i个词到回复第j个词移动的权重；$d_i$表示query中第i个词在query所有词的权重；同样的，$d_i'$表示回复第i个词在回复中所有词的权重。

[0157] S6、通过预训练模型，获取当前查询对话文本与第一、第二回复列表的匹配分数，匹配分数与S5得到的距离作为排序特征，通过精排模型得到排序结果；

[0158] 利用已经预训练好的中文网络模型Bert可以计算上下文是否匹配，分别将query与每个回复 $r_i$和 $a_i$，输入到训练好的Bert模型中，得到最后的匹配分数match_score具体的表示为 $\mathrm{Score}=[score_1,score_2,\cdots,score_K,score_{K+1},\ldots,score_{2K}]$；

[0159] S6_11：根据S2已经处理好的单轮对话样本 $[chat_1,chat_2]$，从里面随机抽取一些对话文本作为正样本即 $[chat_1,chat_2,1]$，随机抽取一些对话，并将其随机组合作为负样本$[chat_1,chat_3,0]$，将正样本和负样本打乱之后，保存。

[0160] S6_12：根据S6_1获取的对话样本集合处理成Bert的输入格式具体的表示为[CLS] $chat_1$ [SEP] $chat_2$[SEP]，label则为0或1，0表示负样本，说明两句对话是不匹配的意思；1表示正样本，说明两句对话是上下文匹配的意思。

[0161] S6_13:将处理好的输入[CLS] $chat_1$[SEP] $chat_2$[SEP]输入到Bert预训练模型中,进行有监督的训练,最后保存模型。

[0162] 基于S5获取的各种距离作为下一步精排的特征,即 $[lcs, cos, bm25, word2vec_{cos}, word2vec_{pearson}, word2vec_{wmd}, match\_score]$,输入到训练好的精排模型LGBMRanker中,获取精排后的回复结果顺序res,选取res的第一个回复作为当前query的回复。

[0163] S6_21:根据S2处理好的单轮对话文本 $[chat_1, chat_2]$,从里面随机抽取一些对话文本作为正样本即 $[chat_1, chat_2, 1]$,随机抽取一些对话,并将其随机组合作为负样本 $[chat_1, chat_3, 0]$,将正样本和负样本打乱之后,保存。

[0164] S6_22:针对对话前后获取的分词信息 $chat_1 = [c_1, c_2, \cdots, c_d]$ 和 $chat_2 = [l_1, l_2, , \cdots, l_d]$,计算相应的距离作为排序的特征,具体的特征有:$[lcs, cos, bm25, word2vec_{cos}, word2vec_{pearson}, word2vec_{wmd}, match\_score]$

[0165] S6_23:将获取的特征和获取的对应的label,输入到LGBMranker模型进行有监督的训练,最后得到已经训练好的精排模型LGBMRanker。

[0166] S6_24:LGBMRanker模型是一种排序模型,主要是针对query文本获取的检索结果的排序,输出一个排好序的检索结果列表。其输入就是将正负样本按照计算的特征形式输入到LGBMRanker模型训练。

[0167] 综上所述,本实施提供的方法,通过结合关键词表征和语义理解表征,提高了检索式回复的流畅性和自然性,能够提高最终检索式回复对话质量。

[0168] 如图5所示,一种结合关键词和语义理解表征的检索式回复对话系统,包括:

[0169] 对话语料采集模块,用于采集对话系统所需的对话语料以及匹配标注;

[0170] 对话语料预处理模块,用于处理成单轮对话文本以及对话分词信息,将所有样本按比例分配训练集、测试集以及验证集;

[0171] word2vec模型训练模块,用于获取训练集所有的对话语料之后,训练word2vec模型;

[0172] HNSW模型建立模块,用于整体模型中建立了两种表征的HNSW模型,一种是基于关键词表征获取句向量表示之后,根据训练集的对话样本,在word2vec模型基础上获取每句的关键词向量表示,利用Faiss检索开源工具,初始化HNSW_1模型;另一种是基于语义理解表征获取句向量表示之后,根据训练集的对话样本,利用Faiss检索开源工具,初始化HNSW_2模型;

[0173] 精排模型训练模块,用于根据训练集的单轮对话样本,基于各种不同的距离度量特征,训练精排模型;基于query检索召回、排序,根据当前的query进行检索召回,依据召回的结果,进行精排,最后返回精排结果的第一个回复。

