论文阅读笔记

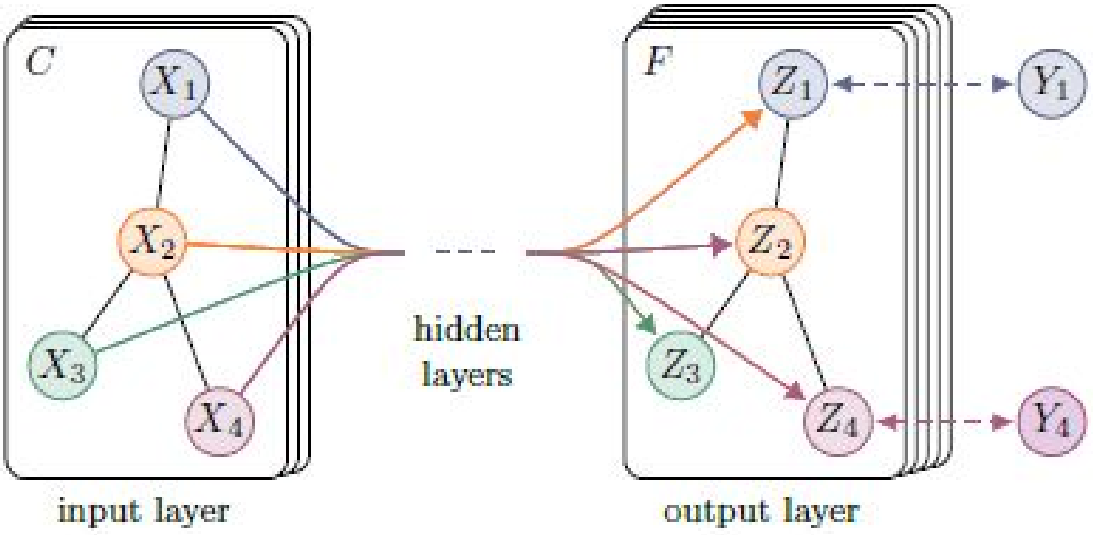
Step8

MF1833063, 史鹏, spwannasing@gmail.com

2019 年 9 月 16 日

1 SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS

本文提出了一种图卷积网络（graph covolutional networks, GCNs），该网络是传统卷积算法在图结构数据上的一个变体，可以直接用于处理图结构数据。从本质上讲，GCN 是谱图卷积（spectral graph convolution）的局部一阶近似（localized first-order approximation）。GCN的另一个特点在于其模型规模会随图中边的数量的增长而线性增长。总的来说，GCN 可以用于对局部图结构与节点特征进行编码。



图卷积神经网络的（单层）最终形式： $H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$

2 Semantic-Unit-Based Dilated Convolution for Multi-Label Text Classification

本文是基于seq2seq的多标签文本分类在attention上的一些改进工作。主要贡献有两点：

1.提出了所谓的“语义单元”，因为在多标签文本分类中，word-level的作用没有那么大，而是“semantic units”来决定文本的分类。

2.使用了Hybrid Attention将semantic units和word level的attention混合起来。

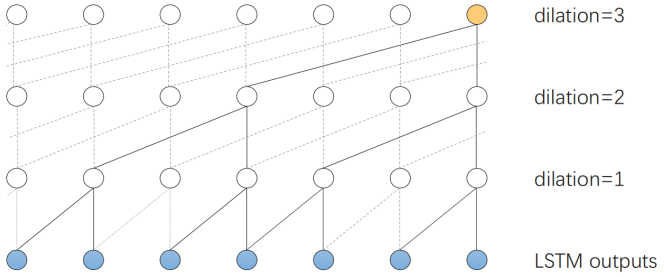


Figure 1: **Structure of Multi-level Dilated Convolution (MDC).** A example of MDC with kernel size $k = 2$ and dilation rates $[1, 2, 3]$. To avoid gridding effects, the dilation rates do not share a common factor other than 1.

