基于全局知识预训练知识库问答系统的研究

摘要

早在计算机和人工智能萌芽时期，“图灵测试”就已将问答系统设为检验人工智能是否达到人类智能的判断标准。近年来，大数据与人工智能技术大放异彩，实现问答系统依然是人工智能自然语言处理的主要发展方向和长期目标之一。虽然人工智能在图像识别等方面表现不错，问答系统的前景也十分广阔，但是其技术和成果至今依然没能达到令人满意的地步。

随着网络知识库的体量日益增长，利用知识库数据量的优势开发问答系统已是必然的发展方向。知识库，比如FreeBase和Wikidata，其结构化的知识包含了很多问题的答案，但知识库查询语言如SPARQL，需要熟悉该查询语言的语法和知识库的实体词汇才能正确使用。相比之下，知识库问答系统可以直接处理自然语言问题，对用户更加友好。

自然语言处理和答案信息检索是知识库问答系统建模的关键。基于知识库的问答系统传统的模型可分为两类：语义解析，信息检索。本论文基于信息检索模型基本原理，建立知识库问答系统模型。获取问题和答案的分布式表达，即词嵌入和知识库嵌入，并对嵌入之间的关系进行研究，通过多层神经网络，提取嵌入中的特征信息，从而确立答案得分，并用成对排序算法训练候选答案排序。最终，模型根据候选答案的得分序列决定正确答案。

在问题部分，将问题嵌入通过双向长短期记忆网络提取问题单词的时序信息，确保每个单词都包含了之前和之后单词的信息。在答案部分，将答案嵌入分成主体、关系、客体三部分，用注意力机制对问题的单词嵌入针对答案的三部分分别计算注意力权重，从而提取问题对于答案三部分的不同特征。同时，利用知识库中已有的全局知识信息帮助候选答案的选取和正负样本集的生成。将知识库嵌入使用TransE算法预训练，使原本随机初始化的知识库嵌入处于有序状态。最后将问题和答案组合，计算得分函数，利用正负样本成对排序学习算法，对该模型的各层神经网络进行训练。

在实验中，我们使用SimpleQuestionsWikidata数据集和Wikidata知识库，建立了本地数据库，生成了正负样本集，并以此训练模型。完成训练后，我们用验证集对模型进行正确率统计。此外，我们还开发了知识库问答系统应用，为本模型提供使用接口。

**关键词：**知识库问答，词嵌入，知识库嵌入，双向长短期记忆网络，注意力机制，全局知识信息

QUESTION ANSWERING OVER KNOWLEDGE BASE WITH KNOWLEDGE BASE EMBEDDINGS PRE-TRAINED BY GLOBAL KNOWLEDGE

**Abstract**

As early as the budding of computer and artificial intelligence technology, the "Turing Test" has used the QA system to judge whether the artificial intelligence meets the human intelligence. In recent years, big data and artificial intelligence technology have become more and more brilliant. The realization of the QA system is still one of the main directions and long-term goals of natural language processing. However, although artificial intelligence works well in image recognition, its achievements in QA system have still not been satisfying.

The form and content of the QA system are different depending on the scenarios. In chatting scenarios, there are applications like QQ chat robot; in task-based QA scenarios, there are applications like Apple Siri. According to the source of the answer, QA system can be divided into two categories: knowledge base based QA and document-based QA. The modeling of different QA system is totally different.

As the volume of the network knowledge base grows, it is natural to take the advantage of the knowledge base to develop QA system. Knowledge bases, such as FreeBase and Wikidata, have structured knowledge that contains answers to many questions, but knowledge base query language such as SPARQL is not easy to use. Users have to be familiar with the syntax of the query language as well as the entity vocabulary of the knowledge base. In contrast, the KBQA system can directly deal with natural language questions, which is user-friendly.

Natural language processing and information retrieve is the core of KBQA model. There are two main KBQA models: semantic parsing and information retrieval. We attempt to establish a KBQA model based on the information retrieval model. We obtain the embeddings of questions and answers, trying to find out the connection between the embeddings through neural networks. Then we establish a score function, using the pairwise ranking loss to train the model.

First, we obtain the embeddings of questions and answers. Second, we obtain the forward hidden state and the backward hidden state of question embeddings though BiLSTM networks. Then, the answer embeddings are divided into three parts: subject, relationship and object. The attention mechanism is used to give different attention weights to question embeddings according to different parts of answer. Additionally, we use the global knowledge information in knowledge base to select candidate answers and generate positive and negative training sets. The knowledge base embeddings is also pre-trained by TransE, so that the randomly initialized knowledge base embeddings can achieve some global knowledge information. Finally, the question and the answer embeddings are combined to generate a score. We use pairwise ranking loss to train out model, so that the score of correct answer can be higher than the wrong answer.

We will use the SimpleQuestionsWikidata dataset and the Wikidata knowledge base to train out model. In addition, we will develop a KBQA application, providing a user-friendly interface to get access to our model.

**Key words:** Knowledge Base Question Answering, Word Embeddings, Knowledge Base Embeddings, Bi-directional Long Short-Term Memory Network, Attention Mechanism, Global Knowledge Information

目录

[摘要 I](#_Toc9877051)

[Abstract III](#_Toc9877052)

[目录 V](#_Toc9877053)

[1 绪论 1](#_Toc9877054)

[1.1 课题的研究背景与意义 1](#_Toc9877055)

[1.2 国内外研究现状 1](#_Toc9877056)

[1.2.1 基于知识库的问答 1](#_Toc9877057)

[1.2.2 基于文档的问答 2](#_Toc9877058)

[1.3 本论文的主要内容和章节安排 2](#_Toc9877059)

[2 相关技术介绍 4](#_Toc9877060)

[2.1 知识库和数据集 4](#_Toc9877061)

[2.1.1 知识库 4](#_Toc9877062)

[2.1.2 数据集 4](#_Toc9877063)

[2.2 神经网络 5](#_Toc9877064)

[2.2.1 神经网络的学习 6](#_Toc9877065)

[2.2.2 损失函数与梯度下降 7](#_Toc9877066)

[2.2.3 排序学习 9](#_Toc9877067)

[2.2.4 TensorFlow 10](#_Toc9877068)

[2.3 词的向量表示 10](#_Toc9877069)

[2.4 双向长短期记忆网络 11](#_Toc9877070)

[2.5 注意力机制 13](#_Toc9877071)

[3 知识库问答系统模型 15](#_Toc9877072)

[3.1 词嵌入与知识库嵌入 16](#_Toc9877073)

[3.1.1 嵌入层设计思路 16](#_Toc9877074)

[3.1.2 嵌入层的实现方法 16](#_Toc9877075)

[3.2 双向长短期记忆网络 17](#_Toc9877076)

[3.2.1 双向长短期记忆网络层的设计思路 17](#_Toc9877077)

[3.2.2 双向长短期记忆网络层的实现方法 18](#_Toc9877078)

[3.3 注意力机制 18](#_Toc9877079)

[3.3.1 注意力机制层的设计思路 18](#_Toc9877080)

[3.3.2 注意力机制层的实现方法 19](#_Toc9877081)

[3.4 全局知识信息 20](#_Toc9877082)

[3.5 训练 21](#_Toc9877083)

[3.5.1 得分函数和损失函数 21](#_Toc9877084)

[3.5.2 数据预处理 22](#_Toc9877085)

[3.5.3 模型参数设置 24](#_Toc9877086)

[4 实验和知识库问答应用 25](#_Toc9877087)

[4.1 实验统计原则 25](#_Toc9877088)

[4.2 实验结果及分析 26](#_Toc9877089)

[4.3 知识库问答应用 27](#_Toc9877090)

[4.3.1 知识库问答应用设计思路 27](#_Toc9877091)

[4.3.2 知识库问答应用的实现方法 27](#_Toc9877092)

[4.3.3 知识库问答应用展示 28](#_Toc9877093)

[5 总结与展望 32](#_Toc9877094)

[5.1 论文工作总结 32](#_Toc9877095)

[5.2 展望 33](#_Toc9877096)

[参考文献 34](#_Toc9877097)

[致谢 39](#_Toc9877098)

# 绪论

## 课题的研究背景与意义

问答系统的目标都是回答自然语言问题，但根据应用场景和答案来源的不同，问答系统的形式和内容都天差地别。在应用照场景方面，有闲聊问答，如QQ聊天机器人；有任务型问答，如苹果智能语音助手Siri；常见的问题问答（FAQ），如百度AnyQ。按照答案来源划分，可分为基于知识库的问答（KBQA）、基于文档的问答（DBQA）。不同类型的问答系统，其建模方法是天差地别的。

知识库由大量实体、关系、实体组成的三元组构成，不同领域的知识库囊括了不同领域的知识。网络知识库的体量日益增长，利用知识库数据量的优势开发问答系统已是必然的发展方向。知识库，比如FreeBase和Wikidata，其结构化的知识包含了很多问题的答案，但知识库查询语句如SPARQL[1]，需要用户熟悉该查询语言的语法和知识库的实体词汇才能正确使用。相比之下，知识库问答（KBQA）系统[2]可以直接处理自然语言问题，对用户更加友好。

处理自然语言以及信息检索是知识库问答系统模型的关键。基于知识库的问答系统传统的建模方法可分为两类：语义解析（Semantic Parsing），信息检索（Information Retrieve）。基于文档的问答系统，最有名的是斯坦福基于SQuAD数据集展开的比赛，其技术依然不够成熟和完善。

## 国内外研究现状

### 基于知识库的问答

语义解析（Semantic Parsing）[3][4][5]通常通过构造一个语法解析器，将自然语言以逻辑形式转化为结构化表达，通过语法解析器，得到表达整个问题语义的逻辑形式。根据问答生成查询语句，在知识库中直接查找正确答案。

信息检索（Information Retrieve）[6][7]通常通过提取问题中的主题实体，直接查询该主题实体在知识库中的子集，作为候选答案。通过对问题特征信息提取，并建立分类器，对候选答案进行筛选，从而得出最终答案。一般来说，信息检索方法更加灵活，实现起来也更加容易。

随着深度学习（Deep Learning）在各领域崭露头角，学者们开始用深度学习的方法改善信息检索法的特征提取的工作。比如：使用卷积神经网络（CNN），并考虑了答案的类型和上下文信息[8]，使用长短期记忆网络（LSTM）[9]，使用卷积神经网络进行实体关系分类[10]，使用记忆网络（Memory Networks），注意力机制（Attention Mechanism）等等。

### 基于文档的问答

除了知识库是一个重要的答案来源之外，互联网也是答案的重要来源之一。在互联网中找答案是一经典的搜索排名的问题，其核心就是衡量问题与网页之间的语义相关性，从而找出与问题最为相关的网页。基本思路是计算网页文本和问题文本的嵌入相似程度，一般可以使用余弦相似度来表示。通过计算问题嵌入与网页文本嵌入的余弦值，经过不断的学习使得问题嵌入与网页文本嵌入的余弦值能够代表语义上的相近性，最终就可以用余弦值来找寻与问题最为相关的网页文本。

除了网页文本，还有一些答案会来自其他数据源比如百科、论坛以及图片视频等。对这类非结构化数据进行综合阅读和推理是一个非常复杂的问题，到目前为止，还没有一个非常好的效果。这一类型的任务中，比较出名的是斯坦福基于SQuAD数据集所展开的比赛。使用非结构化数据进行问答[11]的主要思路也类似于信息检索，首先检索相关文章，然后从文章中检索出具体的文字作为答案。

