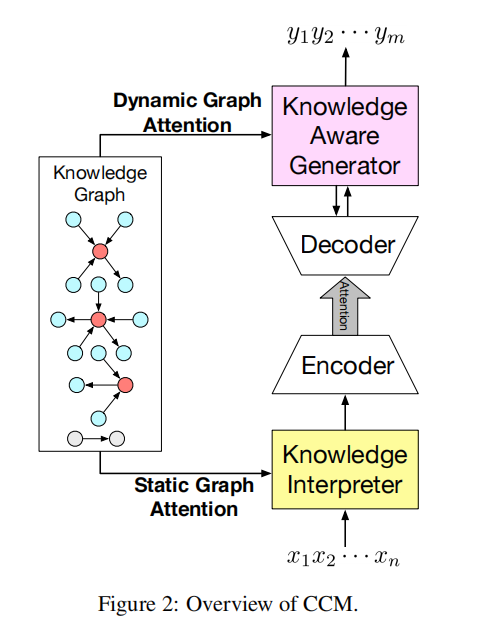
# CCM 2018

https://zhuanlan.zhihu.com/p/522033809

https://blog.csdn.net/guohui\_0907/article/details/89373118



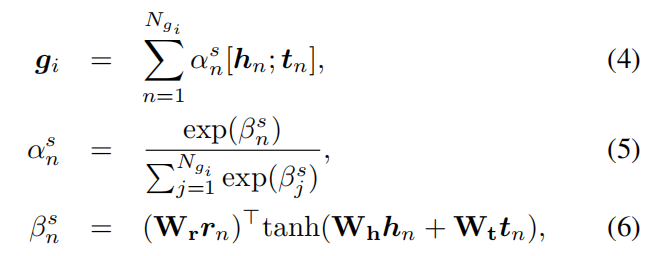
**图注意力机制**在对话生成模型中的作用是帮助模型更好地理解输入的对话历史并生成准确、连贯的回复。对话生成模型通常**需要考虑上下文中的不同部分**，并决定在生成回复时应该关注哪些信息。

图注意力机制通过将对话历史表示为图结构，并学习节点之间的关系，能够有效地捕捉上下文中的重要关联性，并根据需要将注意力集中在相关的部分上。

图注意力机制在对话生成模型中的作用是帮助模型关注对话历史中的重要信息（根据注意力机制更好理解问题），并根据需要动态地调整关注的部分。

**静态图注意力机制，**在对检索到的图编码，以充分理解问题

1. 为每个图生成一个静态的表现形式
2. 输入为图中的三元组向量
3. 输出为



、、是头实体、关系、尾实体的权重矩阵，至此蕴含“注意力”