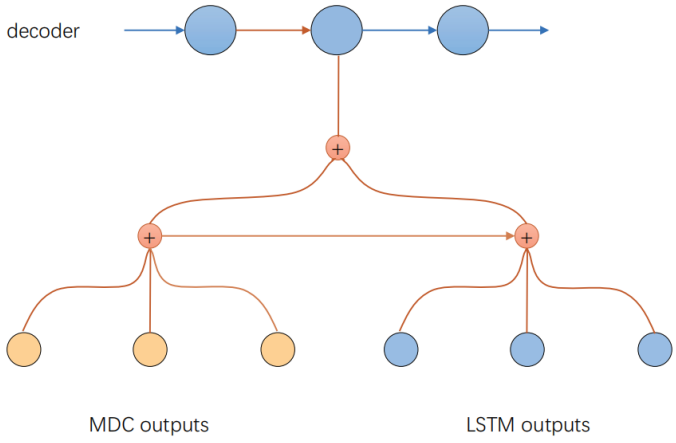


图 1: Structure of Hybrid Attention.

3 A Deep Reinforced Sequence-to-Set Model for Multi-Label Classification

本文的主要Motivation是解决多标签文本分类中Seq2Seq模型的输出序列的顺序问题，因为label之间本来应该是无序的，即交换不变性，这里所做的工作则是提出用强化学习来解决label之间的顺序问题，reward即为预测label序列和答案之间的F1值。

$$\mathcal{L}(\theta) = -\mathbb{E}_{\boldsymbol{y}\sim p_\theta}[r(\boldsymbol{y})] \tag{3.1}$$

$$\nabla_\theta \mathcal{L}(\theta) \approx -\left[r\left(\boldsymbol{y}^s\right) - r\left(\boldsymbol{y}^g\right)\right] \nabla_\theta \log\left(p_\theta\left(\boldsymbol{y}^s\right)\right) \tag{3.2}$$

$$r(\boldsymbol{y}) = \mathrm{F}_1\left(\boldsymbol{y}, \boldsymbol{y}^*\right) \tag{3.3}$$

其它的结构基本一致。

4 Compositional Questions Do Not Necessitate Multi-hop Reasoning

这篇文章主要是提出了在HotPotQA数据集中存在的一个问题：所谓的multi-hop其实不是必要的，大多数问题可以在只提供sigle paragraph的情况下就正确的回答出来。然后提出了一个Single-Paragraph QA模型。

分别将paragraph送入Bert

$$S' = \mathrm{BERT}(S) \in \mathbb{R}^{h\times(m+n+1)} \tag{4.1}$$

然后选取y_{empty} 最小的作为答案输出。

$$[y_{\text{span}}\,; y_{\text{yes}}\,; y_{\text{no}}\,; y_{\text{empty}}\,] = W_1\ \text{maxpool}\ (S') \tag{4.2}$$

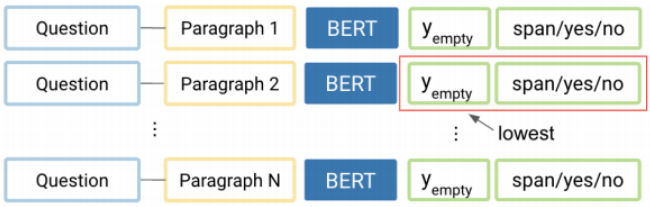


Figure 2: Our model, single-paragraph BERT, reads and scores each paragraph independently. The answer from the paragraph with the lowest y_{empty} score is chosen as the final answer.

5 Attention Guided Graph Convolutional Networks for Relation Extraction

motivation是现有的方法采用基于规则的硬剪枝策略来选择相关的部分依赖结构，但并不总是能得到最优的结果。本文提出了使用GCN来soft-pruning自动学习如何有选择地处理对关系提取任务有用的相关子结构。被称为Attention Guided Graph Convolutional Networks。