## 本论文的主要内容和章节安排

本论文基于信息检索模型基本原理[8]，建立知识库问答系统模型。获取问题和答案的分布式表达[12]，即词嵌入和知识库嵌入，并对嵌入之间的关系进行研究，通过多层神经网络，提取嵌入中的特征信息，从而确立答案得分，用成对排序算法[13]答案的排序情况。最终，模型根据候选答案的得分序列决定正确答案。

在问题部分，将问题嵌入通过双向长短期记忆网络提取问题单词的时序信息，确保每个单词都包含了之前单词的信息也包含之后单词的信息。在答案部分，将答案嵌入分成主体、关系、客体三部分，用注意力机制[14]对问题每个单词嵌入针对答案的三部分分别计算注意力权重，从而提取问题对于答案三部分的不同特征。同时，利用知识库中已有的全局知识信息帮助候选答案的选取和正负样本集的生成。将知识库嵌入使用TransE算法[15]预训练，使原本随机初始化的知识库嵌入处于有序状态。最后将问题和答案组合，计算得分函数，利用正负样本成对排序学习算法，对该模型的各层神经网络进行训练。

在实验中，我们使用SimpleQuestionsWikidata数据集[16]和Wikidata知识库[18]，建立了本地数据库，生成了正负样本集，并以此训练模型。完成训练后，我们用验证集对模型进行正确率统计。此外，我们还开发了知识库问答系统应用，为本模型提供使用接口。

本文章节安排：

第1章：绪论，主要介绍本论文课题的背景及其研究意义，对国内外的当前的研究情况介绍，并对本论文的主要研究内容进行概述，介绍本论文整体章节安排。

第2章：本文涉及的相关技术概述。包括了知识库、数据集的介绍，本文用到的知识库和数据集的生成方式；神经网络的基本概念和原理，Tensorflow模块的介绍，神经网络的训练方法；本文涉及的神经网络层的原理、功能和特性，其中包括了词嵌入（Word Embeddings），知识库嵌入（Knowledge Base Embeddings），双向长短期记忆网络（BiLSTM），注意力机制（Attention Mechanism）。

第3章：本论文的知识库问答系统各层的设计思路和实现方法。同时，我们使用知识库全局知识生成正负样，本并用TransE算法预训练知识库嵌入。此外我们将介绍对数据进行预处理的方法，构造得分函数与损失函数，并对模型进行训练

第4章：实验结果的展示和分析。知识库问答系统应用的设计思路，实现方法和应用展示。

第5章：对本文内容进行总结，对模型优化提出展望。

# 相关技术介绍

## 知识库和数据集

### 知识库

知识库的核心是将非结构化的自然语言知识，组织成为结构化知识，三元组是（实体，实体关系，实体）知识的组织形式，大量的三元组可以构成庞大的知识库。知识库中的知识可以来自任何地方，但往往知识的贡献者是少数该知识领域的专家。代表性的知识库有Freebase[17]，Wikidata，它们结构化的知识来自维基百科，体量庞大，有上亿条三元组。目前Google已经关闭了Freebase的访问API。

Wikidata[18]是一个免费的，协作的，多语言的辅助数据库。因为其免费，多语言等特性，是本论文实验中需要使用的公开知识库，也是目前无法访问的Freebase的替代知识库。在Wikidata中查询一个实体，将能得到实体的相关信息有：别名（aliases），声明（claims），描述（descriptions），编号（id），标签（labels），最后修订编号（lastrevid），最后修改日期（modified），页面编号（pageid），站点链接（sitelinks），标题（title），类型（type）。在Wikidata中，实体编号以字母Q为首，并跟上一串数字；谓词的作用是连接两个实体，代表了两个实体间的关系，其编号以字母P为首，跟上一串数字。

维基数据以JSON格式对知识库进行完全转储，我们将通过Wikidata提供的linked data interface API访问Wikidata知识库。其中最重要的信息是声明（claims），一个声明是指一个实体，通过一个谓词，连接另一个实体。比如Q42（Douglas Adams），P21（sex or gender），Q6581097（male），表示Douglas Adams性别是男性。三元组（Q42，P21，Q6581097）就是一条知识。

### 数据集

带标签的数据集是监督学习的关键，通过数据标签，调整神经网络的参数，使神经网络准确拟合复杂非线性函数。数据集往往为被分为三个部分：训练集，验证集，测试集。

训练集中的数据会被用来训练模型，即修改模型的参数。验证集的作用是通过验证样本，确立拟合效果最好的模型，同时也用于防止过拟合。测试集仅用作最后准确性测试。一般情况下，如果验证集可以有足够好的泛化能力，可以不需要使用测试集做最后的测试。

WebQuestions[19]和SimpleQuestions这两个数据集均是基于Freebase[17]构建的。WebQuestions中，作者主要借助了Google Suggest API，以“where was Barack Obama born?”为初始问题，通过Google Suggest API提供的建议，通过宽度优先搜索算法获取新问题。在所有生成的问题中选择一部分，交由人在Freebase中寻找答案。WebQuestions有4525个单词，5810条问答对，此外还有每个问题的主题实体，方便候选答案的选取。

WebQuestions是通过人工确定问题的答案，形成问答数据集，而本文用到数据集SimpleQuestions，是在确定三元组为答案的情况下，由人工生成的问题组成问答数据集。该数据集的每个问题都依据一个Freebase知识三元组知识，进行人工构建问题，SimpleQuestions最终一共包含了108442条问答对，对比WebQuestions，其数据量大了很多。

由于Freebase知识库的关闭，本文真正使用的数据集是在SimpleQuestions数据集的基础上，通过Freebase知识库与Wikidata知识库的对应关系，将SimpleQuestions的答案三元组映射到了Wikidata上，生成的SimpleQuestionsWikidata数据集[16]。鉴于Freebase知识库与Wikidata知识库并不能完全对应，所以最终使用的数据集是经过整理和筛选的。

## 神经网络

神经网络（Neural Networks）一定程度上模仿了人的大脑思考模型，但又大不相同。其基本单元是神经元，神经元的输入和输出模拟了大脑中突触的行为，大量神经元构成了类似于人脑的信息处理结构。权重是神经网络的关键参数之一，每个输入在神经元中都会乘以对应的权重，被视作是信息处理操作。

由于可能有多个输入信号进入一个神经元，通常，连接的输入值都会被加权求和。然后该值被传递给激活函数，激活函数是非线性函数，用于拟合非线性的目标函数。每个神经元对于输入信号的处理过程如下：

其中，表示第层第个神经元输出信号，为对应输入连接的权重，为偏置，为激活函数。对上述参数进行调整的过程，就是神经网络拟合复杂问题的核心。是线性部分，拟合了复杂问题的线性特征。而激活函数是一个非线性函数，赋予了神经元拟合非线性复杂特征的能力。常见的激励函数有Sigmoid函数、tanh函数、Relu函数。

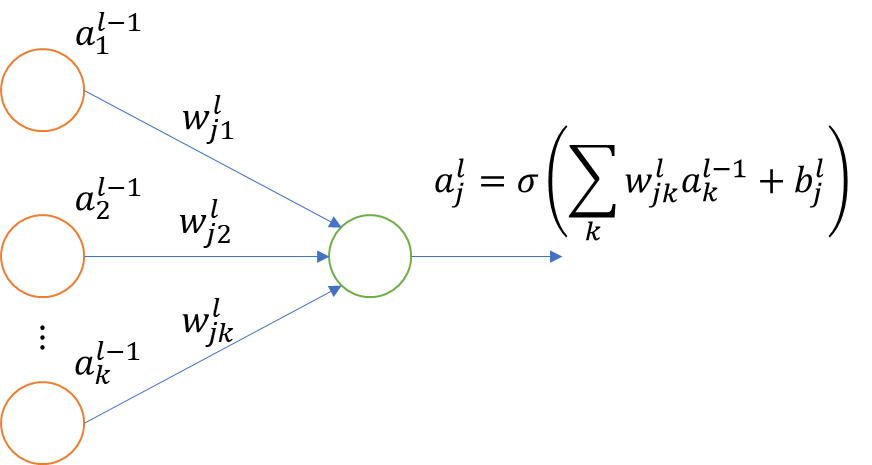


图2‑1 神经元

多个神经元组成层，多层神经元组成神经网络，神经网络的多层结构让其能拟合复杂问题。用多层神经网络学习数据特征，就是深度学习（Deep Learning），每层神经网络的计算可以转化为向量形式：

其中，表示第层的输出向量。

### 神经网络的学习

神经网络主要学习方法[21]有：监督学习，无监督学习，强化学习。此外，还有半监督学习，进化学习等等。

监督学习（Supervised Learning），是通过带标签的数据计算模型的误差，通误差在各层的传递，对神经网络模型参数进行调整。学习的过程中，输入数据经过神经网络模型计算后的输出值与标签逐渐接近，当模型输出值和标签的差距逐渐缩小到一个定值附近，且模型参数也逐渐收敛到一个定值附近，则认为该模型完成了训练。此时，可使用该模型预测未知的新数据。

监督学习的训练集要求包括输入数据和输出标签。神经网络的计算是输入数据特征提取，以及将数据特征转换为输出的过程。监督学习通过标签数据，训练得到一个最优模型，再利用该模型预测未知输入的结果，此时的模型对不存在于训练样本中的未知输入也存在一定的预测能力。

无监督学习（Unsupervised Learning）的输入数据没有标签，需要根据样本间的相似性集进行聚类（Clustering），使同类数据差距最小，异类数据差距最大。实际应用中，不少情况下的数据没有标签，所以只能使用无监督学习方法，学习出一个分类器。

强化学习（Reinforcement Learning）主要是构造一个反馈系统，对当前模型状态信息生成反馈信号，针对反馈信号，生成一系列行为改变模型的状态信息，使正反馈最大。这种机制经常被用在游戏领域，系统根据当前状态决定下一步行为，胜负则是典型的反馈信号。

在本实验中，我们将使用SimpleQuestionsWikidata作为数据集，自然语言问题作为输入，答案三元组作为标签，用监督学习调整模型参数。

### 损失函数与梯度下降

在上一小节的监督学习中，我们提到了模型输出和数据标签的接近程度，该接近程度就代表了模型的误差，也反映了模型的预测效果。通常，模型输出和数据标签的接近程度用损失函数（Loss Function）表示，损失函数直接能评价模型的拟合情况的好坏。对于输入值，标签值，和模型输出值损失函数记作。常见的损失函数有:

平方差误差：

绝对值误差：

平滑平均绝对误差Huber：

log-cosh损失函数：

为了反映整体数据集的情况整体的损失函数可以定义为：

神经网络训练的关键是寻找合适的权重和偏置，训练出符合期望的模型。这就意味着，我们要使误差尽可能小，即找到整体损失函数的全局最小值。然而损失函数的解析式无从得知的，无法通过解析式求出导数，即无法直接得到全局最小值。同时，穷举所有的权重也是不可能的任务，所以需要通过梯度下降的方式，反向传播误差，调整神经网络参数。

梯度（Gradient）即变量的偏导数组合成的向量，梯度向量就是函数上升最快的方向：

对于模型整体损失（见公式2-7），其中是模型目标函数的参数，包括了权重和偏置。用误差更新参数，使减小，即：

其中，为学习率，即每次修改参数的步长。在梯度方向上下降，是减小损失函数的最快方向，故参数的修改是根据该方向决定的。但是，只有输出层的结果可以用计算误差，只能更新最后一层神经网络的参数。我们可以借助链式求导的思想：，在计算了最后一层的误差后，将误差再次传给上一层，即：