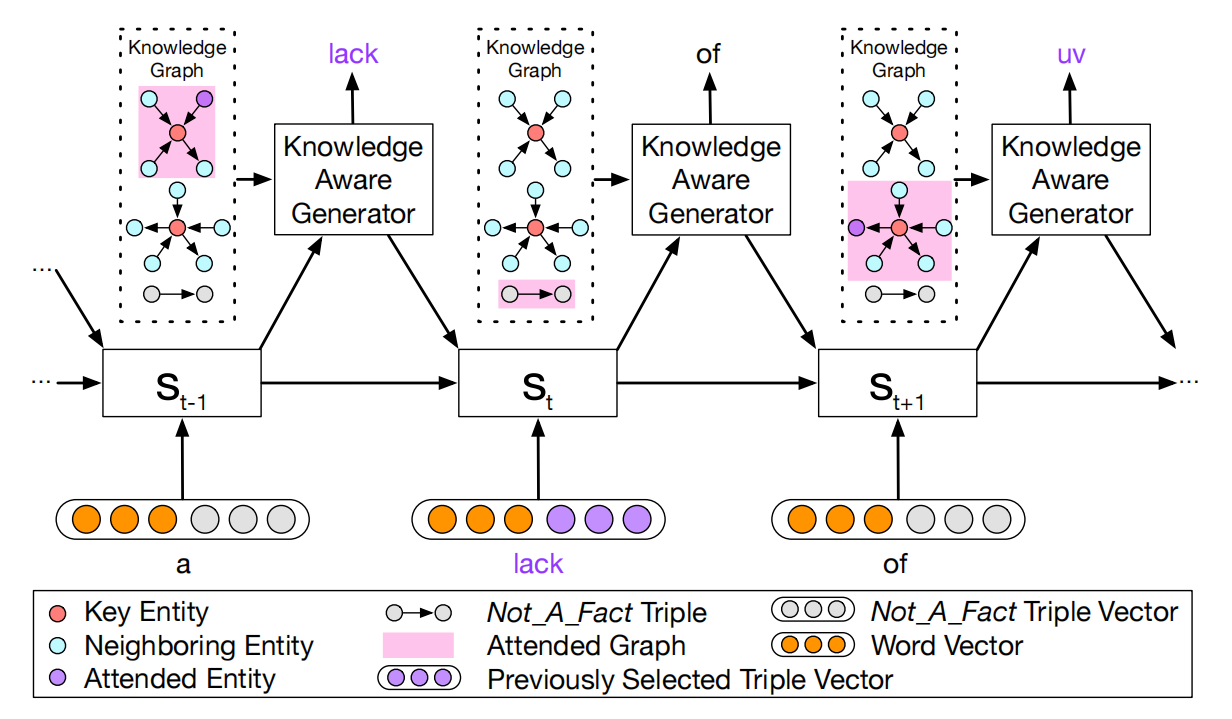
1. “注意力”的权重包含了三元组之间三个量的关联程度。

**动态图注意力机制**，读取所有的知识图和其中的三元组，利用三元组中的语义信息生成更好的回答。

**知识解析器(Knowledge Interpreter)：**

合并单词向量和对应的图向量增强每个单词的语义信息，使得模型更好的理解输入X

1. 输入为词向量，通过常识知识库检索、输出图
2. 通过静态图注意力机制把**单词向量和知识图向量相结合**，就得到了向量，然后进入编码器中的 GRU。



**GRU编码器**：

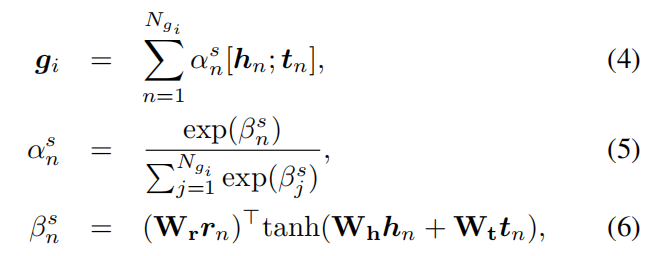
功能：将变长序列输入编码成固定长度的向量表示

控制信息流动（哪些留下哪些去除），捕捉序列中的重要信息，并生成固定长度的向量表示。

**静态图注意力机制：**

在**对检索到的图编码**，以充分理解问题

1. 为每个图生成一个静态的表现形式
2. 输入为对应图中的全部三元组向量
3. 输出为

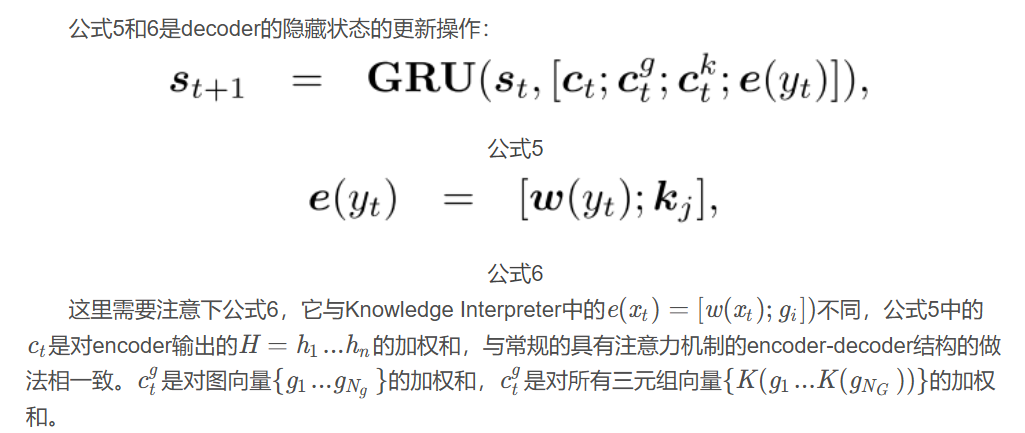


、、是头实体、关系、尾实体的权重矩阵，至此蕴含“注意力”