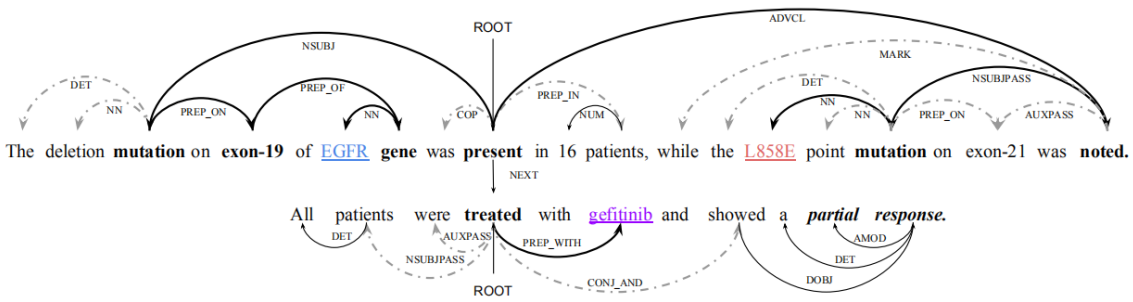


Figure 1: An example dependency tree for two sentences expressing a relation (sensitivity) among three entities. The shortest dependency path between these entities is highlighted in bold (edges and tokens). The root node of the LCA subtree of entities is *present*. The dotted edges indicate tokens K=1 away from the subtree. Note that tokens *partial response* off these paths (shortest dependency path, LCA subtree, pruned tree when K=1).

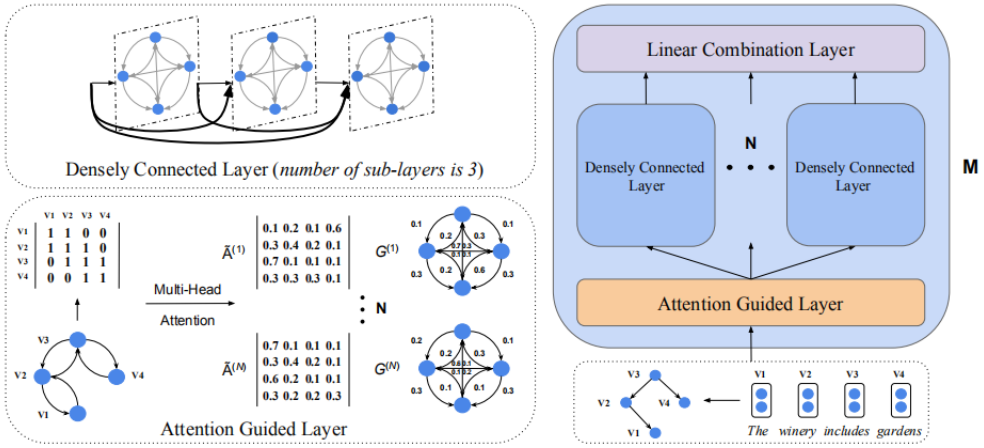


Figure 2: The AGGCN model is shown with an example sentence and its dependency tree. It is composed of M identical blocks and each block has three types of layers as shown on the right. Every block takes node embeddings and adjacency matrix that represents the graph as inputs. Then N attention guided adjacency matrices are constructed by using multi-head attention as shown at bottom left. The original dependency tree is transformed into N different fully connected edge-weighted graphs (self-loops are omitted for simplification). Numbers near the edges represent the weights in the matrix. Resulting matrices are fed into N separate densely connected layers, generating new representations. Top left shows an example of the densely connected layer, where the number (L) of sub-layers is 3 (L is a hyper-parameter). Each sub-layer concatenates all preceding outputs as the input. Eventually, a linear combination is applied to combine outputs of N densely connected layers into hidden representations.

GCNs：

$$\mathbf{h}_i^{(l)} = \rho\left(\sum_{j=1}^{n} \mathbf{A}_{ij}\mathbf{W}^{(l)}\mathbf{h}_j^{(l-1)} + \mathbf{b}^{(l)}\right) \tag{5.1}$$

Attention Guided Layer：

$$\tilde{\mathbf{A}}^{(\mathrm{t})} = \mathrm{softmax}\left(\frac{Q\mathbf{W}_i^Q \times \left(K\mathbf{W}_i^K\right)^T}{\sqrt{d}}\right) \tag{5.2}$$

Densely Connected Layer：

$$\mathbf{g}_j^{(l)} = \left[\mathbf{x}_j; \mathbf{h}_j^{(1)}; \ldots; \mathbf{h}_j^{(l-1)}\right] \tag{5.3}$$

$$\mathbf{h}_{t_i}^{(l)} = \rho\left(\sum_{j=1}^{n} \tilde{\mathbf{A}}_{ij}^{(t)} \mathbf{W}_t^{(l)} \mathbf{g}_j^{(l)} + \mathbf{b}_t^{(l)}\right) \tag{5.4}$$