反复迭代，修改参数，梯度不断下降，就能达到局部最小值。而学习率的选取，不仅会影响的收敛速度，过大的还会导致反复在局部最小值两侧震荡，无法收敛。

根据每次迭代的数据量不同，传统梯度下降优化算法分为全量梯度下降法、随机梯度下降法、小批量梯度下降法等，不同梯度下降法，计算效率和收敛效果不同。

全量梯度下降法（Batch Gradient Descent）每次都使用整个训练集，因此每次参数调整都能使整体的损失降低，最后能够保证收敛于极小值。但是每次都用整个训练集训练，学习时间太长，消耗大量内存。

随机梯度下降法（Stochastic Gradient Descent）每次只使用一条数据，显然学习整个训练集的迭代次数是很大的。但是每次只有一条数据，会导致模型过分关注当前数据的值，而忽略其他数据，故随机梯度下降法调整参数后可能并不会让整体损失降低，损失函数会有反复震荡。

小批量梯度下降法（Mini-Batch Gradient Descent）每次使用小批量的数据。并且如果批长度（Batch Size）选择合理，可以避免无法同时加载所有训练集的计算资源不足问题，而且损失的震荡程度也会有所减轻。

在本实验中，我们将使用小批量梯度下降法，将过大的训练集分解成诸多批次进行训练。同时对于梯度下降的过程，使用适应性矩估计（Adaptive Moment Estimation，Adam）优化算法对梯度下降算法优化。Adam算法结合了适应性梯度算法（Adaptive Gradient Algorithm，AdaGrad）[22]和均方根传播（Root Mean Square Prop，RMSProp）算法的优点，通过计算梯度的一阶矩估计和二阶矩估计而为不同的参数设计不同的学习率。

### 排序学习

排序学习（Learning to Rank）[13]起源于信息检索领域，目标是针对给定目标，对候选结果进行排序。至今排序学习依然是很多信息减速领域的核心模块，比如例如搜索引擎搜索结果排序，推荐系统候选集排序，在线广告排序等等。

通过建立一个得分函数，用得分函数对候选结果计算得分，再根据得分情况的确定排序关系是排序学习的基本思想。在基于神经网络的排序学习中，可以用神经网络模型提取输入信号特征向量，并用特征向量构造得分函数。常见排序学习方法有这三类：成点排序、成对排序、成表排序。

成点排序（Pointwise）方法是通过近似为回归问题解决排序问题。标签能直接计算结果得分，构造损失函数衡量标签得分和预测得分的差距，训练排序模型。

成对排序（Pairwise）方法是通过近似为分类问题解决排序问题。对任意两个候选结果，通过二分类器决定这两个候选结果的排序。构造损失函数衡量标签分类和候选结果分类的差距，训练排序模型，使得候选答案得分收敛于两个类别。

成表排序（Listwise）方法直接优化排序列表。标签决定了候选答案的排序列表，构造损失函数衡量标签排序和预测候选答案排序的差距，训练排序模型。由于损失函数很多具有非连续性的性质，优化困难。

本实验中使用的成对排序方法，通常的损失函数为:

其中是正样本集，是负样本集，是正负样本偏置，正负样本即为分类标签，该损失函数能保证正样本得分至少比负样本得分高出，故在最高分的范围内的答案，可以被认为是同一类，即正确答案。

### TensorFlow

TensorFlow[23]是一个采用数据流图（Data Flow Graphs），用于数值计算的开源软件库。在TenosrFlow中，首先要根据模型定义数据流图，图代表了数据的计算过程。图中输入数据为多维数组（Tensor，张量），数据经过多层神经网络的操作（operation）计算得相应的输出。然后通过会话（Session）执行定义好的数据流图，在执行的过程中可以计算损失函数，调整神经网络参数等。特殊的，每次执行的过程，并不需要计算整个数据流图，只需要计算执行结果依赖的一部分操作。

## 词的向量表示

通常图像或音频处理的是图片中所有单个原始像素点的灰度值（或者RGB值）或者音频频率（频域或时域），数据本身就是高维度的数据向量。在图像或语音识别任务中，我们所需的信息已经都存储在原始图像或音频数据，然而，自然语言系统中，我们只能通过生成词典来获取单词的编号。显然单词编号只是将词映射到了1维，每个词只包括1维信息，会导致词语义信息的流失，无法使用神经网络模型模型提取语义特征信息。比如在提取“cat”和“dog”的语义时，我们除了需要区别这两个单词外，还期望知晓他们的属性，如他们都有尾巴。

除了用字典编号表示词，还可以将词转换为独热编码（One-hot Code）表示。独特编码，将词转换为维度是词典长度的稀疏向量，只有表示该单词的维度数据为1，其余为0。显然，这种表示方法与字典编号一样，包含的信息量较少，而且在字典词较大时，需要较大的空间资源。

图2‑2 词嵌入

词的向量表示（Distributed Representations of Words）[11]，是解决字典编号丢失语义信息的方法之一。word2vec模型用于学习文字的向量表示，称之为词嵌入（Words Embedding）。

词嵌入将每个词用一个向量表示，词向量元素数据类型为浮点型（而不是独热编码的比特型），且不再是稀疏向量，词嵌入将原来不包含语义的词编号映射到了一个向量空间之中，每一个词都对应一个词向量，词向量中的每一维度表示该词的一个隐藏属性，能通过每个维度上的数字，描述词在该属性上语义。

虽然词嵌入模型的维度可以被描述为属性，但在实际研究中，词嵌入没有显式的决定这些属性的具体含义是什么（对比与图像处理中，每个图向量的维度描述了一个像素点的灰度或RGB信息），维度的数量也是在建立数据流图时，人为地设定的。

## 双向长短期记忆网络

长短期记忆网络（LSTM）[31]是循环神经网络（RNN）的一种延伸。

循环神经网络能提取输入向量的时序信息，循环神经网络神经元（或循环神经网络单元，Cell）接受两个值：一个是当前时刻的输入，另一个是前一个神经元的输出。通过将前一时刻的运算结果添加到当前时刻的运算中，从而实现了“考虑上文信息”的功能。理论上，越晚输入的信息，对最后结果的影响越大。双向神经网络（BiRNN）[25]则是在循环神经网络的基础上添加了反向运算，最终的结果为正向循环神经网络的结果与反向循环神经网络的结果拼接。这样，模型就可以实现考虑上下文信息了。

设为时刻的输入，为时刻是循环神经网络单元隐藏单元值，为时刻循环神经网络的输出。隐藏单元值不仅由此刻的输入决定，还受时刻的隐藏单元的影响。对于时刻，隐藏单元值为：

其中是激活函数，是输入权值，是隐藏权值，是隐藏偏置。对于时刻，输出为：

其中是输出权重，是输出偏置。由于循环神经网络中传递给下一个神经元的数值包含了之前所有时刻的输入信息，但是实际上RNN每个输出更依赖近期的输入信息。在考虑到某些词不同其他时刻词的依赖关系不同，所以需要长短期记忆网络（LSTM）表达输入中的长期依赖信息。

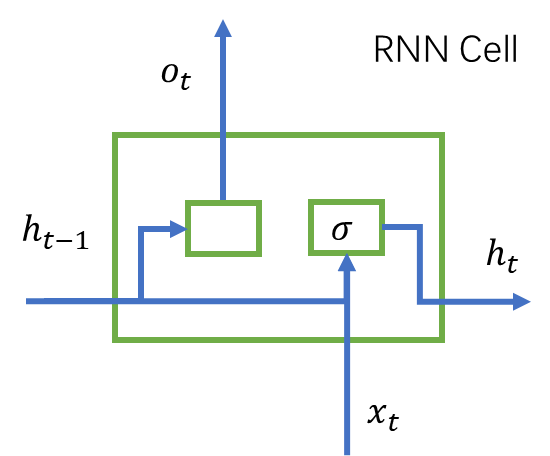
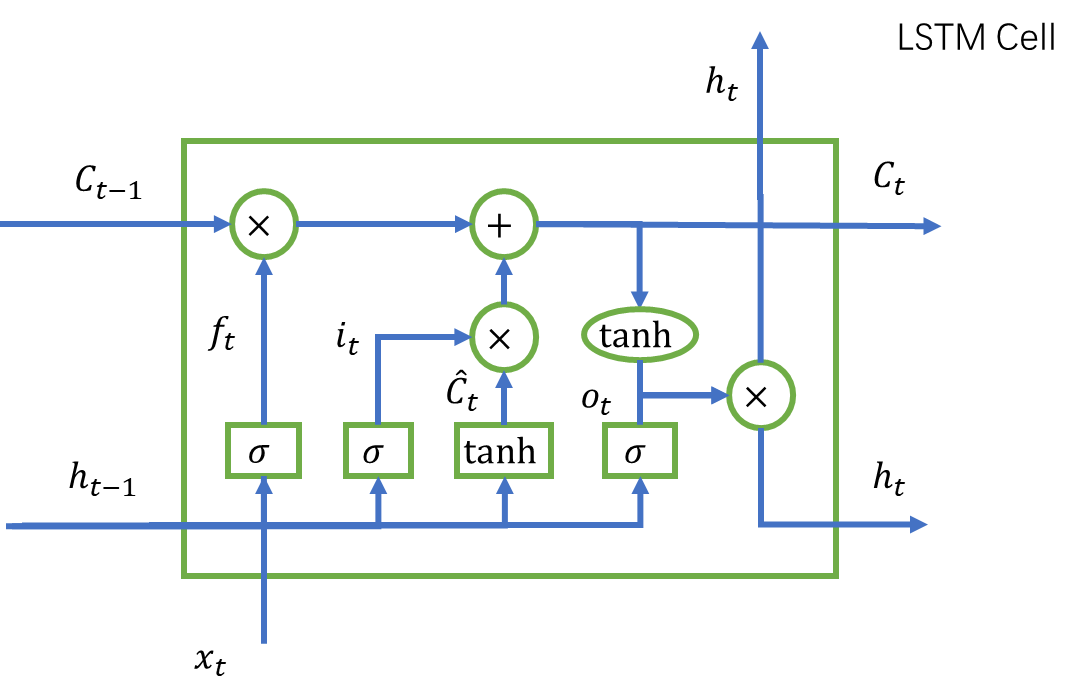
 

图2‑3 RNN神经元与LSTM神经元

在LSTM中，一共有3种门结构，分别是遗忘门（Forget Gate）、输入门（Input Gate）与输出门（Output Gate）。门函数表达式为，也即为Sigmoid激活函数，把输入值的值映射到(0,1)上。每个神经元都有三个输入与三个输出。是时刻的输入，与是短期记忆信息和长期记忆信息。

对于遗忘门：

对于输入门：

临时记忆单元（Cell）：

长期记忆信息计算：

对于输出门：

当前输出状态：

双向长短期记忆网络（BiLSTM）与双向循环记忆网络（BiRNN）同理，通过正向长短期记忆网络和反向长短期记忆网络的拼接，实现了考虑上下文的长短期信息。

## 注意力机制

深度学习中的注意力机制（Attention Mechanism）[14][24]和人类的注意力机制类似，目标是在大量信息中选择关键的信息重点关注。往往注意力机制会依附在编码-解码（Encoder-Decoder）[31]框架下的。

