论文阅读笔记

Step6

MF1833063, 史鹏, spwannasing@gmail.com

2019 年 7 月 9 日

1 Cognitive Graph for Multi-Hop Reading Comprehension at Scale

这篇文章基于Bert和GNN，在迭代中逐步构建出cognitive graph：$\mathcal{G}$，图中的每一个节点都和一个实体或者一个可能的答案有关。由两部分组成：implicit extraction（System 1）和explicit reasoning（System 2）。

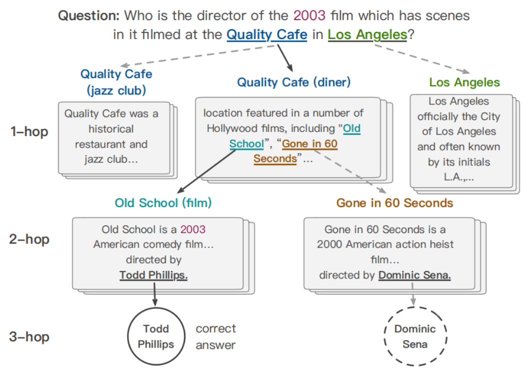

Figure 1: An example of cognitive graph for multi-hop QA. Each *hop node* corresponds to an entity (e.g., "Los Angeles") followed by its introductory paragraph. The circles mean *ans nodes*, answer candidates to the question. Cognitive graph mimics human reasoning process. Edges are built when calling an entity to "mind". The solid black edges are the correct reasoning path.

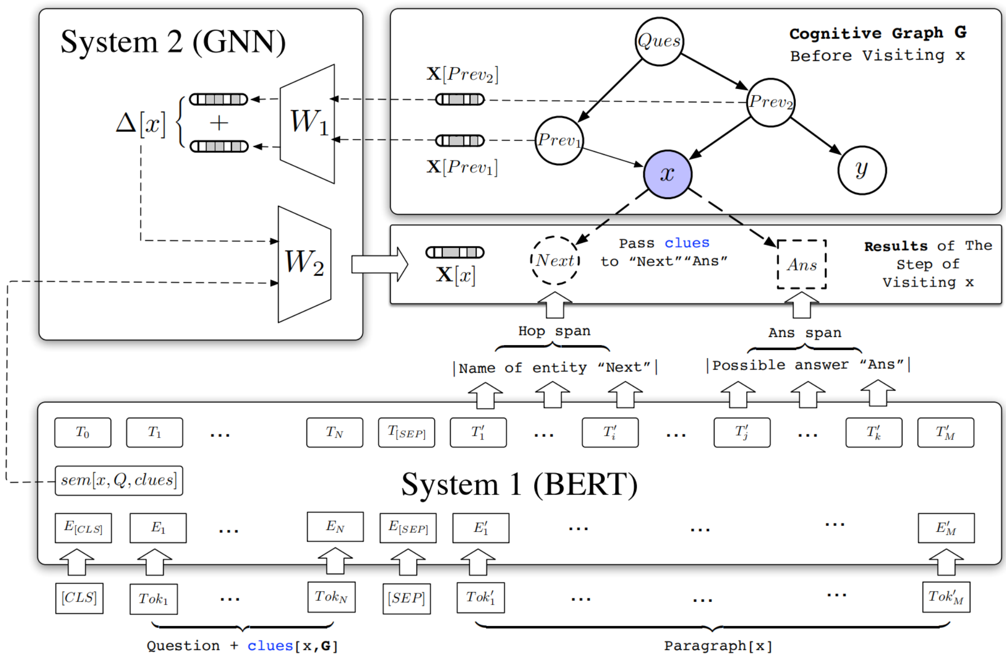

Figure 2: Overview of CogQA implementation. When visiting the node x, System 1 generates new hop and answer nodes based on the $clues[x, \mathcal{G}]$ discovered by System 2. It also creates the inital representation $sem[x, Q, clues]$, based on which the GNN in System 2 updates the hidden representations $\mathbf{X}[x]$.