Linear Combination Layer:

$$h_{comb} = W_{comb} h_{out} + b_{comb}$$

$$h_{out} = [h^{(1)}; ...; h^{(n)}]$$

6 Learning a Deep ConvNet for Multi-label Classification with Partial Labels

本文解决的问题是只有Partial Label的多标签分类问题。贡献点一个是提出了一个新的针对partial的损失函数，其次使用GNN来对multi-label之间的关系进行建模，最后还能通过此算法来对没有提供label的那一部分进行预测。

$$l(x,y) = \frac{g(p_y)}{C}\sum_{c=1}^{C}[1_{[y_c=1]}log(\frac{1}{1+exp(-x_c)}) + 1_{[y_c=-1]}log(\frac{exp(-x_c)}{1+exp(-x_c)})]$$

这是对应的损失函数，其中p_y是该条数据中，有标签的label所占的比例，g是正则化函数。

GNN用于Multi-Label：

Message update：

$$m_v^t = \frac{1}{|\Omega_v|}\sum_{u\in\Omega_v} f_M(h_u^t)$$

Hidden state update:

$$h_v^{t+1} = GRU(h_v^t, m_b^t)$$

7 Inferential Machine Comprehension: Answering Questions by Recursively Deducing the Evidence Chain from Text

这篇文章主要是解决阅读理解中的多跳推理问题，这里提出了一个Operation Cell，以及一个termination机制，在train的过程中，递归的执行Cell，termination机制负责决定什么时候停止。

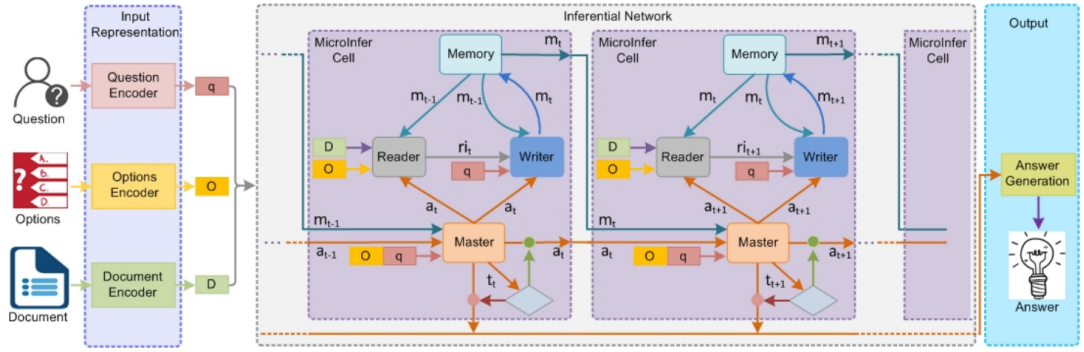


图 2: Overview

主要有四个组成部分：

1. Memory：负责存储中间结果
2. Master：用self-attention来分析question，在每一个step上focus在特定的aspect上
3. Reader：根据question aspect和text content来抽取出相关的text content
4. Writer：整合previous的results，生成新的临时结果。

8 A STRUCTURED SELF-ATTENTIVE SENTENCE EMBEDDING

介绍了一种提取句子embedding的方法。通过attention机制，提取出句子重要的r个部分。类似r个head的Attention，然后引入类似KL散度的惩罚项，鼓励不同的head计算出不同的侧重点。