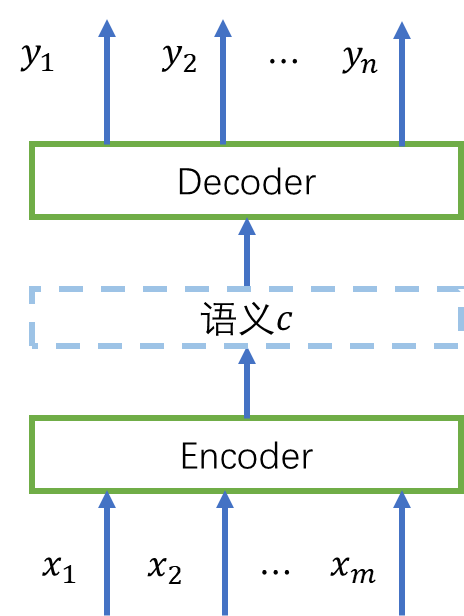
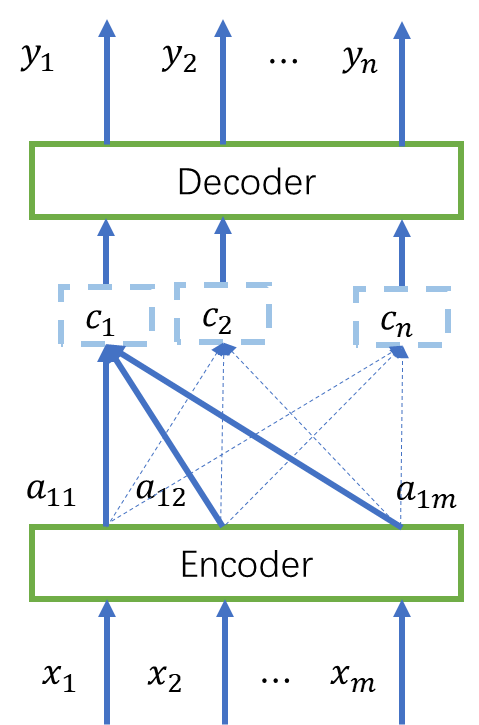
 

图2‑4 编码-解码模型和注意力机制

目标是将输入，通过编码-解码模型生成目标。

Encoder阶段：

Decoder阶段：

显然，依赖于语义，且每个依赖于之前的结果，即对时序有依赖。实际情况下，的依赖并不是平均分布在所有输入中的，故对应每个输出计算每个输入的注意力权重，通过加权求和方式计算语义：

则。对于该模型，关键就是权重的计算。通常对应的是循环神经网络的隐藏结点状态。通过注意力权重矩阵和偏置可以训练出合适的权重:

# 知识库问答系统模型

知识库问答（Knowledge Base Question Answering，KBQA）[2]即给定自然语言问题，通过对问题进行语义理解和解析，同时利用知识库进行查询，用已知信息（包括知识库信息和问题信息）推理出正确答案。设自然语言问题为，目标是给出实体集作为其答案。

首先我们需要确定问题的主题实体，SimpleQuestions的每一条问答对，其问题是有人工构造的，其答案三元组的主体就是该问题的主体实体。然后用知识库查询关于该主体实体的子集，SimpleQuestions中的答案都是一个三元组，故候选答案只需要关注主题实体的1跳范围（即查询深度为1）。由于FreeBase知识库已经无法访问，故我们使用SimpleQuestionsWikidata，将SimpleQuestions中的三元组映射到Wikidata数据库中，并通过Wikidata API，构造了一个本地知识库，包含了SimpleQuestionsWikidata中所有主题实体，以及他们在Wikidata中的1跳范围的子集。由于部分主题实体子集非常大，故我们的本地知识库对主题实体子集也进行了筛选和精简。

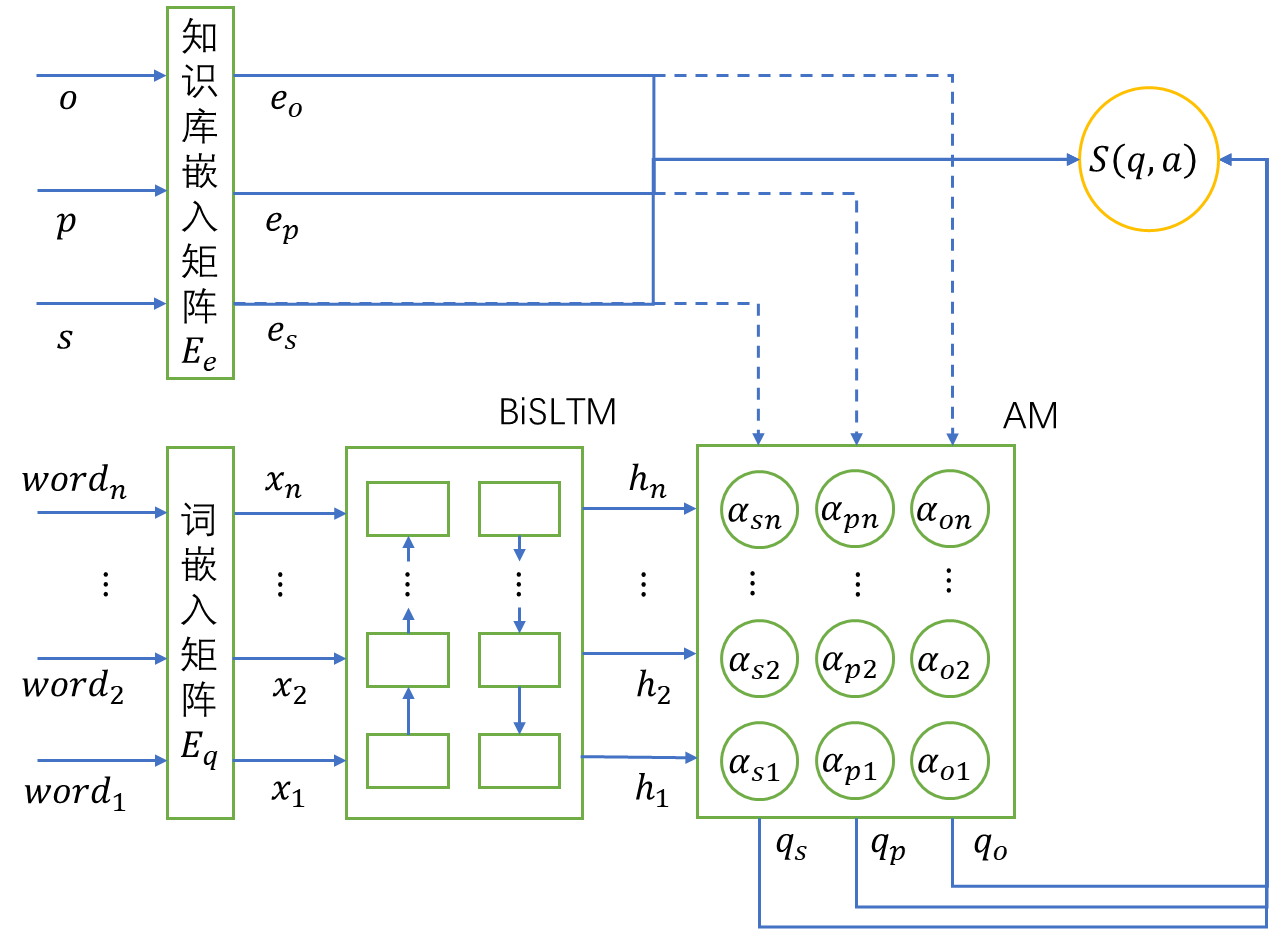


图3‑1 知识库问答系统模型

## 词嵌入与知识库嵌入

### 嵌入层设计思路

在这一层中，我们将设法将单词与知识库实体转换为词嵌入和知识库嵌入。

作为知识库问答模型的第一步，首先需要将问题和答案转换为可以被神经网络处理的多维数组，并设法提取问题和答案的语义信息。将词转换为词嵌入能带来的一下优势：

（1）词嵌入有利于数据处理。将词和实体转换为嵌入表示后，词和实体从自然语言转换为了数字，有利于神经网络的处理以及特征值的提取。

（2）词嵌入的每一维，可以对应语义的一项。词嵌入与词的字典编码和词的独热编码表示不同。字典编码只是给单词的项分配了一个数字编号，该编号不具有具体的语义信息。独热编码表示是一个维度为字典长度的稀疏向量，只有对应该单词的维度为1，其余为0。同样独热编码表示也不具有具体的语义信息，过长的稀疏向量给数据处理带来不便。逻辑上，词嵌入的每个维度，都是一个具体的语义信息。实际实验中，词嵌入的语义信息不是显示声明的，而是由神经网络学习而得，每一维度代表的语义也无法用自然语言描述，只能通过计算提取向量数字上的特征。

### 嵌入层的实现方法

自然语言问题单词序列，其中第个单词为。根据预处理的词汇字典，对应编码该序列所有单词，其中第个单词编码为。根据单词编码，生成独热编码序列，其中第个单词的独热编码后为。每个独热编码的都是长度为词汇字典长度的稀疏向量，该向量在只有在位置为1，其余为0。

根据问题的独热编码，使用词嵌入矩阵，其中表示词嵌入维度，表示词汇字典长度，计算出每个单词的词嵌入。其中每个词的独热编码为1的一列，对应词嵌入矩阵的一行，表示该词的词嵌入，即

则被转换为，其中第个单词的词嵌入为。是随机初始化的，并在训练中收敛。实际操作中，我们不需要生成稀疏向量，也不需要计算。由于中所有的0都会被忽略，只会计算值为1的列，故只需要将中第行取出，就能完成词嵌入计算。本层的核心是tensorflow.nn.embedding\_lookup模块，其本质是对嵌入矩阵进行查表运算。

同样的，使用知识库嵌入矩阵，将答案三元组转变为知识库嵌入，其中表示知识库嵌入维度，表示知识库字典长度。是随机初始化的，并在训练中收敛。

## 双向长短期记忆网络

### 双向长短期记忆网络层的设计思路

在这一层中，我们将设法提取问句的特征，即提取多个单词按一定顺序组合所表达的语义信息。

在自然语言中，每个问句是由多个单词按一定顺序组合而成。除了每个单词所包含的各自的语义，单词的组合以及顺序都会影响一个句子的语义信息。比如，问句“What is the name of a dubstep album”，其中的“dubstep”对“album”进行了限制，而“name”又被“dubstep album”限制，问句的单词之间互相影响，该特征会直接影响到答案的选取。故我们需要使用双向长短期记忆网络层在提取单词的时序信息，用双向长短期记忆网络层提取特征信息有以下优势：

（1）双向长短期记忆网络可以提取问句长期和短期的特征。不同于循环神经网络，循环神经网络的每个节点都包含了之前节点的信息，而且越靠近本节点的节点信息对本节点的影响就越大。双向长短期记忆网络的神经元中的遗忘门计算隐藏信息单元，可以保持长期记忆信息；输入门计算本节点的和上一结点的输入，代表短期记忆信息；输出门结合长期记忆信息和短期记忆信息，决定本节点的输出。

（2）双向长短期记忆网络可以提取正向和反向顺序特征的结合。不同于单向长短期记忆网络，双向长短期记忆网络不仅从左到右，正向提取输入问句的特征，还将问句从右到左，反向提取输入问句的特征。经过双向长短期记忆网络的处理，每个输出节点，既包括之前单词对本单词语义的影响，也包括之后单词对本单词语义的影响。

### 双向长短期记忆网络层的实现方法

对于问题词嵌入序列，使用双向长短期记忆网络（BiLSTM）提取问句的时序信息。将通过正向长短期记忆网络，则，再将通过反向长短期记忆网络，则。为了词嵌入的维度经过双向长短期记忆网络后，维度保持不变，对于词嵌入维度我们将长短记忆网络单元的隐藏单元设置为。每一个节点的输出为正向长短记忆网络输出与反向长短记忆网络输出的拼接，即，输出维度保持不变。至此，问题词嵌入序列被转换为了。