Algorithm 1: Cognitive Graph QA

Input:
System 1 model $\mathcal{S}_1$, System 2 model $\mathcal{S}_2$,
Question Q, Predictor $\mathcal{F}$,Wiki Database $\mathcal{W}$

Initialize cognitive graph $\mathcal{G}$ with entities mentioned in Q and mark them *frontier nodes*
repeat
    pop a node x from frontier nodes
    collect $clues[x, \mathcal{G}]$ from predecessor nodes of x
      // eg. $clues$ can be sentences where x is mentioned
    fetch $para[x]$ in $\mathcal{W}$ if any
    generate $sem[x, Q, clues]$ with $\mathcal{S}_1$ // initial $\mathbf{X}[x]$
    if *x is a hop node* **then**
        find hop and answer spans in $para[x]$ with $\mathcal{S}_1$
        for *y in hop spans* **do**
            if *$y \notin \mathcal{G}$ and $y \in \mathcal{W}$* **then**
                create a new hop node for y
            if *$y \in \mathcal{G}$ and $edge(x, y) \notin \mathcal{G}$* **then**
                add edge (x, y) to $\mathcal{G}$
                mark node y as a frontier node
        end
        for *y in answer spans* **do**
            add new answer node y and edge (x, y) to $\mathcal{G}$
        end
    end
    update hidden representation $\mathbf{X}$ with $\mathcal{S}_2$
until *there is no frontier node in $\mathcal{G}$ or $\mathcal{G}$ is large enough*;
Return $\arg\max_{\text{answer node } x} \mathcal{F}(\mathbf{X}[x])$

[注] 本文选取的数据集是HotpotQA，HotpotQA 是一个大型问答数据集，它包含约 11.3 万个具备上述特征的问答对。也就是说，这些问题要求问答系统能够筛选大量的文本文档，以找到与生成答案相关的信息，并对找到的多个支撑性事实进行多步推理，从而得出最终答案。

System 1从段落中提取与问题相关的实体和answer candidate，并对其语义信息进行编码。提取的实体被组织成一个Cognitive Graph。然后，系统2对图进行推理，并收集线索指导系统1更好地提取下一跳实体。

System 1(Bert)：

$$\underbrace{[CLS]Question[SEP]\,\text{clues}[x, \mathcal{G}][SEP]}_{\text{Sentence } A}\ \underbrace{\text{Para}[x]}_{\text{Sentence } B} \tag{1.1}$$

System 2(GNN)：

$$\begin{aligned} \Delta &= \sigma\left(\left(AD^{-1}\right)^T \sigma\left(\mathbf{X}W_1\right)\right) \\ \mathbf{X}' &= \sigma\left(\mathbf{X}W_2 + \Delta\right) \end{aligned} \tag{1.2}$$

2 Exploiting Explicit Paths for Multi-hop Reading Comprehension

本文基于WikiHop数据集，在该数据集中问题以三元组的形式出现$(h_e, r, ?)$，h_e代表head entity，r代表head entity和未知的tail entity之间的关系。任务是从给定的candidates集合中选出一个答案:$(c_1, c_2, \ldots, c_N)$。