[0174] 与前述结合关键词和语义理解表征的检索式回复对话方法的实施例相对应,本发明还提供了结合关键词和语义理解表征的检索式回复对话装置的实施例。

[0175] 参见图6,本发明实施例提供的一种结合关键词和语义理解表征的检索式回复对话装置,包括一个或多个处理器,用于实现上述实施例中的结合关键词和语义理解表征的检索式回复对话方法。

[0176] 本发明结合关键词和语义理解表征的检索式回复对话装置的实施例可以应用在任意具备数据处理能力的设备上，该任意具备数据处理能力的设备可以为诸如计算机等设备或装置。装置实施例可以通过软件实现，也可以通过硬件或者软硬件结合的方式实现。以软件实现为例，作为一个逻辑意义上的装置，是通过其所在任意具备数据处理能力的设备的处理器将非易失性存储器中对应的计算机程序指令读取到内存中运行形成的。从硬件层面而言，如图6所示，为本发明结合关键词和语义理解表征的检索式回复对话装置所在任意具备数据处理能力的设备的一种硬件结构图，除了图6所示的处理器、内存、网络接口、以及非易失性存储器之外，实施例中装置所在的任意具备数据处理能力的设备通常根据该任意具备数据处理能力的设备的实际功能，还可以包括其他硬件，对此不再赘述。

[0177] 上述装置中各个单元的功能和作用的实现过程具体详见上述方法中对应步骤的实现过程，在此不再赘述。

[0178] 对于装置实施例而言，由于其基本对应于方法实施例，所以相关之处参见方法实施例的部分说明即可。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发明方案的目的。本领域普通技术人员在不付出创造性劳动的情况下，即可以理解并实施。

[0179] 本发明实施例还提供一种计算机可读存储介质，其上存储有程序，该程序被处理器执行时，实现上述实施例中的结合关键词和语义理解表征的检索式回复对话方法。

[0180] 所述计算机可读存储介质可以是前述任一实施例所述的任意具备数据处理能力的设备的内部存储单元，例如硬盘或内存。所述计算机可读存储介质也可以是任意具备数据处理能力的设备的外部存储设备，例如所述设备上配备的插接式硬盘、智能存储卡(Smart Media Card，SMC)、SD卡、闪存卡(Flash Card)等。进一步的，所述计算机可读存储介质还可以既包括任意具备数据处理能力的设备的内部存储单元也包括外部存储设备。所述计算机可读存储介质用于存储所述计算机程序以及所述任意具备数据处理能力的设备所需的其他程序和数据，还可以用于暂时地存储已经输出或者将要输出的数据。

[0181] 以上实施例仅用以说明本发明的技术方案，而非对其限制；尽管参照前述实施例对本发明进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述实施例所记载的技术方案进行修改，或者对其中部分或者全部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发明实施例技术方案的范围。