实际操作中，我们需要将问句按照时间顺序将单词进行拆解，即将每个词嵌入单独提取输入到长短期记忆网络单元中进行计算。对于单词数量为的问题，理论上需要个长短期记忆网络单元。虽然这些单元的权重和偏置是可以不同的，但是实际操作中，个长短期记忆网络单元共享同一组权重和偏置，即个长短期记忆网络单元的计算实际为使用1个长短期记忆网络单元，分别计算每个输入并将输出迭代给下一次计算。

本层的核心是tensorflow.nn.rnn\_cell.LSTMCell模块，LSTMCell代表了长短期记忆网络单元，在此我们需要一个正向单元和一个反向单元，该单元内部就包含了长短期记忆网络的权重和偏置，初始值都为0，并通过训练收敛。使用tensorflow.contrib.rnn.static\_bidirectional\_rnn模块，计算正向长短期记忆网络单元和反向长短期记忆网络单元，并拼接双向长短期记忆网络的输出结果。

## 注意力机制

### 注意力机制层的设计思路

在本层中，我们将设法计算每个单词对应答案不同部分的权重分布。答案的不同部分，对应每个单词，会有不同的权重。

在自然语言中，人们对一个句子中每个单词的注意程度不是平均分布的，而是有注意力集中机制。在我们使用的数据集SimpleQuestionsWikidata中，答案是一个三元组，问题是在以三元组为答案的情况下人为生成的。尽管所有的候选答案三元组都必定包括了主体实体，即其嵌入对我们计算的影响实际是一个定值。另一方面，由于Wikidata和Freebase的映射问题，原本SimpleQuestions中的答案三元组的主体总是主题实体，三元组也能在Freebase中找到，但是对应在Wikidata中三元组却是以反向关系的形势存放的，即Freebase中三元组为（主体，关系，实体），对应在Wikidata中的三元组为（客体，逆关系，主体）。在这种情况下，答案三元组的客体才是主题实体，与原数据集是不同的。

为了使主题实体在答案三元组不同部分的时候的特征不同，也为了能将问题不同单词注意力集中在不同答案部分上，本层将使用注意力机制，答案的知识库嵌入在本层中被拆分成（主体，关系，客体）三部分，每个词嵌入将分别计算对应部分的注意力权重。使用注意力机制有以下优势：

（1）对答案有直接影响的单词有较大的权重，而忽略一些不重要的单词的而影响，可以排除干扰，提高正确率。双向长短期记忆网络层的输出包括了单词本身信息，短期信息和长期信息。但是所有单词对于答案的贡献是一样的。通过注意力机制，将每个单词针对答案的三个部分拥有不同的权重，则可以排除不重要的单词的干扰，提高关键单词的影响力，有利于提高正确答案的预测效果。

（2）对于只需要一条三元组就能回答的SimpleQuestions问题，关系的正确与否是最重要的一项，所以分将答案分成三部分，有利于分析问题关键词的和答案不同部分的对应关系。比如问题“what kind of film is wildschut?”，可以确认主题词是Q5177927（wildschut），在计算答案得分的过程中“kind”担任了最重要的作用，确认了答案的关系是P136（genre）。通过主体Q5177927和关系P136可以通过知识库查找到答案Q13228657（criminality），而“kind”显然对答案的客体部分影响力不如关系部分。

### 注意力机制层的实现方法

本模型从整体上看，本模型也是一种编码-解码模型。目标通过自然语言问题，设法给出实体集。在编码部分，我们使用的是双向长短期记忆网络，将其输出与答案的知识库嵌入拼接后，参考公式（2-23, 2-24）计算对应注意力权重：

其中，为中间矩阵，为偏置。之后，将重要性权重和双向长短期记忆的输出进行加权求和：

其中，代表了问题在方面的特征。特别的，与公式（2-22）不同的是，本模型在此增加了激活函数。实验中，注意力机制的中间矩阵，和偏置是随机初始化的，并在训练中收敛。而且为了过滤单词不重要特征，我们在这一层中的输出层，补了一个Relu激活函数，目的是为了弥补在双向长短期记忆网络层的输出层没有使用激活函数。

## 全局知识信息

知识库全局知识信息（Global Knowledge Information）[13][33]能帮助正负样本的构造。理论上，对于有正确答案标签的问答对，除了正确答案外，所有的实体对都是潜在的负样本。通过知识库全局知识信息，我们在知识库中查找相关主题实体的子集后，负样本就能在该子集范围内生成。这样生成的负样本更加有针对性，排除了不存在的三元组以及和问题毫无关系的三元组。

除了提取问题和答案之间的对应特征来建模进行问答，知识库本身也隐含着答案间的联系。通过知识库的全局知识信息，预训练知识库嵌入矩阵，使随机初始化的知识库嵌处于有序状态，有利于答案特征的提取。

设表示所有实体的集合，表示域中所有关系（谓词）的集合。二元关系是实体对的子集，更高元的关系的定义与此类似。关系的特征函数表示为每个可能的实体对是否是的一部分。我们将（可能的）关系实例表示为，其中表示不对称关系的第一和第二变量。称为主体和客体，称为三元组。学习知识库全局知识信息，首先要将三元组组合后映射到向量空间中:

其中是关系和实体的嵌入，是Sigmoid激活函数；表示所有知识库嵌入的集合；表示组合算子，它从嵌入中为创建组合(Composite)向量表达。

在知识库嵌入模型中，组合算子就是三元组嵌入组合后映射到向量空间的计算方法。为了能使可能正确的答案得分尽可能高，我们使有关系知识库嵌入，即的实体嵌入之间形成特殊的数学关系。本实验中，我们尝试使用TransE模型，用知识库嵌入间的距离作为三元组组合向量表达。参考公式（2-11），损失函数选为：

其中，表示向量间的距离，表示正样本集，表示负样本集，是正负样本偏置。这里的负样本，可以是的所有实体对。

## 训练

### 得分函数和损失函数

得分函数是知识库问答模型的关键，得分的高低决定了答案的选取。结合前几层神经网络的输出，得分函数的定义如下

为了让正确答案的得分比错误答案的得分高，我们使用成对排序损失（pairwise ranking loss）衡量模型的预测效果，参考公式（2-11），损失函数定义如下：

其中，为正负样本的偏置，表示正样本集，表示负样本集。训练的算法采用Adam优化算法，每次的学习率根据梯度矩估计调整。

### 数据预处理

接下来我们需要为训练构建合适的样本集。

实验使用的数据集是SimpleQuestionsWikidata，该数据集是由SimpleQuestions数据集整理而来。SimpleQuestions是一个包含简单问答对的数据集，每个问题的答案对应一个三元组，格式为（主体，关系，客体），知识来自于Freebase知识库。70%作为训练集，10%作为验证集，剩余的20%作为测试集。由于Freebase已经关闭，SimpleQuestionsWikidata将SimpleQuestions的答案三元组映射到到Wikidata知识库中的三元组上。只有部分实体能在Wikidata中找到对应实体，而且，即使能在Wikidata中找到对应的实体，Wikidata中也不一定存在SimpleQuestions中的答案三元组。特别的，SimpleQuestionsWikidata的答案三元组中的关系若以R开头，这说明该关系在Wikidata中是逆关系，即Wikidata中存在（客体，逆关系，主体）三元组。

首先处理自然语言问题部分。读取问题后对每条问题做以下处理：

（1）去除问句尾的“?”和“.”，保留句子内部的“?”和“.”。句子内部的“?”和“.”有可能是问句中的特殊名词（如书名，电影名）的一部分。

（2）将问句首的"what's"和"who's "改写为"what is"和"who is"。疑问词是问答的关键词，在有缩写的情况下"what's"会被认为是单独的一个单词，改写疑问词有利于数据集中疑问词的统一，句中的"what's"和"who's "不再有疑问词的特征，而是作为特殊名词的一部分。

（3）将问句中的特殊符号，如“(”，“:”，“,”等特殊符号在两侧添加空格。英语的这些特殊符号是紧跟在单词后的，在分词时会被视为单词内部字符。为了单词的统一，以及提取特殊符号的语义，在特殊符号两侧添加空格，便于分词。

（4）将问句中所有的“’s”前添加空格。英语的“’s”虽然是一种缩写，但是其有自身的语义，将其与单词分离，有利于提取“’s”自身的语义。

（5）手工修改一些数据集中的干扰符号，比如单独出现的“[”等。

（6）对于句子中出现“http:”或“https:”的句子，不做“/”和“?”的处理。“http”和“https”有自身的语义，通过对“:”的处理以及能够分解关键词，之后url的部分可以忽略。

（7）将所有问句按空格划分为单词，并生成单词字典，即每个不同的单词都将有不同的数字编号。虽然可以确定，目前所有的单词都肯定会出现在字典中，但是为了能处理未知数据输入，我们在词典首部插入“UNKNOWN”作为未知单词。

对答案三元组（主体，关系，客体）做以下处理：

（1）通过Wikidata提供的linked data interface API访问Wikidata知识库，包括答案三元组在内，将知识库中以主体实体和中心一跳范围内的子集加载到内存，并将三元组保存到本地知识库。由于SimpleQuestions和Wikidata体量较大，而且通过网络访问Wikidata较慢，实验中并未把全部相关三元组都保存到本地，当包含同一条关系的三元组数量超过三个，我们就停止对该关系的继续加载，即同一条关系下，至多有三个三元组。

（2）本地知识库中，所有的实体（包括三元组中的主体，关系和客体）生成知识库字典，即每个实体都有不同的数字编号。并在知识库字典首部插入“UNKNOWN”作为未知实体。

SimpleQuestionsWikidata中的问答对是正样本集，经过知识库全局知识信息处理后，我们在知识库中查询主题实体的子集，除去正确三元组后，其余三元组作为负样本集。对于样本集中的问答对做以下处理：

（1）在问答对的问题部分，将问题划分为单词序列，通过单词字典转码成问题序列，问题序列长度为20，并忽略长度超过20的问句。序列中的每项对应该时序的单词字典编码。

（2）对于长度不足20的问句，将问句序列用-1补齐到20。-1在之后的嵌入层中，会返回全0的嵌入向量，降低对实际语义产生影响的同时，把问题向量长度固定在20，方便神经网络的批量数据处理。

（3）在问答对的答案部分，将答案三元组（主体，关系，客体），通过知识库字典转码成答案序列，序列长度为3。序列中每项对应三元组中项。虽然所有的候选答案都会包括主体实体，但是主题实体在答案中的部分（主体和客体）不同，其代表的语义就不同（主题实体不一定是答案三元组的主体的原因在2.1.2章节有讲述），所以我们还是保留了主题实体在序列之中。

（4）针对每个问题，通过问题的主题实体，在本地知识库中找出所有候选答案（包含主题实体的三元组）。将候选答案中除了正确答案的三元组作为错误答案放到负样本集中，负样本集的生成利用了知识库全局知识信息。显然对于每个问题，负样本的数量是不确定的，正负样本集在数量上也有很大差距，负样本的数量比正样本多很多。一般为了结果样本不均衡会采取过采用（Oversampling）和欠采样（Undersampling）的方法。在本实验中我们使用过采样的方式，即对于每个负样本，对应问题的正样本都会被计算得分，并计算成对排序损失。

### 模型参数设置

整体训练轮数（Epoch）是5，每批数据大小（Batch Size）为128。本模型的词嵌入矩阵和知识库嵌入矩阵，维度都设为=128，且嵌入矩阵是随机初始化的。成对损失函数的正负偏置，预训练损失正负偏置。问题向量长度限制为20，忽略问题长度超过20的问题，并用-1补齐问题向量。双向长短期记忆网络层中，正向和反向长短期记忆网络神经元的隐藏单元都设置为=64，长短期记忆网络神经元的权重和偏置都是默认初始化的，即为全0。注意力机制层中，中间矩阵是256行1列的矩阵，是随机初始化的，偏置初始值设为0.1。