1. “注意力”的权重包含了三元组之间三个量的关联程度。

**知识感知生成器 (Knowledge Aware Generator)**

1. 读取所有**检索到的图**，来获取一个图感知**上下文向量**，并用这个向量来更新更新decoder的隐藏状态。

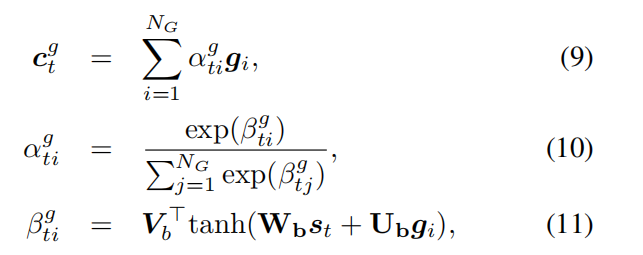


1. 自从被检索的图中或者单词表中选择**合适的单词**（通用词/实体）生成【利用动态图注意力机制】

**动态图注意力机制**

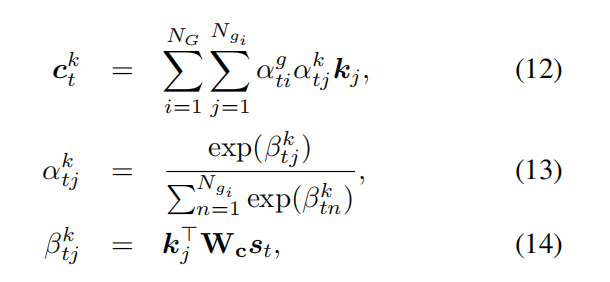
1. 利用注意力机制为图向量 做加权和得到

（其中，注意力机制计算的是decoder的**隐藏状态**和每个**图向量**的关系）

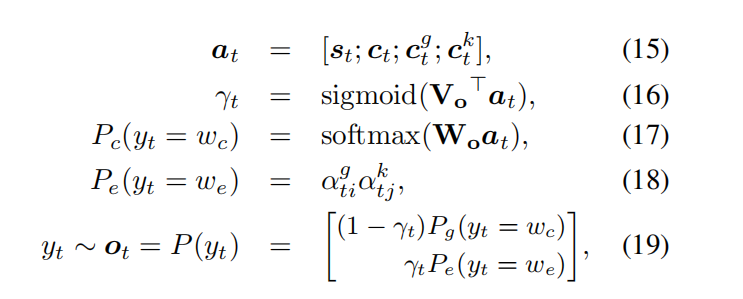


1. 利用注意力机制对每个图向量中的所有ki做加权和得到

（其中，注意力机制计算的是decoder的**隐藏状态st**和每个**三元组向量ki**的关系）



1. 通过规则生成单词：



# Mem2Seq——ACL2018

翻译：https://blog.csdn.net/qq\_27009517/article/details/89383937

略读：https://blog.csdn.net/sanra123/article/details/87890781

**面临问题：**

1. 需要将外部KG纳入RNN的隐藏状态作为参考，但是RNN在长序列上不稳定。
2. 长序列太耗时。

**MemNN**

端到端记忆网络（MemNN），需要巨大外部内存，

1. 可以记忆外部KB信息
2. MemNN的多跳机制被证明在推理任务中能够实现很高的性能
3. 但是只能从源候选池中复制出结果，不能逐字生成

所以提出**Mem2Seq**

该模型使用**顺序生成架构**扩充了现有的MemNN框架，使用全局**多跳注意机制**直接从对话历史或KB中复制单词。

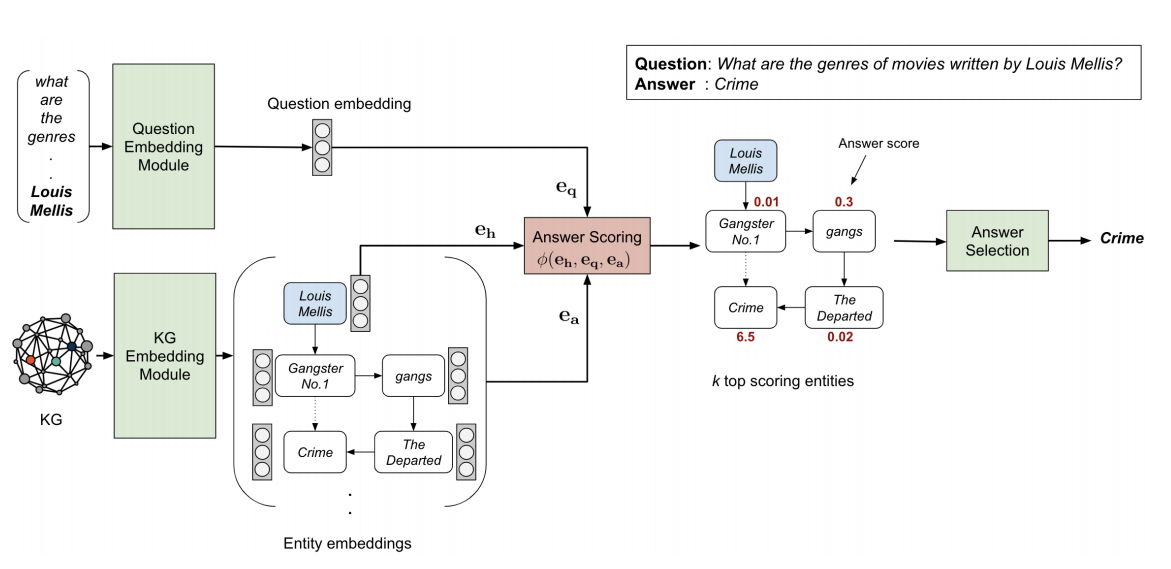
**主要贡献：**

1. Mem2Seq是第一个将**多跳注意力机制**和**指针网络**结合起来的模型，它允许我们有效的合并外部的Knowledge base信息。
2. 学习如何生成动态查询--->以控制内存访问
3. 可以更快地进行训练