所做的工作是在预测答案的同时，将推理的path展示出来。方法很intuitive，但难的是想到并去做这个工作。

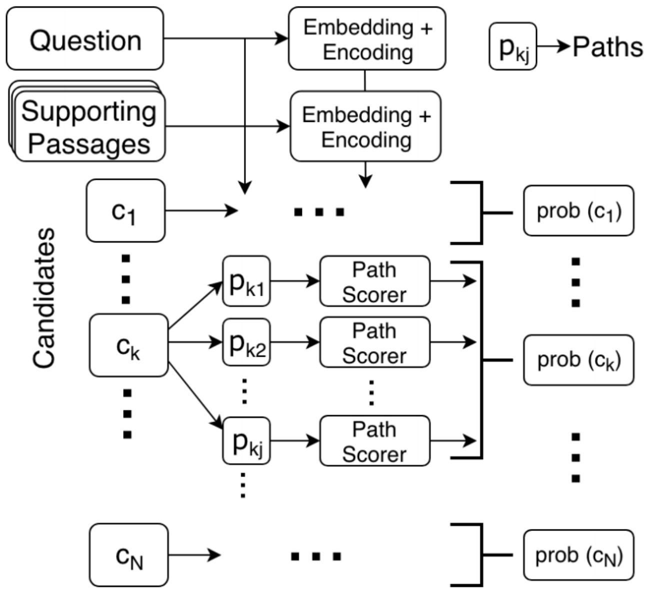

图 1: Architecture of the proposed model

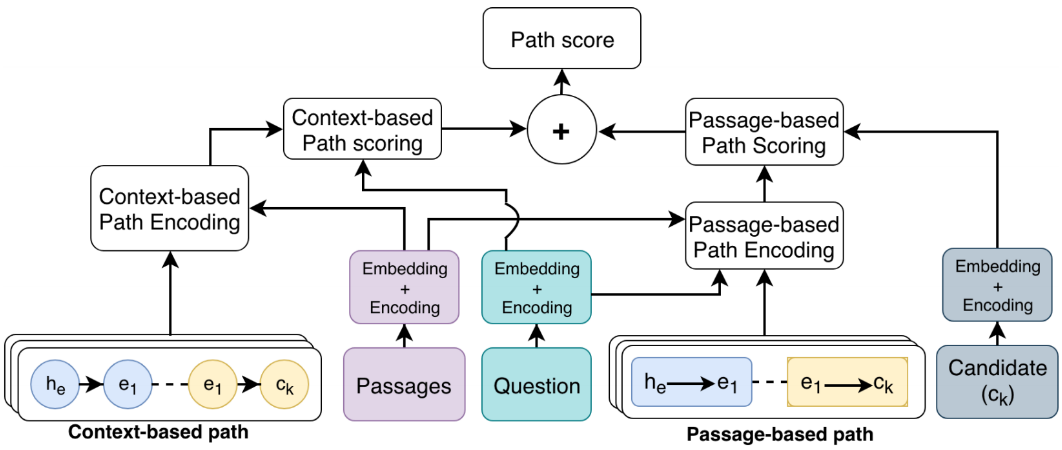

图 2: Architecture of the proposed path scoring module

模型的流程是：

1. 提取Path 在含有h_e的第一篇文章中，找出在那句话或者下一句话中出现的所有实体，然后在其它的passage中寻找这些实体，如果该passge也包含candidate中的单词那么则构建出了一个path。为每一个candidate构建一个从h_e的路径。

2. 对所有的path进行encoding以及score，选出answer以及输出path。

   [1] Context-based Path Encoding:

   对于2-hop的path：$(h_e, e_1), (e_1, c_k)$:

$$\mathbf{g}_{h_e} = \mathbf{s}_{p_1,i_1} \| \mathbf{s}_{p_1,i_2} \tag{2.1}$$

$$\mathrm{FFL}(\mathbf{a}, \mathbf{b}) = \tanh\left(\mathbf{a}\mathbf{W}_a + \mathbf{b}\mathbf{W}_b\right) \tag{2.2}$$

$$\mathbf{r}_{h_e,e_1} = \mathrm{FFL}\left(\mathbf{g}_{h_e}, \mathbf{g}_{e_1}\right) \tag{2.3}$$

$$\mathbf{x}_{ctx} = \mathrm{FFL}\left(\mathbf{r}_{h_e,e_1}, \mathbf{r}_{e_1,c_k}\right) \tag{2.4}$$

   [2] Passage-based Path Encoding:

   首先计算相似矩阵：$\mathbf{A}_p \in \mathbb{R}^{T\times U}$，然后分别计算question-aware passage和passage-aware question：$\mathbf{S}_p^{q_1} = \mathbf{A}\mathbf{Q}$和$\mathbf{Q}_p = \mathbf{A}^\top \mathbf{S}_p$，根据更新的question表示再计算$\mathbf{S}_p^{q_2} \in \mathbb{R}^{T\times H}$，其中$\mathbf{S}_p^{q_2} = \mathbf{A}\mathbf{Q}_p$

   然后将两次计算的结果拼接：$S_p^q \in \mathbb{R}^{T\times 2H} = \mathbf{S}_p^{q_1} \| \mathbf{S}_p^{q_2}$。

$$\begin{aligned} a_t^p &\propto \exp\left(\mathbf{s}_{p,t}^q \mathbf{w}^\top\right) \\ \tilde{\mathbf{s}}_p &= \mathbf{a}^p \mathbf{S}_p^q \end{aligned} \tag{2.5}$$

$$\mathbf{x}_{psg} = \mathrm{FFL}\left(\tilde{\mathbf{s}}_{p1}, \tilde{\mathbf{s}}_{p_2}\right) \tag{2.6}$$

   [3] Path Scoring:

$$\tilde{\mathbf{q}} = \left(\mathbf{q}_0 \| \mathbf{q}_U\right) \mathbf{W}_q \tag{2.7}$$

$$\mathbf{y}_{x_{ctx},q} = \mathrm{FFL}\left(\mathbf{x}_{ctx}, \tilde{\mathbf{q}}\right) \tag{2.8}$$

$$z_{ctx} = \mathbf{y}_{x_{ctx},q} \mathbf{w}_{ctx}^\top \tag{2.9}$$

$$z_{psg} = \tilde{\mathbf{c}}_k \mathbf{x}_{psg}^\top \tag{2.10}$$

$$z = z_{ctx} + z_{psg} \tag{2.11}$$