# 实验和知识库问答应用

## 实验统计原则

由于SimpleQuestionsWikidata数据集的很多的问题是可以有多条答案的，比如“what location is in the north american central time zone”，显然三元组满足（Q\*，P421（时区是），Q2086913（北美中央时区））都是正确答案。但是数据集中，正确答案只标记了一条，这导致统计时，有可能得分最高的候选答案并不是标记的正确答案。针对上述的情况，制定如下统计原则：

（1）候选答案数量为1的问题不计入正确率统计。

（2）对于问题，候选答案集合,计算所有候选答案得分。

（3）对最高得分的答案，其范围内的答案，都视为正确答案集的元素，是上节中的正负样本偏置，见公式（3-8）。

（4）实验中，由于很多问题的候选答案，几乎都在都在中。为了避免统计时过大导致无法用准确率描述模型好坏，我们只选择中得分最高的至多5个元素组成最终答案集。即，，且，有。

（5）若最终答案集中包括了标签答案，则认为该问题被完全正确回答；包括标签答案的关系部分，则认为问题被部分正确回答。我们认为关系是判断答案正确与否的重要一项。记问题的总数为，被完全正确回答的问题数量为，被部分正确回答的问题数量为，则完全正确率和部分正确率：

根据上述原则，我们统计了知识库问答模型的准确率。

## 实验结果及分析

表4‑1 知识库问答模型准确率统计

|  |  |  |
| --- | --- | --- |
| 模型 | （%） | （%） |
| IR | 51.9 | 90.9 |
| IR+GKI | 56.3 | 95.9 |
| IR+BiLSTM+AM | 61.0 | 92.5 |
| IR+BiLSTM+AM+GKI | 62.7 | 94.5 |

表4-1中，IR表示基础的信息检索模型[8]，只有嵌入层。BiLSTM表示双向长短期记忆网络层，AM表示注意力机制层，GKI表示使用全局知识信息对知识库嵌入矩阵进行预训练。

实际实验中，GKI是在模型训练一轮后，再使用TransE预训练知识库嵌入矩阵。这是考虑到初始时随机初始化的知识库嵌入矩阵还没有有价值的语义信息，经过一轮训练后已经包含了一些语义特征，此时使用GKI可以将语义特征扩展到全知识库。

在上小节中，我们提到实际正确答案数量有可能远大于标记正确答案数量（1条）的情况，所以对于很多问题来说，候选答案之间的确难以区分。而且标签答案只包括一条三元组，对于一些实际需要信息超过一条三元组的信息的问题，模型是很难做出区分。比如，“what male actor was born in warsaw”，该问题除了关键的P19（出生地），Q270（warsaw）外，其实“male”也是确认答案的依据，不过实际答案标签中，没有“male”的相关信息，只能期望由答案嵌入的隐藏语义信息提取相关信息。针对这一类，无法靠标签提取额外信息的问题，往往无法完全正确回答，这也是我们设置部分正确率的原因。

全局知识信息将知识库中有关联的知识库嵌入收敛到了一起，故理论上其影响会体现在部分准确率上。实验中，预训练对预测的效果有一定的提升，但并不明显，可能的原因是本地数据库信息量较小，只包括与数据集有关的三元组，训练过程产生了过拟合，预训练算法无法发挥其优势。也有可能是标签的答案只有唯一一条，即使将其他语义接近的三元组选为答案，也不会提升准确率。

## 知识库问答应用

在本节中，我们将设计图形用户界面，为知识库问答系统开发应用。该应用的基本框架是PyQt5，是Qt的python模块。

### 知识库问答应用设计思路

知识库问答应用的开发原则是提供友好的用户界面，提供知识库问答接口。在界面中，除了有必要的标签信息外，需要有输入问题的搜索框，展示答案的文本区域。此外还需要对界面用Qt样式表（Qt Style Sheets）进行一定的美化。

由于知识库问答的特殊性，主题是知识库问答的关键因素，只有确定了主题才能在知识库中查询候选答案。关于如何在问句中确立主题实体也是一个复杂的问题，除了需要对问题进行分词和词性标注，还要进行消歧，这部分在语义解析模型中会有更深入的研究。

在本应用中，为了避免确立主题带来的诸多不便，我们人为地指定主题词。而且本模型是基于SimpleQuestionsWikidata数据集和Wikidata知识库的，对于完全没有学习过的问题和不存在于本地知识库的主题显然无法正确回答。故在应用限制主题的选择避免了一些难以处理的问题。关于推荐主题我们使用了SimpleQuestionsWikidata中测试集的数据，生成推荐主题。

本地知识库中只存储了三元组，关于实体的具体信息则需要向Wikidata查询。为了让用户能了解主题词和答案三元组的相关信息，应用会实时查询Wikidata，并在展示区域能显示相关实体信息。

### 知识库问答应用的实现方法

界面是基于PyQt开发的，所以我们需要关注Qt的核心模块：信号与槽。用户在界面中操作，由被操作的部件发出信号，槽接收信号后经过处理信号并将结果返回到界面中的部件，是Qt人机交互的基本处理流程。

在左侧，有三个按钮部件，处理窗口的关闭和最小化等操作。在左侧的列表视图部件中，展示了可选的主题实体。这些主题实体来自于SimpleQuestionsWikidata的测试集，而且已经将主题相关信息保存在本地文件中。初始化时，应用会读取本地文件，加载主题词列表，并显示在左侧列表视图部件中。左侧列表视图部件的点击信号连接了数据读取操作，每次点击都会读取内存中的主题词列表并显示在右侧文本展示框部件中，而且还会将当前问题的主题词设置为该主题实体。

在右侧，行文本编辑框部件是问题输入接口，用户将在此输入有关主题的问题。在该部件的回车信号连接了知识库问答操作，首先会读取输入问题文本，对文本预处理后，根据当前主题，在知识库中寻找候选答案。将问题与候选答案使用后台的知识库问答系统模型进行处理，计算出答案得分后，选择得分最高的5个答案三元组显示在结果列表视图部件中。为了让用户获取答案三元组的详细信息，在右侧结果列表视图部件中，点击信号连接了Wikidata查询操作。点击右侧结果列表视图会将结果读取，并连接Wikidata获取答案三元组的相关信息，该操作是在线查询，故需要一定的等待时间。