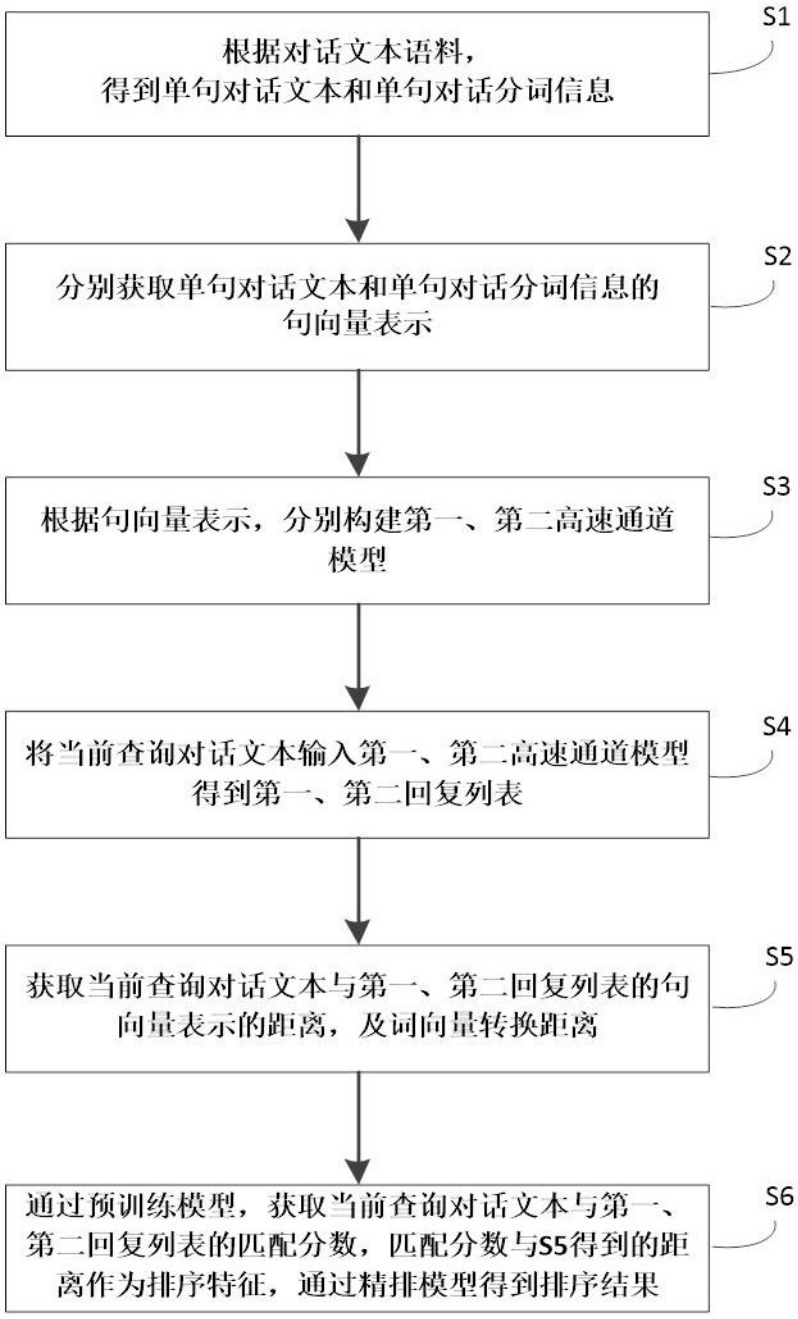

S1
根据对话文本语料，
得到单句对话文本和单句对话分词信息
S2
分别获取单句对话文本和单句对话分词信息的
句向量表示
S3
根据句向量表示，分别构建第一、第二高速通道
模型
S4
将当前查询对话文本输入第一、第二高速通道模型
得到第一、第二回复列表
S5
获取当前查询对话文本与第一、第二回复列表的句
向量表示的距离，及词向量转换距离
S6
通过预训练模型，获取当前查询对话文本与第一、
第二回复列表的匹配分数，匹配分数与S5得到的距
离作为排序特征，通过精排模型得到排序结果

