论文引介 | Neural Generative Question Answering

原创 2016-07-13 谢若冰 智能立方

文章原名：Neural Generative Question Answering

作者：Jun Yin, Xin Jiang,Zhengdong Lu, Lifeng Shang, Hang Li, Xiaoming Li

单位：School of Electronic Engineering and Computer Science, Peking University Noah’s Ark Lab, HuaweiTechnologies

译者：谢若冰

链接：
http://arxiv.org/abs/1512.01337（可戳下方阅读原文）

1 导读

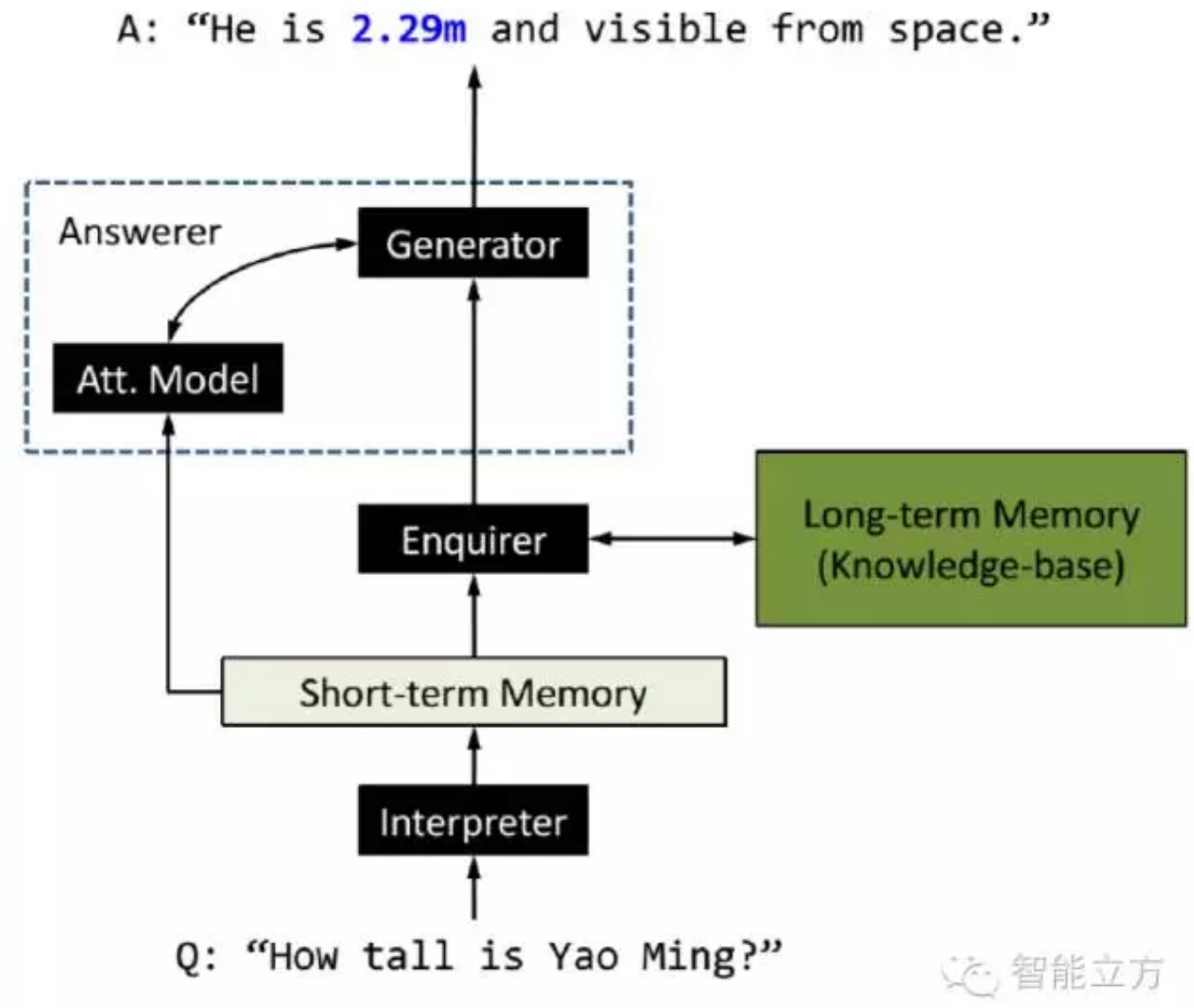
本文提出了一个基于神经网络的生成式问答模型Neural Generative QuestionAnswering (GENQA)，能够结合知识库（knowledge base, KB）信息，针对问题自动生成自然语言的回答。GENQA模型基于encoder-decoder框架，采用sequence-to-sequence模型，创新性地将QA任务与自然语言生成任务结合。论文基于中文知识库（如百度百科等）以及问答系统（如百度知道、搜狗问问等）进行了实验，实验结果证明GENQA能够针对需要知识库信息的问题作出流畅的句式回答，效果要优于其它baseline方法。