### 知识库问答应用展示



图4‑1 知识库问答系统应用展示

通过点击左侧列表视图部件，可以在右侧查看主题词的详细说明。

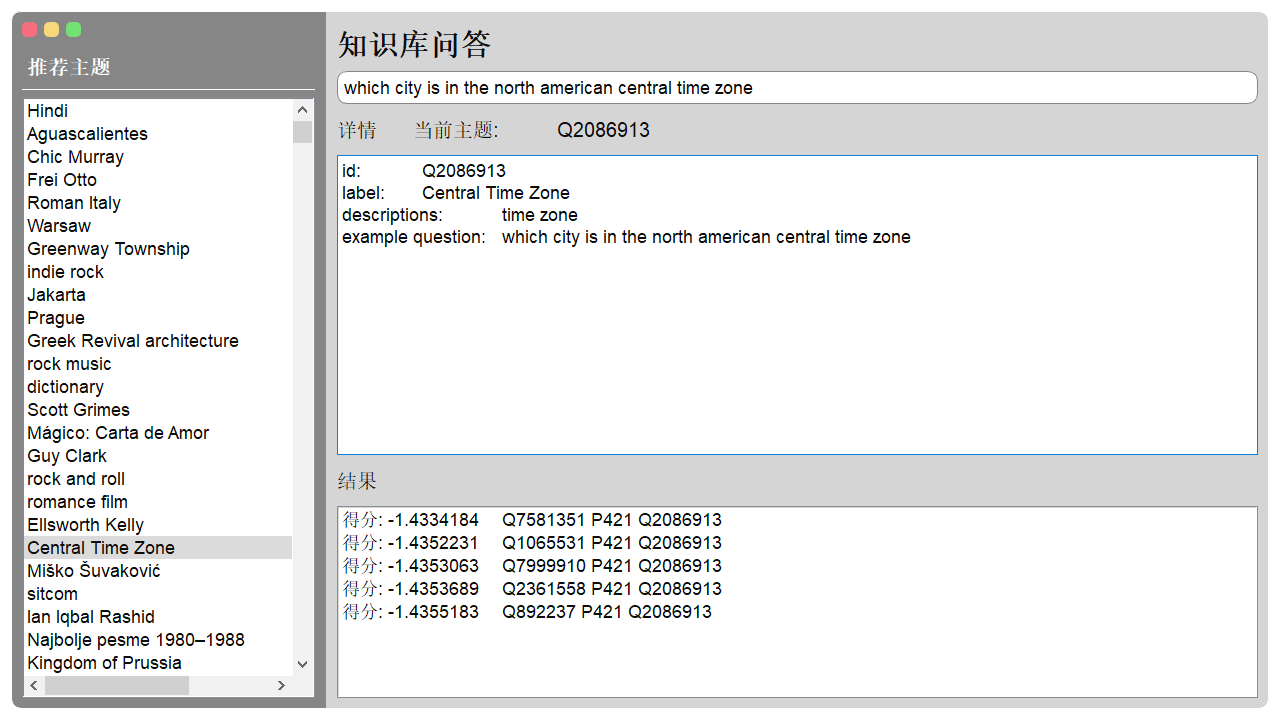


图4‑2 知识库问答系统应用展示

通过对输入问题的处理，将问题答案展示在右侧结果列表视图部件中。通过点击右侧结果列表视图部件，实时查询Wikidata知识库，获取答案三元组的相关信息。

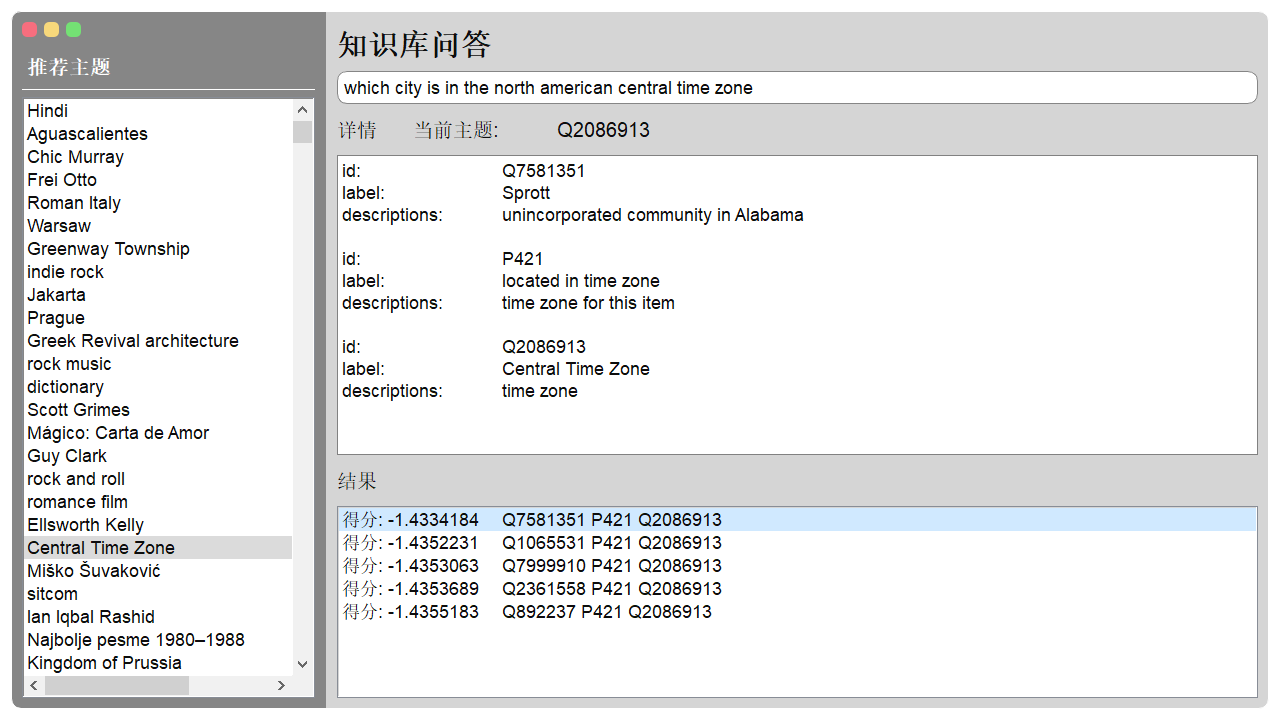


图4‑3 知识库问答系统应用展示

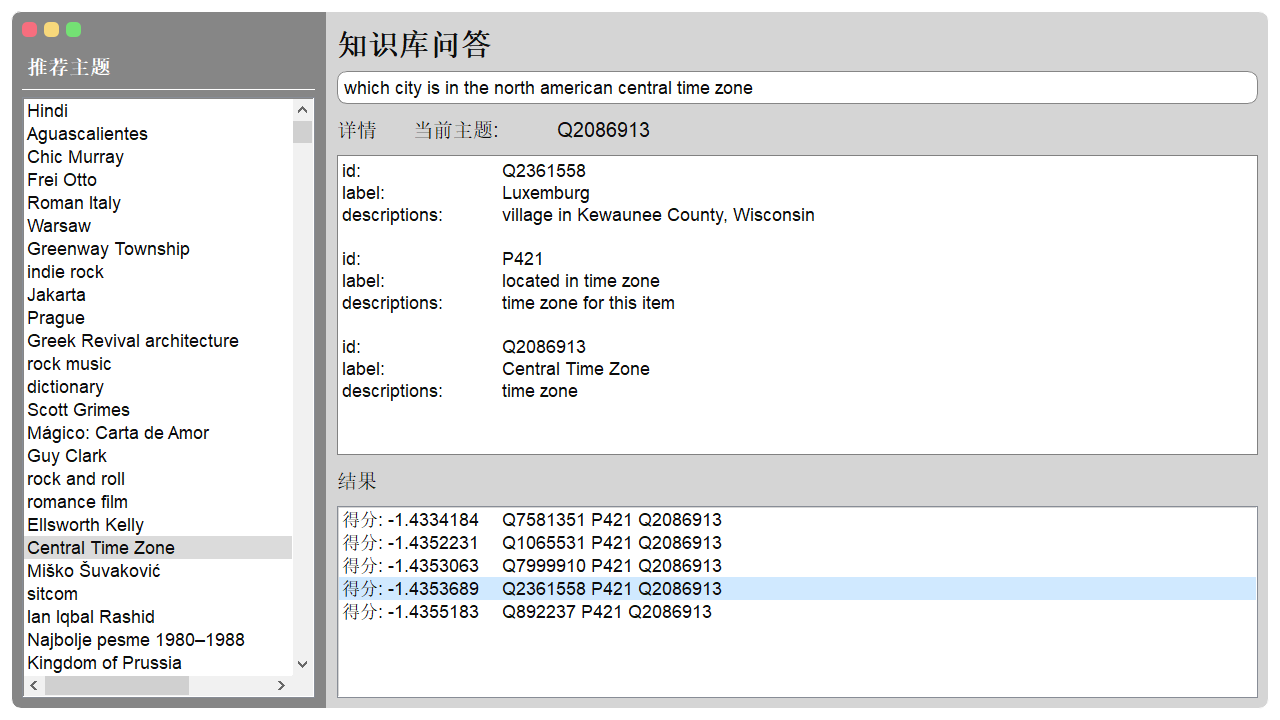


图 4‑4 知识库问答系统应用展示

答案得分直接代表答案正确程度，对于部分问题，答案得分的差距较大的情况下，有可能会包含错误答案。

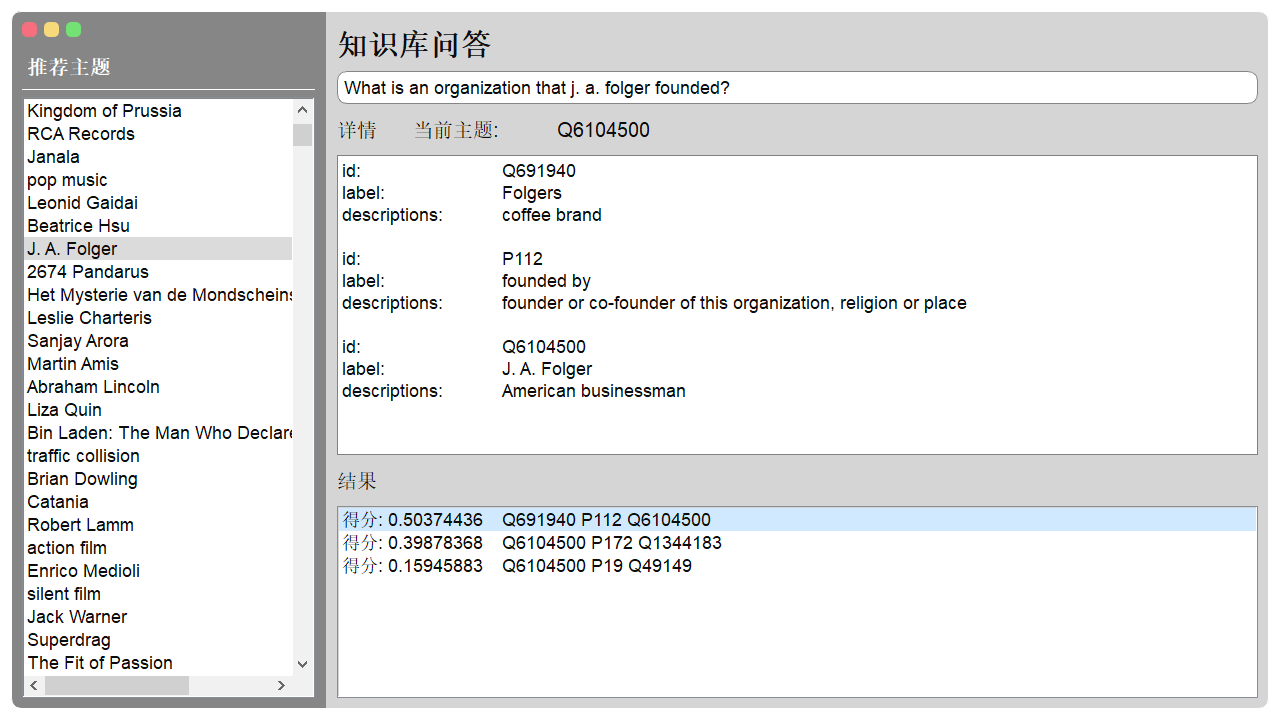


图 4‑5 知识库问答系统应用展示

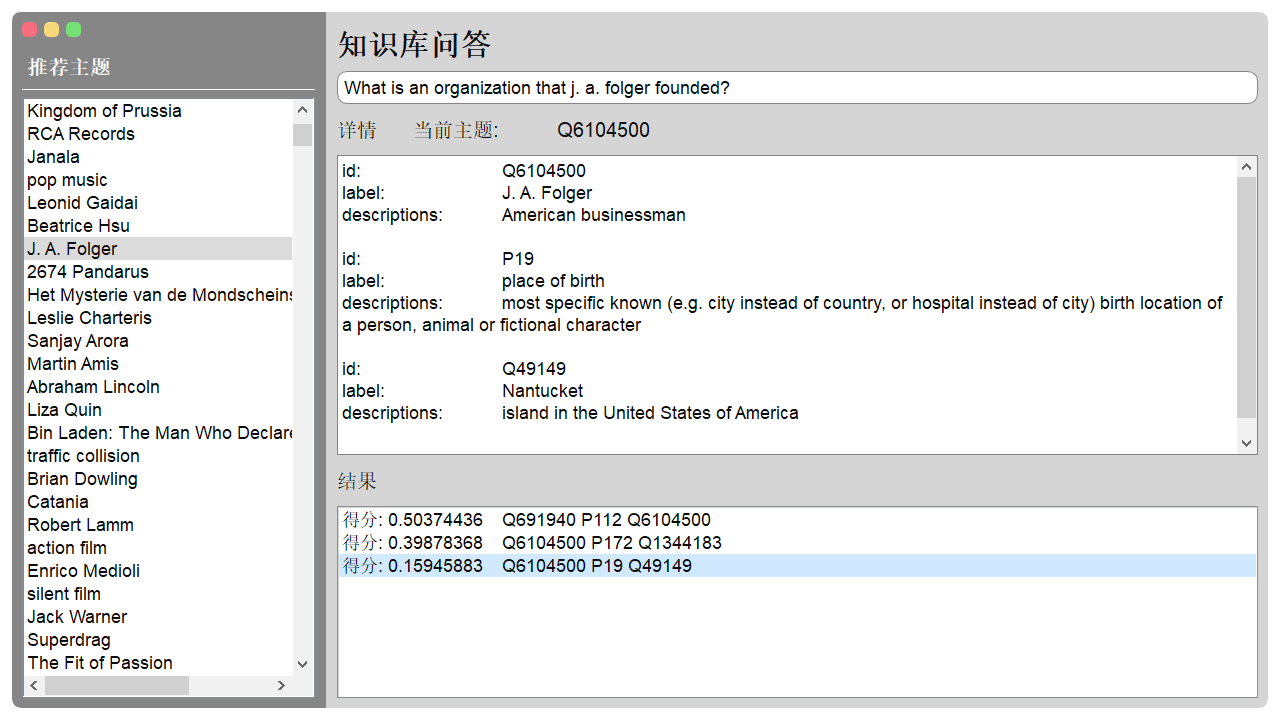


图 4‑6 知识库问答系统应用展示

# 总结与展望

## 论文工作总结

本论文基于信息检索模型基本原理，建立知识库问答系统模型。获取问题和答案的分布式表达，即词嵌入和知识库嵌入，并对嵌入之间的关系进行研究，通过多层神经网络，提取嵌入中的特征信息，从而确立答案得分，并用成对排序学习算法学习候选答案的排序情况，在候选答案中选择排序靠前的答案作为正确答案。

在问题部分，将问题嵌入通过双向长短期记忆网络提取问题单词的时序信息，确保每个单词都包含了之前单词的信息也包含之后单词的信息。在答案部分，将答案分成主体、关系、客体三部分，用注意力机制对问题每个单词的嵌入针对答案的三部分计算注意力权重，从而提取问题对于答案三部分的不同特征。同时，利用知识库中已有的全局知识信息帮助候选答案的选取和正负样本集的生成。将知识库嵌入使用TransE算法预训练，使原本随机初始化的知识库嵌入保持有序。最后将问题和答案组合，计算得分函数，利用正负样本成对排序学习算法，对该模型的各层神经网络进行训练。此外，我们开发了一个知识库问答应用，为知识库问答模型的使用提供了接口。