# Exaqt 2021

# EmbedKGQA 2020

**<https://zhuanlan.zhihu.com/p/420777654>**



**核心：**

1. 放宽了范围参数的限制（之前的候选需要指定一个多跳数值，现在是把所有实体都看做候选）
2. 融合了知识图谱的嵌入技术

**由于EmbedKGQA将所有实体作为候选答案，它不受现有多跳KGQA方法存在的有限邻域触问题。**

**嵌入的使用使它更有效地处理KG稀疏性**

**问题描述：**

输入：已有的KG，输入的问题q，头结点eh

输出：答案et

**模型大纲：**

1. 为所有实体创建embedding（KG Embedding Module）
2. 输入了一个问题，找到这个问题对应的embedding（Question Embedding Module）
3. 选择答案（Answer Selection Module）

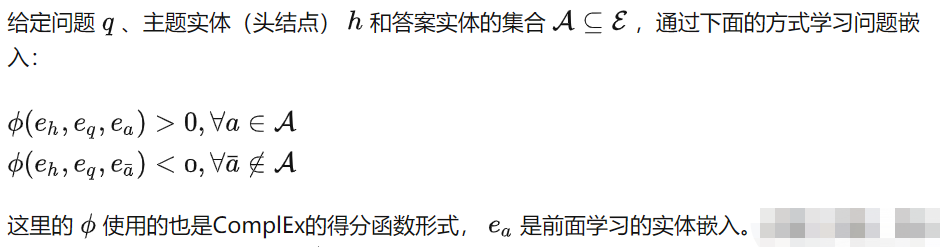
**KG嵌入模块（KG Embedding Module）**

**嵌入的使用使它更有效地处理KG稀疏性**

1. 使用ComplEx嵌入
2. 实体的嵌入结果后面会用来训练——头实体、问题和答案实体的得分函数。
3. 而根据KG实体在QA训练集中的coverage，现在这里得到的实体嵌入表示也会在后续的步骤中确定frozen还是fine-tuned（微调）。

**问题嵌入模块（Question Embedding Module）**

1. 把问题嵌入到一个固定维度向量中
2. 这个嵌入过程使用**前馈神经网络**
   1. 通过RoBERTa把问题表示成768维的向量，再通过4层全连接层和ReLU函数投影到复数空间中。
   2. 如何学习这个嵌入的网络呢？【学习到好的得分函数？不对，得分函数是固定的】

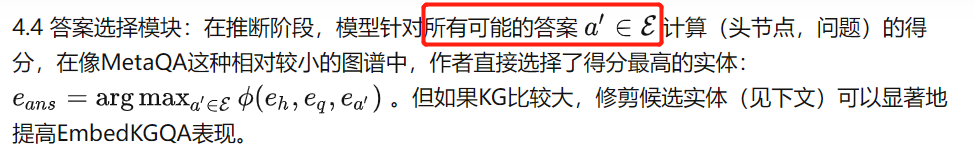


* 1. 对每个问题，得分函数是使用**所有候选答案实体**计算的，模型通过**最小化网络的输出层sigmoid结果(得分)和标签（正1负0）的二元交叉熵损失函数**来学习，在实体总数很多时，模型也进行了标签平滑。

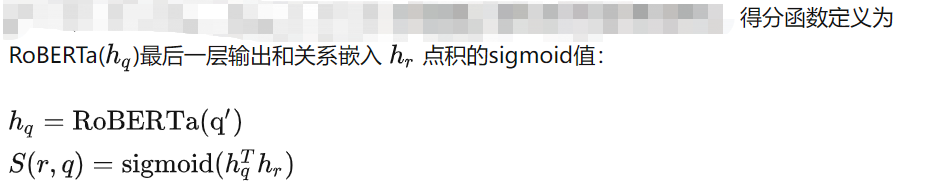
**答案选择模块（Answer Selection Module）（包含推理过程）**