我们针对一个问答系统的实例对任务的可能挑战与实现意义进行进一步说明。例如问题是“How tall is Yao Ming?”，对应的回答应该是“He is 2.29m and visible from space”。需要注意的是，针对这样的问答任务，传统的基于检索的问答系统只能在已有数据库中进行检索，难以覆盖所有的问题，并且可拓展性较差；如果单纯采用基于神经网络的生成式语言模型，虽然能够保证自然语言的多样性，可拓展也得到提高，但是其检索结果的正确性难以保证。GENQA的基本思想，即是将基于神经网络的生成模型与知识库信息联合起来，以解决问答任务。由于KB规模很大，将其直接学进神经网络的参数中显然难以实现与拓展，并且随着memory-based neural network的发展，将KB作为外部信息使用变得自然。

Question & Answer	Triple (*subject, predicate, object*)
Q: *How tall is Yao Ming?* A: *He is 2.29m and is visible from space.*	(Yao Ming, height, 2.29m)
Q: *Which country was Beethoven from?* A: *He was born in what is now Germany.*	(Ludwig van Beethoven, place of birth, Germany)
Q: *Which club does Messi play for?* A: *Lionel Messi currently plays for FC Barcelona in the Spanish Primera Liga.*	(Lionel Messi, team, FC Barcelon)

2 模型

GENQA模型分为三个模块：Interpreter，Enquirer和Answerer，以及一个外部的知识库，其中Answerer也分为Attention Model和Generator两个子模块。整体的框架见下图：



Interpreter接收question作为输入，使用词序列对问题进行表示，输出一个向量数组表示short-term memory。GENQA采用双向RNN模型+GRU，在每个t时刻针对每个词向量x_t，会输出一个隐状态h_t。最终short-term memory的向量数组长度即为句子长度，在每个时刻t包括了隐状态向量h，词向量x以