本论文的创新之处在于：

（1）使用全局知识信息有针对性地生成正负样本集。理论上，负样本可以是除答案外的所有三元组，但通过知识库已有的知识，将负样本确定在主题实体一跳范围的子集内，使训练更有针对性。

（2）将答案分成了主体，关系，客体三个部分，让问题针对三部分具有不同的注意力权重。注意力机制使得即使是同样的实体，在代表答案的不同部分时，依然能有不同的特征。避免候选答案总会包括主题实体嵌入，而体现不出差异的问题，区别了主题实体在答案中处于主体和处于客体的不同情况。

（3）对出双向长短期记忆网络层的激活函数，放在在了注意力机制层之中。在注意力机制层使用双向长短期记忆网络输出的全部信息计算注意力权重，在计算注意力权重后，再用激活函数过滤对答案选择无用的信息。

（4）研究了使用全局知识信息对知识库嵌入矩阵预训练后，模型的预测效果。

## 展望

知识库全局知识信息还有很多，比如实体类型，三元组时效性（比如美国的首都信息，在不同历史时间，首都是变化的）等，它们也包含了答案的一部分信息，如何考虑更多的信息帮助答案特征提取是进一步研究的方向。

在自然语言处理方面，本实验和应用都没有解决提取主题实体的问题，而是人为规定主题实体。实际上的知识库问答应用，提取主题实体（和问题中其他实体）是必要的一步。该部分的思想可以很大程度上参考语义解析模型，需要对问题分词，标注词性，语义消歧。同样的，处理该问题也可以使用神经网络框架进行建模。针对复杂的问题，自然语言处理难度极大，到目前也没有很好的成果。

在知识库嵌入方面，知识库嵌入组合向量表达包含的是三元组的组合信息，其真正的应用是在知识库预测方面。通过组合向量表达计算三元组的得分，可以判断三元组是否合理，也可以用于三元组分类。而组合向量表达其实有很多不同计算方法，即很多组合算子，比如RESCAL模型[36]，ER-MLP模型[37]，HOLE模型。若能有效利用组合向量表达，则可以在知识库方便帮助知识库问答。对知识库嵌入矩阵的预训练，是对知识库全局知识信息的学习过程，本实验中，对全局知识的利用该是非常有限的的，效果也不明显。或许对于处理复杂问题和未知问题的情况下，答案实体间的联系能发挥更大的作用。我们希望在以后的研究学习中，进一步加深对知识库知识的利用。

知识库问答模型的优化方案还有很多，在防止过拟合方面，还能使用标准化和Dropout等等值的尝试的方法。标准化让词嵌入和知识库嵌入的值保持在0到1之间，即其代表的语义信息将更加明确。Dropout可以忽略无用属性的影响，在注意力机制层更需要Dropout提取关键信息，忽略无关影响。

# 参考文献

1. Eric Prud’Hommeaux, Andy Seaborne. Sparql query language for rdf[R]. W3C recommendation. 2008.
2. Christina Unger, André Freitas, Philipp Cimiano. An Introduction to Question Answering over Linked Data[C]. Reasoning Web, Reasoning on the Web in the Big Data Era. Lecture Notes in Computer Science. 2014, 8714: 100-140.
3. Luke S Zettlemoyer, Michael Collins. Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars[C]. Conference on Uncertainty in Artificial Intelligence. UAI. 2005: 658–666.
4. Luke S Zettlemoyer, Michael Collins. Learning Context-Dependent Mappings from Sentences to Logical Form[C]. International Joint Conference on ACL. DBLP, 2009: 976–984.
5. Cai Qingqing, Yates Alexander. Large-scale semantic parsing via schema matching and lexicon extension[C]. The Association for Computer Linguistics. ACL(1). 2013: 423-433.
6. Xuchen Yao, Benjamin Van Durme. Information extraction over structured data: Question answering with freebase[C]. The Association for Computer Linguistics. ACL. 2014: 956-966.
7. Antoine Bordes, Jason Weston, Nicolas Usunier. Open question answering with weakly supervised embedding models[C]. ECML/PKDD, Lecture Notes in Computer Science. Springer. 2014, 8724(1): 165-180.
8. Li Dong, Furu Wei, Ming Zhou, Ke Xu. Question answering over freebase with multi-column convolutional neural networks[C]. The Association for Computer Linguistics. ACL(1). 2015: 260-269.
9. Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, Zhi Jin. Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Paths[C]. The Association for Computational Linguistics. EMNLP. 2015: 1785-1794.
10. Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao. Relation Classification via Convolutional Deep Neural Network[C]. COLING. 2014: 2335-2344.
11. Pum Mo Ryu, Myung Gil Jang, Hyun Ki Kim. Open domain question answering using Wikipedia-based knowledge model[J]. Information Processing and Management. 2014: 50(5), 683–692.
12. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality[J]. Advances in Neural Information Processing Systems, 2013, 26: 3111-3119.
13. Tie-Yan Liu. Learning to Rank for Information Retrieval[M]. Springer Berlin Heidelberg, 2011: 49-70.
14. 余本功, 许庆堂. 基于协同注意力机制的答案选择算法研究[C]. 第十三届（2018）中国管理学年会. 2018: 7.
15. Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, Oksana Yakhnenko. Translating Embeddings for Modeling Multi-relational Data[C]. NIPS. 2013: 2787-2795.
16. Dennis Diefenbach, Thomas Tanon, Kamal Singh, Pierre Maret. Question Answering Benchmarks for Wikidata[C]. CEUR Workshop Proceedings. CEUR-WS. 2017, 1963.
17. Kurt Bollacker, Robert Cook, Patrick Tufts. Freebase: A Shared Database of Structured General Human Knowledge[C]. the Twenty-Second AAAI Conference on Artificial Intelligence. AAAI. 2007: 1962-1963.
18. Vrandečić, Denny, Krötzsch, Markus. Wikidata: a free collaborative knowledgebase[J]. Communications of the Acm, 2014, 57(10):78-85.
19. Jonathan Berant, Andrew Chou, Roy Frostig, Percy Liang. Semantic parsing on freebase from question-answer pairs[C]. EMNLP. 2013: 1533-1544.
20. Herve Bourlard, C.J. Wellekens. Links between Markov models and multilayer perceptrons[J]. Pattern Analysis & Machine Intelligence IEEE Transactions on, 1990, 12(12):1167-1178.
21. Juergen Schmidhuber. Deep Learning in Neural Networks: An Overview[J]. Neural Networks. 2015, 61: 85-117.
22. John Duchi, Elad Hazan, Yoram Singer. Adaptive Subgradient Methods for Online Learningand Stochastic Optimization[J]. Journal of Machine Learning Research. 2011, 12:2121–2159.
23. Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro,Greg Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow,Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur,Josh Levenberg, Dan Man, Rajat Monga, Sherry Moore, Derek Murray, Jon Shlens, BenoitSteiner, Ilya Sutskever, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Oriol Vinyals, PeteWarden, Martin Wicke, Yuan Yu, Xiaoqiang Zheng. TensorFlow: Large-Scale MachineLearning on Heterogeneous Distributed Systems[C]. 12th USENIX conference on Operating Systems Design and Implementation. 2016: 272-283.
24. Fabiola Mancini, Andrea Savarino, Maria Losardo, Antonio Cassone, Alessandra Ciervo. Neural Machine Translation by Jointly Learning to Align and Translate[J]. Microbes and Infection. 2014, 11(3): 367–73.
25. Mike Schuster, Kuldip Paliwal. Bidirectional Recurrent Neural Networks[J]. IEEE Transactions on Signal Processing, IEEE, 1997, 45(11): 2673--2681.
26. Wen tau Yih, Ming-Wei Chang, Xiaodong He, Jianfeng Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base[C]. The Association for Computer Linguistics. ACL(1). 2015: 1321-1331.
27. 陆汝钤.自然语言理解和机器智能[J].语言战略研究,2018,3(03):7-8.
28. Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, Luke Zettlemoyer. Scaling semantic parsers with on-the-fly ontology matching[C]. EMNLP. 2013: 1545-1556.
29. Daojian Zeng, Kang Liu, Yubo Chen, Jun Zhao. Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks[C]. The Association for Computational Linguistics. EMNLP. 2015: 1753-1762.
30. Minwei Feng, Bing Xiang, Michael R. Glass, Lidan Wang, Bowen Zhou. Applying Deep Learning to Answer Selection: A Study and An Open Task[J]. 2015 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU. 2015: 813-820.
31. Sepp Hochreiter, Jürgen Schmidhuber. Long Short-Term Memory[J]. Neural Computation. 1997. 9(8): 1735-1780.
32. Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, Zachary Ives. DBpedia: A Nucleus for a Web of Open Data[C]. The Semantic Web. Lecture Notes in Computer Science. Springer. 2007, 4825: 722-735.
33. Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge. Freebase: a collaboratively created graph database for structuring human knowledge[C] 2008 ACM SIGMOD international conference on Management of data. 2008: 1247-1250.
34. Richard Socher, Brody Huval, Christopher D. Manning, Andrew Y. Ng. Semantic Compositionality through Recursive Matrix-Vector Spaces[C]. EMNLP-CoNLL, ACL. 2012: 1201-1211.
35. Maximilian Nickel, Kevin Murphy, Volker Tresp, Evgeniy Gabrilovich. A Review of Relational Machine Learning for Knowledge Graphs[J]. IEEE. 2016, 104(1): 11-33.
36. Maximilian Nickel, Volker Tresp, Hans-Peter Kriegel. A Three-Way Model for Collective Learning on Multi-Relational Data[C]. International Conference on International Conference on Machine Learning. ICML. Omnipress. 2011: 809-816.
37. Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, Wei Zhang, Knowledge vault: A web-scale approach to probabilistic knowledge fusion[C]. the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. 2014: 601-610.
38. Zhen Wang, Jianwen Zhang, Jianlin Feng, Zheng Chen. Knowledge Graph Embedding by Translating on Hyperplanes[C]. Twenty-eighth AAAI Conference on Artificial Intelligence. AAAI. 2014: 1112-1119.
39. Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, Bo Xu. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification[C]. ACL. 2016: 207–212.
40. Yexiang Xue, Yang Yuan, Zhitian Xu, Ashish Sabharwal. Expanding Holographic Embeddings for Knowledge Completion[C]. Nips. 2018: 4496–4506.
41. Richard Socher, Danqi Chen, Christopher D. Manning, Andrew Y. Ng. Reasoning with Neural Tensor Networks for Knowledge Base Completion[C]. NIPS 2013, 13: 1-10.
42. John Prager. Open-Domain Question–Answering[J]. Foundations and Trends in Information Retrieval. 2006, 1(2): 91–231.

# 致谢

本课题是对四年学习的总结，从最初开始，本课题就遇到了重重困难，无论是难以落地，还是知识库的关闭，以及预训练的不如人意等等。好在最终我没在任何一环中放弃，知识库问答模型是我的心血结晶，在这里我想感谢帮助我一路走来的人。

我要感谢我的指导教师黄永锋副教授对我的严格要求，黄老师总能指出我的不足，并多次给我改善模型的建议。研究生助教梁立位也在每次的汇报和验收时尽心尽责，我在课题各项验收时，受到他很多的帮助。同时我也要感谢和我一组的其他同学，虽然课题不同，但是我们间的交流为我开拓了很多思路。

我也要感谢辅导员姜霖霖，在校期间为我处理了很多学生事务，也给了我很多心理辅导。当然我还要感谢我的父母和家人，永远在背后支持我，让我专心于本课题的研究，父母的养育我永远铭记于心。

本课题虽然告一段落，但是对于该方向的研究将依然继续，感谢学校对我的培养。最后对所有关心和帮助我的人致以衷心的敬意。