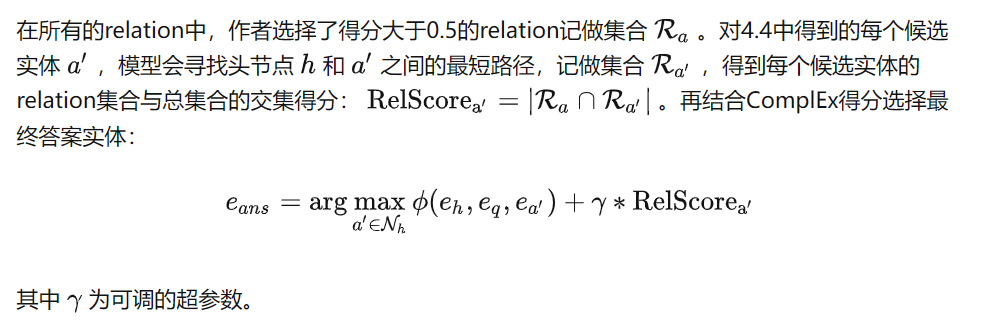
**作用：通过合并问题和关系相似性得分来选择最终答案**

1. 对所有可能的得分，如果是小的图谱，取最高分；如果是大KG，需要修剪实体。



1. 关系匹配（Relation matching）
   1. 训练了一个得分函数，针对给定问题q，给出的一个问题与关系匹配程度的得分值



* 1. 来了一个问题，要对全部关系都算一次这个得分，然后选得分大于0.5的关系记作
  2. 

# CronKGQA 2021（主要创新在于构建数据集）

<https://zhuanlan.zhihu.com/p/440178798>

**嵌入：**

1. 嵌入表示是将文本转换为向量形式的过程，其中每个单词或字符都被映射到一个高维向量空间中的定点位置。
2. 嵌入表示能够捕捉到词语之间的语义和语法信息。

**BERT的作用：**

1. 在问答模型中，我们一般需要对问题和回答进行处理，以便将其转化为机器可以理解和处理的形式。
2. BERT是一种预训练的语言模型，通过大规模的无标签文本数据进行训练，能够学习到丰富的语言知识和语义表征。
3. 可以通过BERT获得嵌入表示

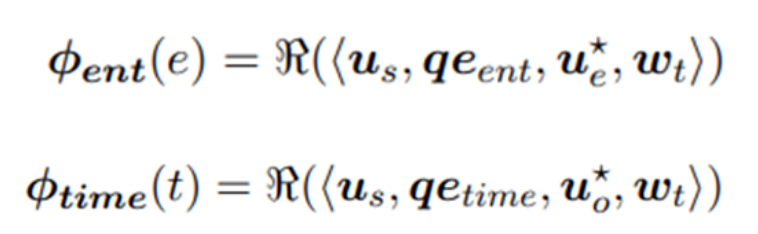
**首先使用EmbedKGQA**

由于EmbedKGQA使用ComplEx作为知识图谱嵌入的方法，并不能够处理包含时间的信息（只能处理非时间kg和单个实体问题），于是作者在这里采用了**T**ComplEx来获取实体和时间戳的embedding。

**步骤：**

1. 使用时间KG嵌入模型（EmbedKGQA用的是ComplEx，CronKGQA用的是**TComplEx**），生成时间知识图中的每个**时间戳和实体的嵌入**
2. BERT用于获得两个**问题的嵌入**：
3. 问题中提到的**实体/时间的嵌入**与使用公式4和公式5的**问题嵌入**相结合，得到实体和时间预测的得分向量。