及词自身的信息。Short-term memory将作为Enquirer以及Answerer的输入。

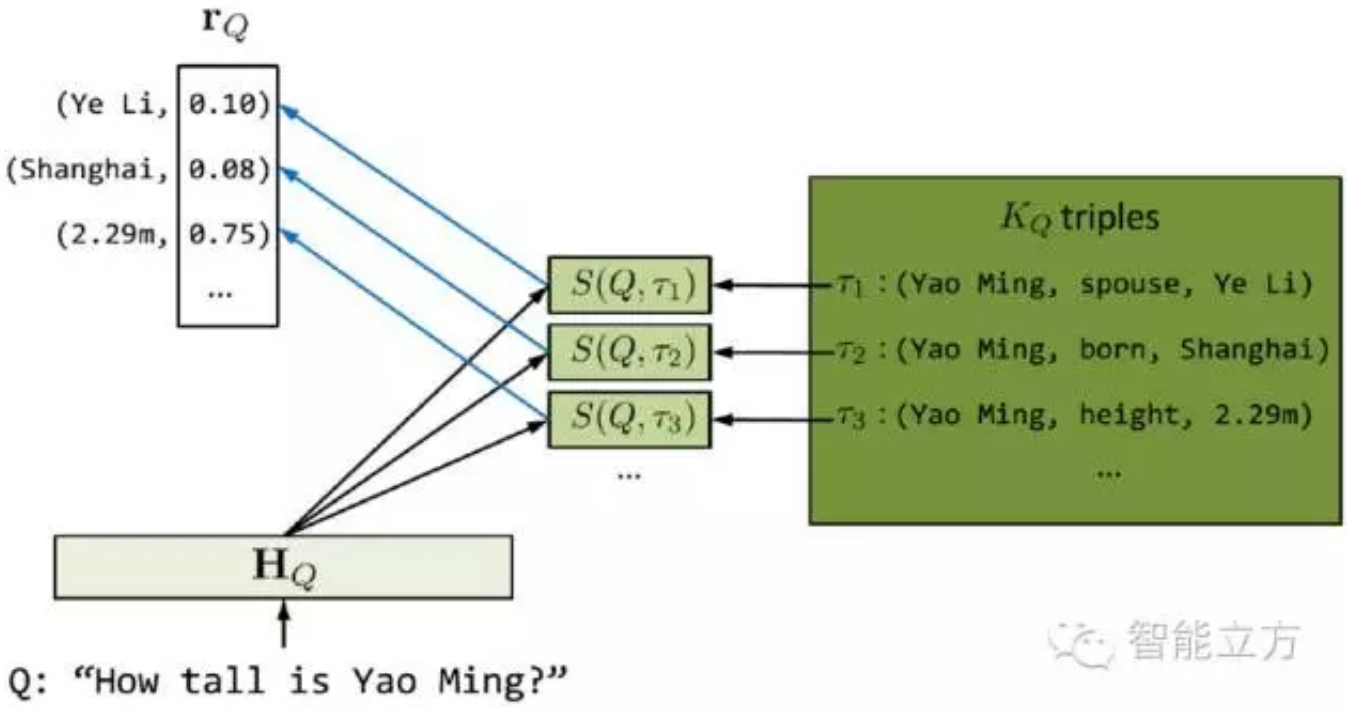
Enquirer的工作是根据知识库的知识（即long-term memory）以及对应的问题，获取正确的triple。Enquirer首先使用简单的基于term的匹配在知识库中寻找合适的三元组候选，然后计算这些候选三元组与问题之间的相似度，然后归一化作为Enquirer的输出。GENQA提出了两种计算问题与三元组之间相似度的模型，第一种是双线性模型，采用如下公式计算相似度：