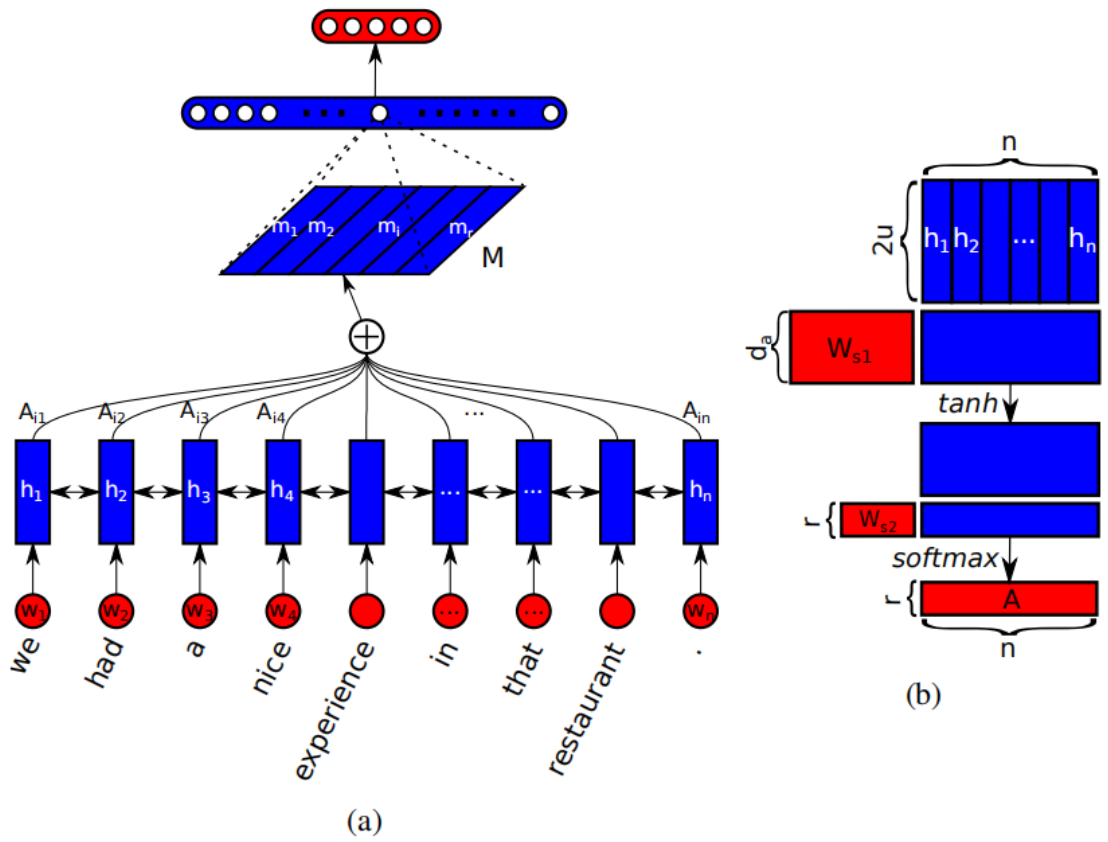


图 3: Overview

9 Coherent Comment Generation for Chinese Articles with a Graph-to-Sequence Model

本文解决的问题是自动生成文章评论,提出了一个模型：将文章表示为Topic之间相互联系的图，然后又提出了一个graph-to-sequence模型来生成评论。

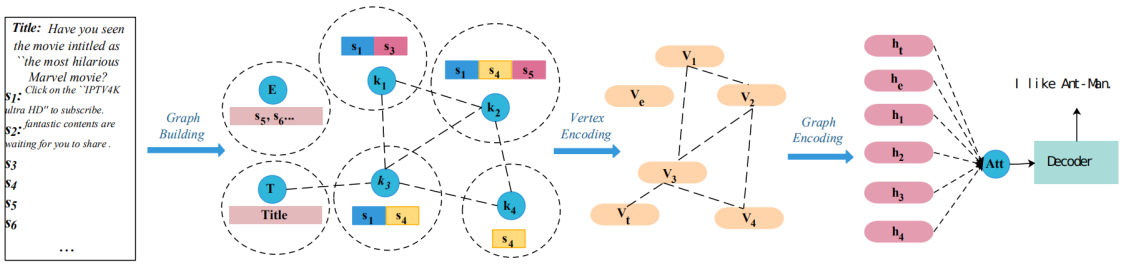


Figure 1: A brief illustration of our proposed graph-to-sequence model. A vertex in the interaction graph consists of a topic word k_i and the sentences containing k_i. If a sentence contains no topic word, it is archived to a special "Empty" vertex. Each vertex is first encoded into a hidden vector v_i by the vertex encoder. Then the whole graph is fed into the graph encoder and get the final vertex representation h_i encoded with structure information. A RNN decoder with attention mechanism is adopted to generate comment words.