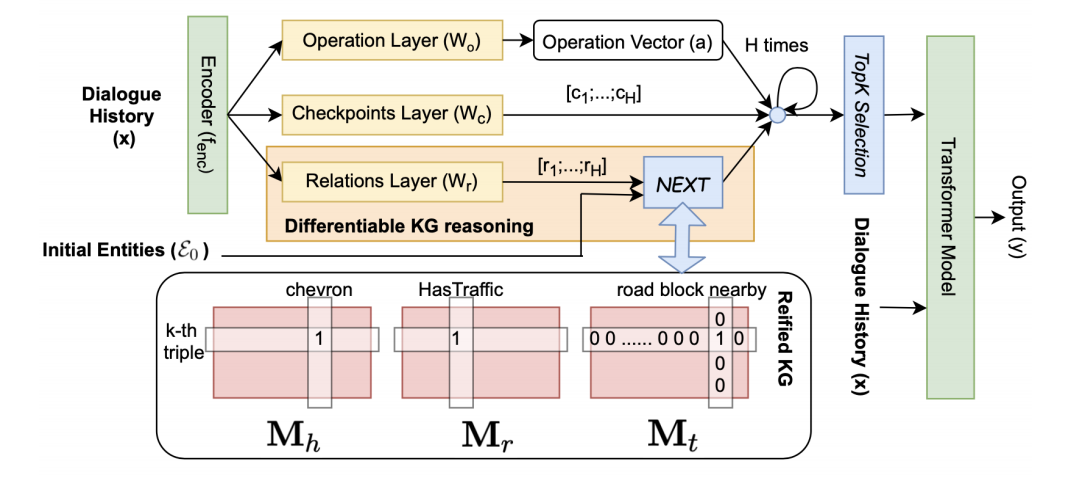
CronKGQA使用了两个预测函数获得了问题的实体嵌入和时间嵌入，而后分别计算在实体级别和时间级别的得分：（以下是在实体、时间两个级别上的 两个得分函数）



1. 使用softmax获得概率，预测最终的答案

# DiffKG 2022/03/20

https://zhuanlan.zhihu.com/p/573893418



**对话系统的三个部分：**

1. 自然语言**理解**(Natural Language Understanding, NLU)

从用户话语中提取意义

1. 对话**管理**(Dialogue Management, DM)

对意义进行推理，必要时与外部应用通信（一些额外的知识图），生成下一个系统动作

1. 自然语言**生成**(Natural Language Generation, NLG)