图1

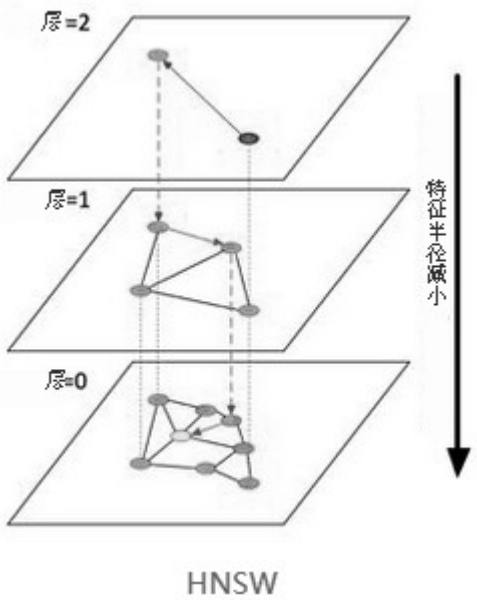

层=2
层=1
层=0
特征半径减小
HNSW

图2

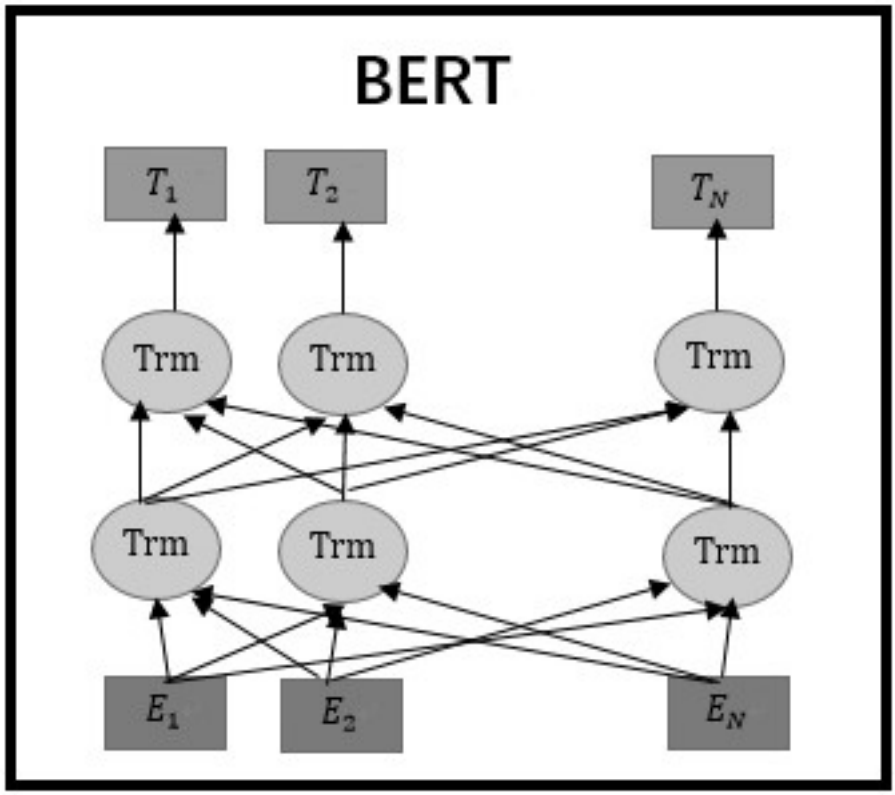

BERT
$T_1$
$T_2$
$T_N$
Trm
Trm
Trm
Trm
Trm
Trm
$E_1$
$E_2$
$E_N$

图3

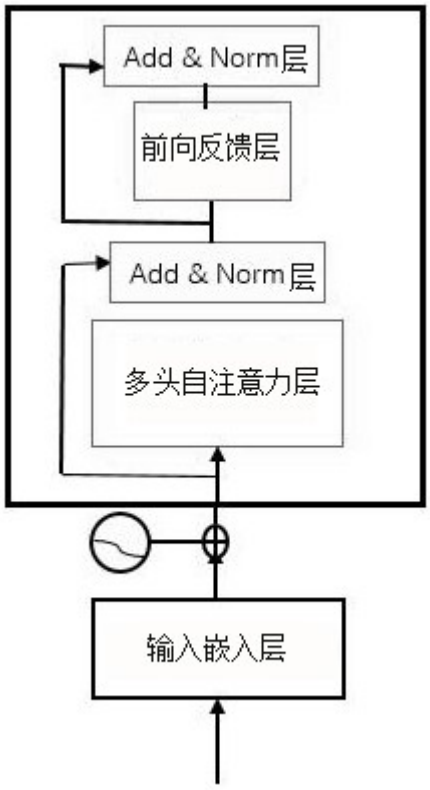