1.Graph Construction：

用TextRank算法抽取关键词，每一个vertex包含一个keyword和含有这个词的sentence，然后如果一个sentence中包含多个keyword，那么对应的节点之间建立联系。

2.Vertex Encoder：

先Embedding，然后Self-Attention。然后用keyword的vector来表示这个节点。

3.Graph Encoder：

用GCN来更新节点。

4.Decoder：

送入带有attention机制的RNN来输出结果。

10 BERT with History Answer Embedding for Conversational Question Answering

就是在原本的bert的每一个word的embedding中添加了一个所谓的History Answer Embedding (HAE)，来标记这个单词在history中有没有出现过。而这里的history selection则是naive的选取最近的几个question-answer pair。

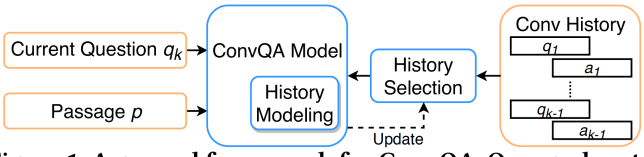


Figure 1: A general framework for ConvQA. Orange denotes model input and blue denotes model components.

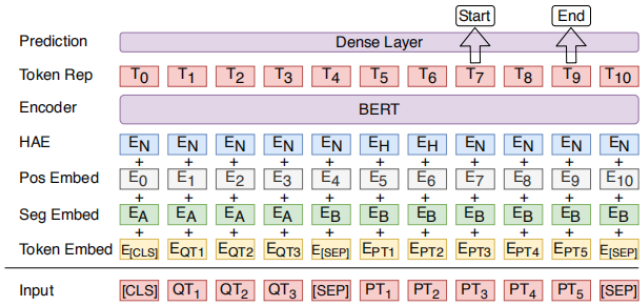
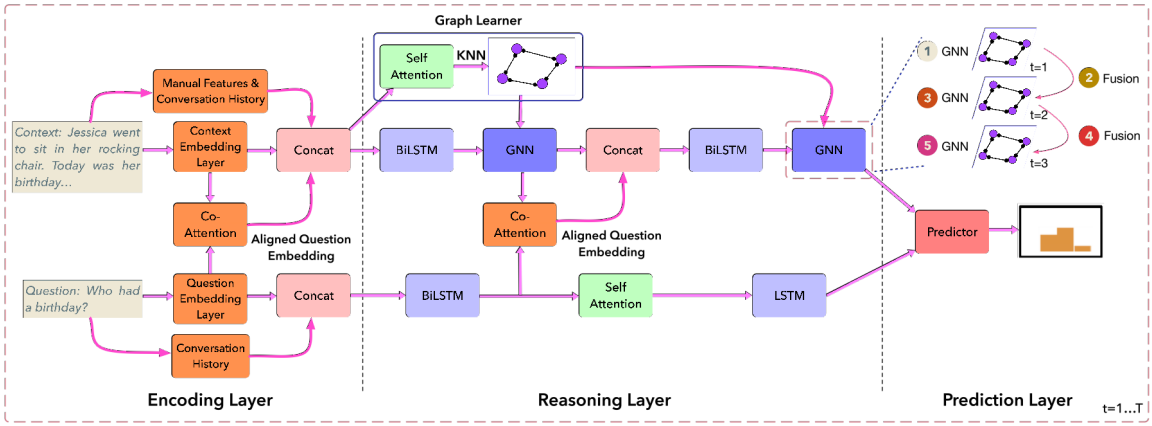


Figure 2: Architecture of the ConvQA model with HAE. E_H/E_N in HAE denote the token is in/not in history answers.

11 GRAPHFLOW: Exploiting Conversation Flow with Graph Neural Networks for Conversational Machine Comprehension

对话型阅读理解的工作，不知道是不是得益于Bert，效果比FlowQA得F1高0.8。



主要创新：

1.构建图的方式：

$$A = ReLU(UW_C^{(i)})^T ReLU(UW_C^{(i)})$$

这里就用self-attention机制来构建一个attention矩阵来充当邻接矩阵的功能，另外避免过多的计算开销以及全连接的话大多数相邻点其实没有多少作用,所以用了一个KNN-style的方法，过滤掉大部分无用的消耗。

2.GraphFlow过程中的Fuse：

$$C_i^l = GNN(\overline{C_i^{l-1}}, \widetilde{A_C^{(i)}})$$
$$\overline{C_i^{l-1}} = Fuse(C_{i,j}^{l-1}, C_{i-1,j}^l)$$

Advance：

构件图的方式感觉不是很intuitive，有待改进。另外，跟进一下和FlowQA的对比试验，采用同样的Embedding。