生成下一个文本

**DM现存问题：**

现有DM可以从外部KG中检索信息，也可以对话历史回答。但是DM可能需要手工来制作规则。

都用手工制作规则的话，想要全覆盖需要很高成本。这类交互的用户的输入查询需要时特定的，所以可能破坏用户粘性。

**现有一些办法：**

现有一些根据**对话历史**和**外部知识图**生成相应的方法，但是可能难于解释、限于小的知识图

于是设计一种新的**可微知识图谱模型**

**新的DiffKG：**

1. 是单一的transformer model
2. 生成一系列关系来表达 在reified-KG上的多跳推理
3. 用检索到的实体生成响应

此模型在一个reified-KG上获得多跳推理，并使用推理得到的实体生成回复。

**任务定义**

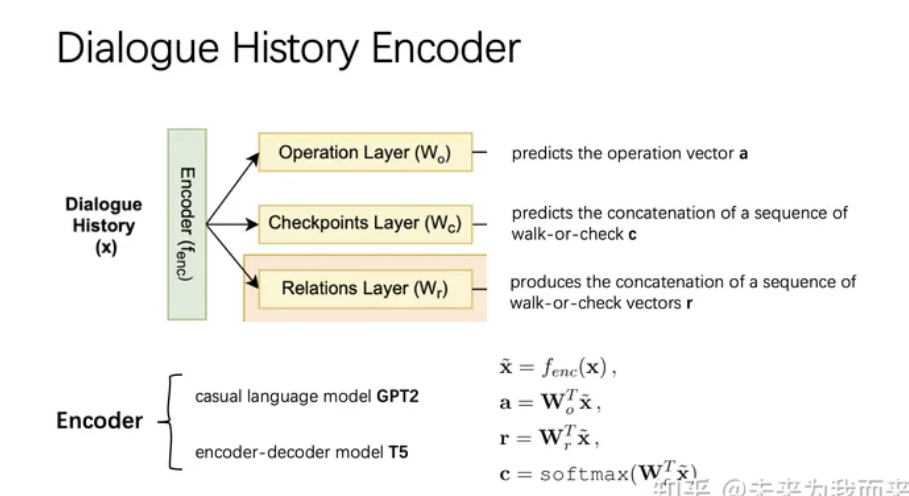
输入：对话历史X、当前可以用的KG

输出：预测的正确响应Y

这需要：准确检索KG里的信息，and对信息进一步推理

**对话历史编码dialogue history encoder**

1. 使用GPT2/T5对**对话历史**进行**编码**（两种编码论文里都有结果，T5的更好）
2. 编码的结果是——后面需要用到的各个变量

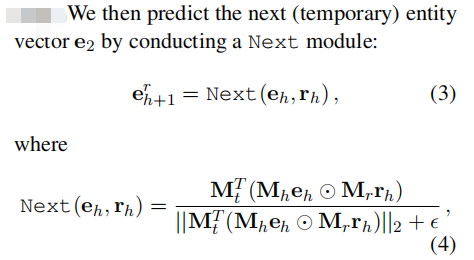


**可微KG推理模块differentiable KG reasoning module**

1. 使用reified-KG表示知识

用矩阵表示头实体、关系、尾实体，类似关系矩阵，这样就可以通过矩阵的乘法作为在知识图谱上的多跳推理。

1. 先在三个矩阵里找到第一个实体e1，然后通过Next模块计算下一个实体e2

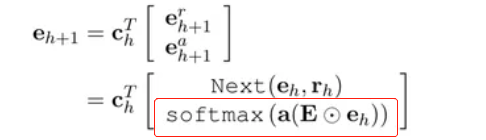


**实体嵌入Entity Embeddings**

将每个实体标记化（tokenize），然后按照token切分，再利用每个token对实体embedding。

**可学习逻辑运算模块learnable logical operations module**

1. 最终的输出是
2. 然后，使用一个walk-or-check向量 来得到最终的输出向量，用于下一步的回答生成，公式如下



1. 红框见下



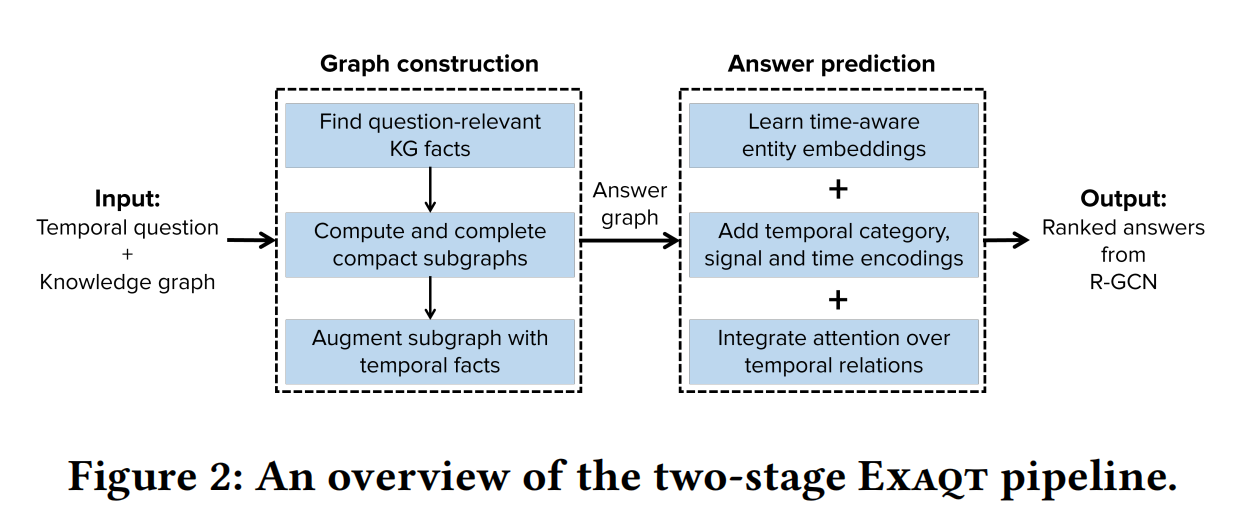
其中，实体嵌入E，第h跳处的实体向量eh，这俩的乘法是 转换后的实体嵌入

然后，操作向量a，和这个“转换后的实体嵌入”点积，这个点积结果作为实体向量传入softmax层。

**回答解码器response decoder (transformer)**

将上一步的实体进行排序，取可能性最大的前Top-k个，然后与对话历史编码向量共同输入到transformer模型中，用于生成回答。

# EXAQT



**构造答案图**

1. **寻找与问题相关的KG事实**
2. 找到所有链接的事实

NERD：根据问题中提到的单词链接到KG中，会获得几百个甚至更多的相关事实。

1. 训练分类器（判断正确/错误）：