图4

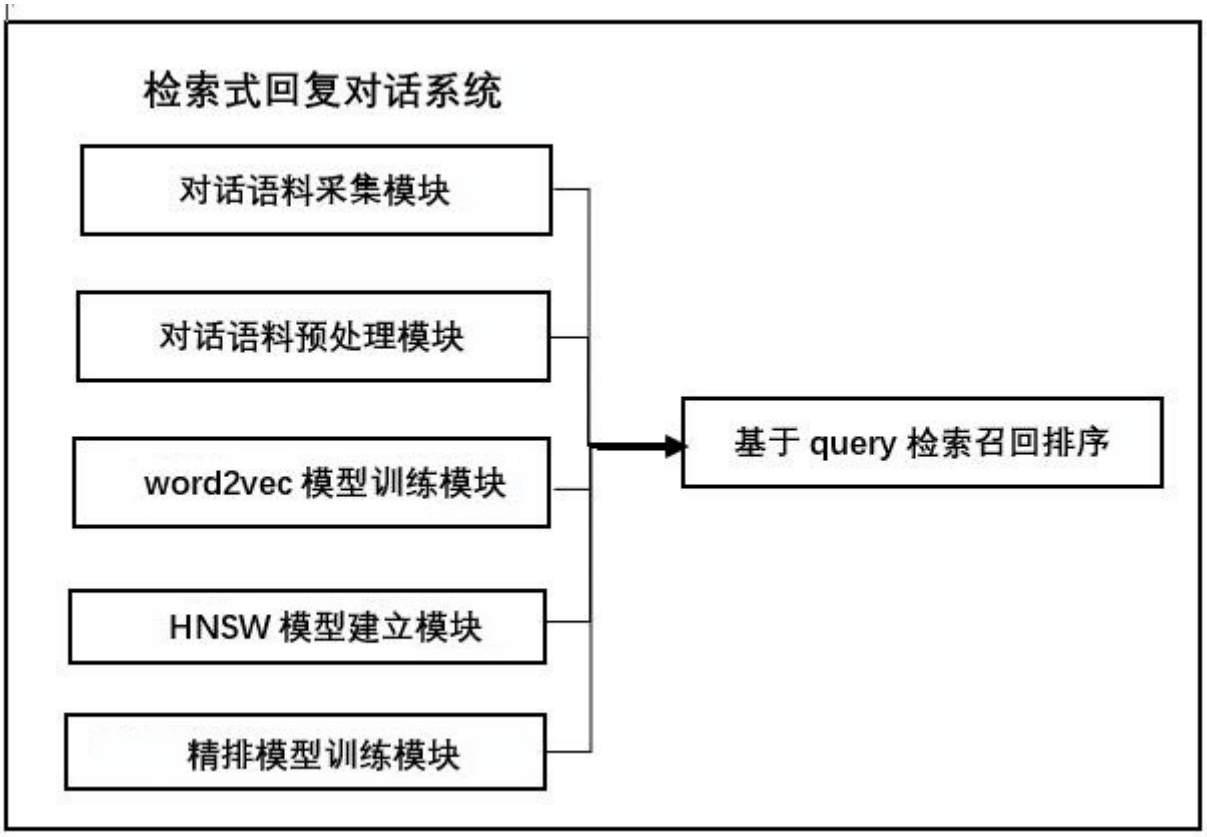

检索式回复对话系统
对话语料采集模块
对话语料预处理模块
word2vec 模型训练模块
HNSW 模型建立模块
精排模型训练模块
基于 query 检索召回排序

图5

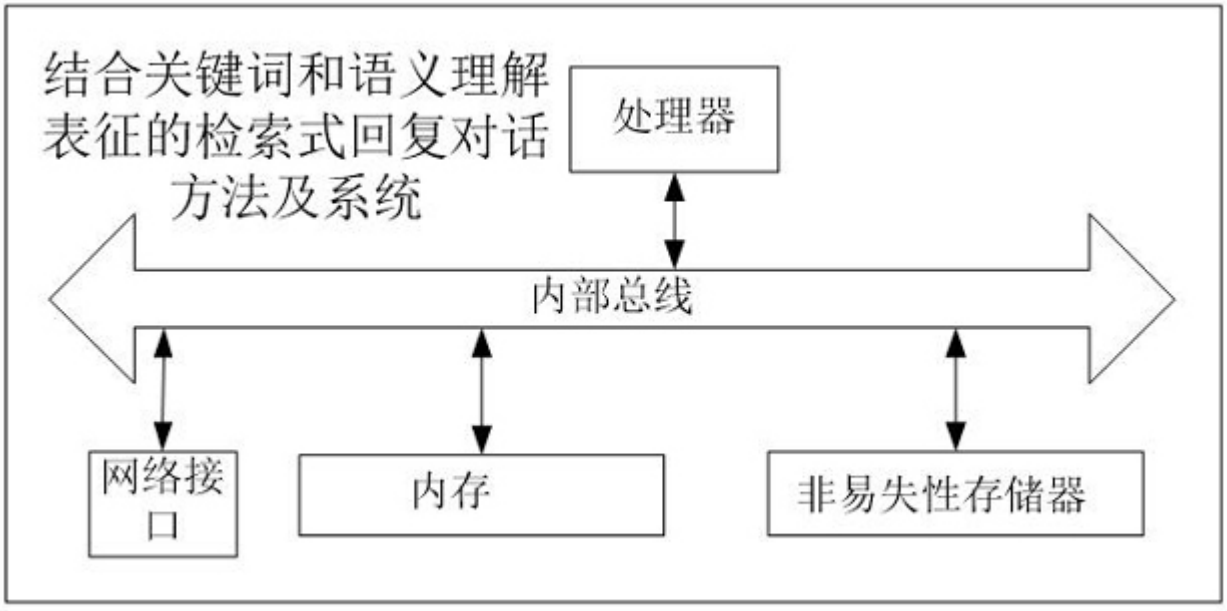

结合关键词和语义理解表征的检索式回复对话方法及系统
处理器
内部总线
网络接口
内存
非易失性存储器

图6