$$\bar{S}(Q,\tau) = \bar{\mathbf{x}}_Q^\top \mathbf{M} \mathbf{u}_\tau,$$

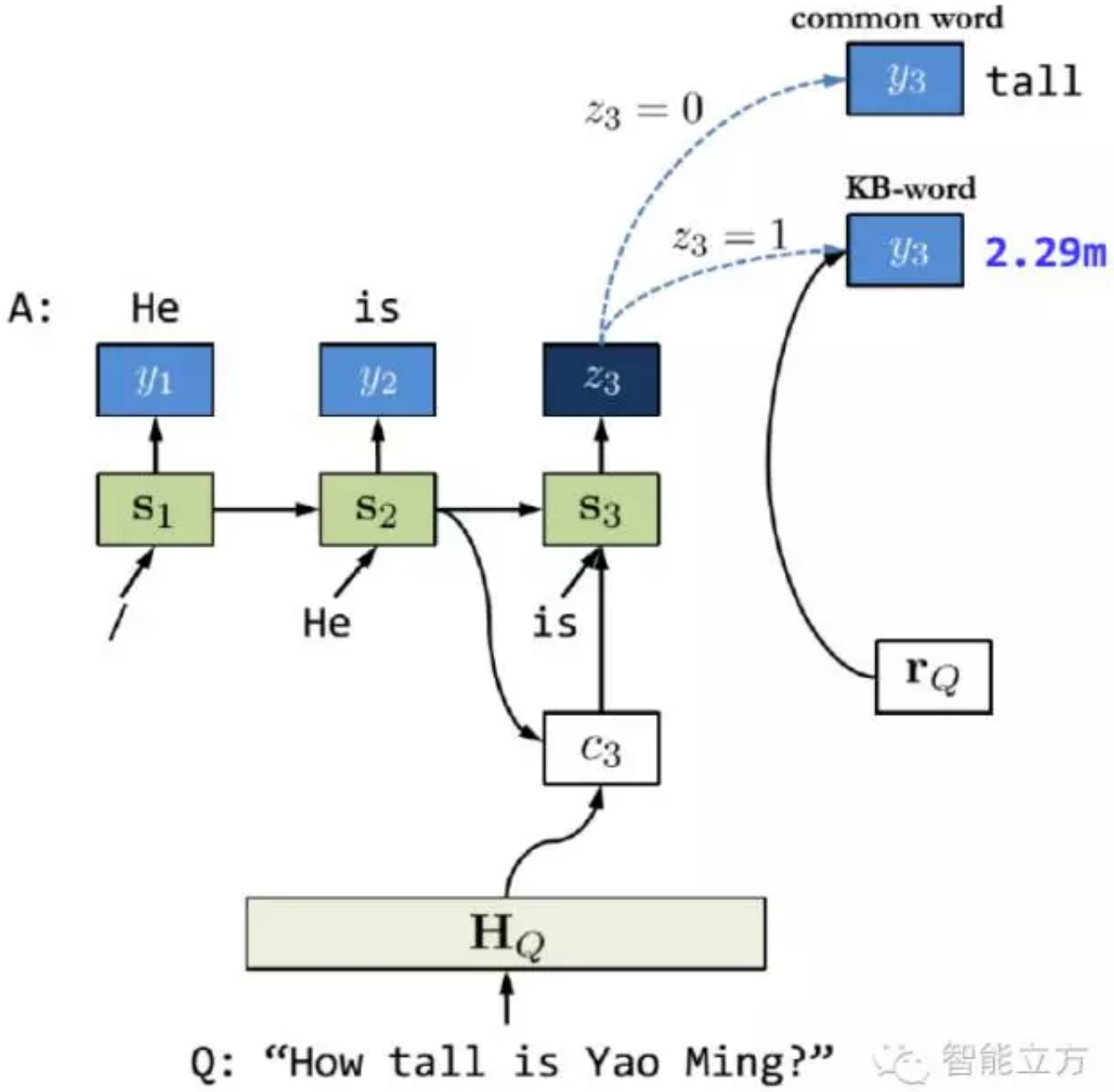
其中x表示问题序列中的词向量的均值，u表示三元组中subject（主语）与predicate（谓语）向量的均值，而M则作为待学习的参数矩阵。另一种方法是基于CNN的模型，公式如下：

$$\hat{S}(Q,\tau) = f_{\text{MLP}}([\hat{\mathbf{h}}_Q; \mathbf{u}_\tau])$$

其中h为问题序列经过单层CNN后得到的向量表示，u表示三元组中subject（主语）与predicate（谓语）向量的均值，然后两个向量拼接经过MLP得到最终的相似度得分。



Answerer接受上两个模块的输入，在知识库的指导下生成自然语言。生成过程基于普通seq2seq模型的框架，使用RNN decoder，在t时刻基于t-1时刻的隐状态、t时刻的上下文向量（context vector）生成当前词。上下文向量是基于Attention Model从short-term memory中学习到的。与通常seq2seq模型不同的是，Generator模块会使用一个逻辑斯蒂回归判断当前需要生成的是普通的词，还是知识库中的实体词。逻辑斯蒂回归使用当前时刻隐状态作为输入，输出0时生成普通的词（基于普通seq2seq的方法并考察attention），输出1时，根据知识库的long-termmemory选择知识库中对应的实体词进行生成。



3 实验

本文基于中文知识库（如百度百科等）以及中文问答系统（如百度知道、搜狗问问等）的数据进行了实验。实验采用了基于神经网络生成模型（NRM）、基于检索的模型以及基于向量的模型等三个模型作为baselines。实验的评判标准有二：问答的准确率，以及生成的自然语言流畅度。准确率的结果如下图：

Table 4: Test accuracies

Models	Test
Retrieval-based QA	36%
NRM[13]	19%
Embedding-based QA [7]	45%
GENQA	47%
GENQA$_{CNN}$	52%

下表展示了一些GENQA产生的回答，其中1~4句取得了不错的结果；5~6句虽然获得了正确的三元组信息，但是在上下文生成中产生了一些错误的信息（比如Swift是爱尔兰人而不是法国人）；7~8句获得了错误的三元组，所以结果自然就会产生错误。

阅读原文