利用BERT训练一个分类器，输入一个正例和几个包含问题中实体的负例（并不是毫无关系，错误的实例中也包含一个来自问题中的实体），进行分类。

1. 应用分类器

给定一个新的<问题，即事实>对，它将输出与该问题相关的事实的概率（和标签）。然后可以按照这个问题相关性的概率降序排序，就可以选取得分最高的事实了。

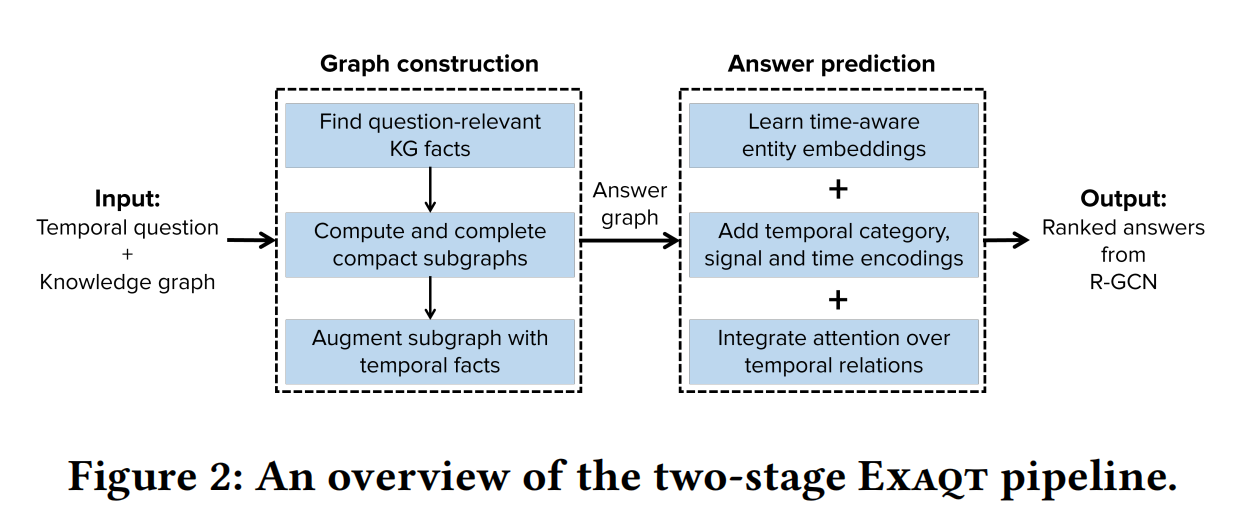
1. **计算紧凑子图**
2. 注入连接

BERT在NERD检测出来的事实中选择出来一些，但是这些可能不会相互连接，所以需要一个连通图给连接起来。

为了在由BERT事实诱导的图中注入连通性，我们计算每对问题实体之间的最短KG路径，并将这些路径添加到我们的图中。

如果两个实体之间有多条长度相同的路径，则将路径转换为一组事实后进行编码，然后计算了[CLS]（问题）和[CLS]（路径）的余弦相似度，并将余弦相似度最高的路径添加到我们的答案图中。

1. GST model.
2. Edge costs
3. GST algorithm.
4. GST completion.
5. **用时间事实来扩充子图（用时间事实增强）**
6. 首先从KG中检索已完成的GST中每个实体的所有时间事实。
7. 处理时间事实的问题相关性（使用一个类似的微调的BERT模型，该模型针对每个时间事实，**预测了其包含答案的可能性**，这个模型也是训练出来的）
8. 设已完成GST中一个实体的时间事实𝑡𝑓的似然得为𝐵𝐸𝑅𝑇（𝑡𝑓𝑞𝑟𝑒𝑙）。与前面一样，我们取得分最高的{𝑡𝑓𝑞𝑟𝑒𝑙}，将它们添加到答案图中



**用R-GCN预测答案**

R-GCN基础

**问题表示（核心是对问题的嵌入——使用时间类别、时间信号、预先训练过的单词嵌入（𝒘𝒊）、LSTM网络来进行初始化，然后通过一个FFN前馈神经网络进行更新）**

问题嵌入的初始化：使用**GRAFT-Net模型**（可以适应R-GCNs，且可以应用于异构问答，这里的异构问答指KG+txt），这个模型需要有向关系图，所以将上一步的答案图转换为有向关系图（涉及到转存）

为了编码一个时间问题，我们首先确定它的时间**类别**，并提取时间**信号**

1. 时间类别编码（TCE）

根据一些关键词和词典来判定类别。

1. 时间信号编码（TSE）

将已有定义的全部13中时间关系，设置了一组映射关系，映射到7个时间信号，然后用一个7位数表示（编码）

问题嵌入的更新：在随后的层中，问题的嵌入通过对属于它的实体（即从NERD获得的问题实体）的嵌入进行更新

**实体表示（其实就是实体嵌入）**

**实体嵌入初始化：**通过预测邻近实体来学习实体嵌入，产生更可靠的实体嵌入

**实体嵌入更新：**

时间编码（TE）：确保唯一且顺序

可感知时间的实体嵌入（TEE）

此部分利用LSTM网络更新，把和实体e相关联的所有时间事实tf按照时间编码排序，输入LSTM网络，然后这个网络的输出就是对实体e的——可感知时间的实体嵌入

注意时间关系（ATR）

Putting it together.

**答案预测**

在𝑙层（网络的最后）获得的最终实体表示，然后通过二分类选择答案。