Lecture 17. Multitask Learning as Question Answering

Ⅰ. Intro

POS 태깅, 감정 분석, 번역의 모델들을 아예 joint하게 묶어서 일반적인 질문에 대답하는 형태로 만들 수 있지 않을까해서 Question Answering이 등장하였다. 해당 작업에는 2가지 어려움이 있다.

1) Task 별로 최고 성능을 내는 구조가 모두 다르고 task(POS 태깅, 감정 분석, 번역)과 상관없이 최고의 성능을 내는 neural architecture가 없다.

2) fully joint learning이 상당히 어렵다는 것이다. Fully joint learning은 각각 모델의 레이어를 모두 이어 붙여서 함께 트레이닝 하는 것이다. 실제로 joint learning은 하위 레이어만 떼다가 붙여 쓰는 경우가 많다. 특히, word vector의 경우 이미 학습된 word vector의 하위 레이어만 갖다 붙이는 거고 (하위 레벨의 learned weight만 갖다 쓰면서 업데이트) 실제로 fully joint learning해서 좋은 성능을 낸 연구는 찾기 어렵다.

Ⅱ. Multitask Question Answering Network(MQAN)

총 5개의 모듈로 이루어져 있다. 모듈을 나눠쓰면 좋은 점은 task에 따라서 필요한 부분만 교체하거나 스위치를 끌 수 있다는 점이다. 특히 Answer Module이 softmax를 포함하는 classification model이라면, 이 부분만 바꿔서 task에 필요한 답을 내도록 할 수 있다.

1) Semantic Memory Module : 주로 pre-trained된 word2vec
2) Input Module : GRU로 모든 input phrase(or sentence)의 hidden state을 계산. 각 문장의 last hidden state(deep GRU의 마지막 layer)에 접근이 가능하고 3번의 episodic memory module에서는 이 값을 input으로 쓴다.
3) Question Module : GRU로 question sentence을 vectorize함. Attention을 포함하고 있어서 input module에 있는 last hidden states에 접근할 수 있음.

$$q_t = GRU(v_t, q_{t-1})$$

4) Episodic Module : GRU로 attention에 따라서 input module의 input 중 question과 관련있는 문장을 골라내는 역할을 함. Input module에 있는 모든 input을 여러 번 걸쳐서 훑어서 각기 다른 문장에 attention을 준다. 각 key는 각 iteration(=pass) 마다 하나의 input에 다른 attention을 준다. 그리고 최종적인 m만 answer module에 넘긴다.

$$h_i^t = g_i^t GRU(s_i, h_{i-1}^t) + (1 - g_i^t) h_{i-1}^t$$

(i : current time step, t : 몇 번째 pass인지, g : 문장에 얼만큼의 attention을 줘야할지를 결정하는 single scalar number)

g는 문장이 question과 relevant할 경우 더 많은 attention을 준다. 아래 수식과 같은 벡터간의 유사도를 이용하여 relevance를 결정한다.

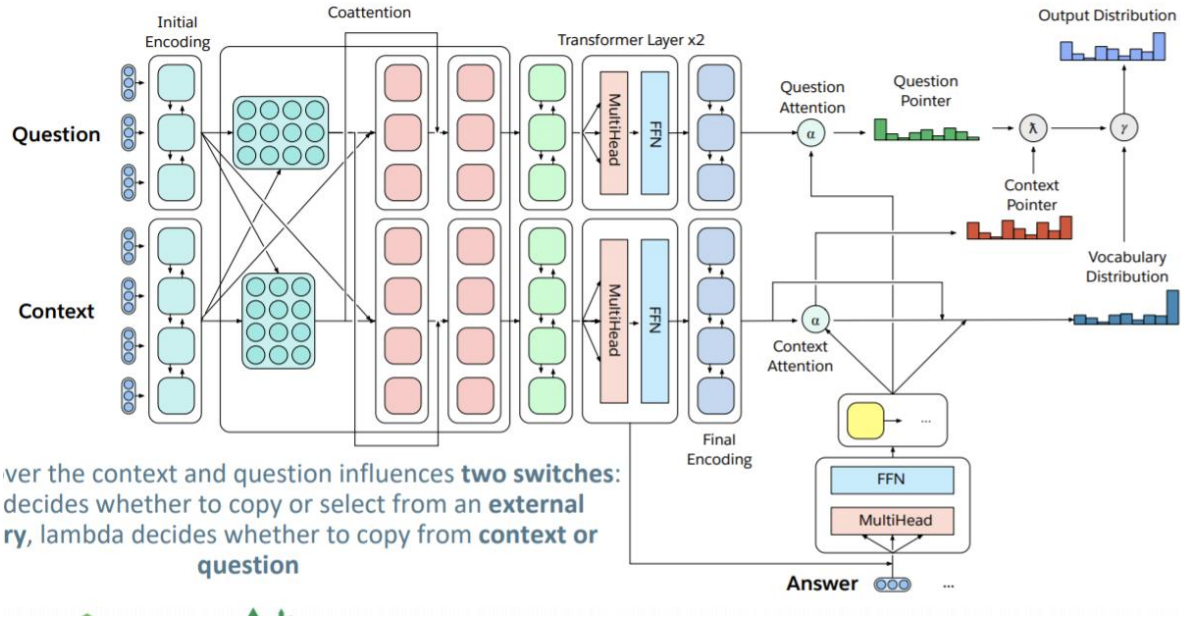
$$z_i^t = [s_i \circ q ; s_i \circ m^{t-1} ; |s_i - q| ; |s_i - m^{t-1}|]$$

$$Z_i^t = W^{(2)} \tanh\left(W^{(1)} z_i^t + b^{(1)}\right) + b^{(2)}$$

$$g_i^t = \frac{\exp(Z_i^t)}{\sum_{k=1}^{M_i} \exp(Z_k^t)}$$

5) Answer Module : GRU로 softmax를 가지고 정답을 만든다. Candidate 중에서 가장 높은 확률이 있는 단어(혹은 구)를 리턴하기 때문에 이 전에 나온 문장 혹은 단어가 아니라면 답을 못한다.

이 모든 모듈을 end-to-end로 학습시킬 수 있다. 즉, Answer module에서 낸 답이 정답이 아니었으면 모든 모듈을 거쳐서 semantic module까지 error signal을 보내 backpogate한다.



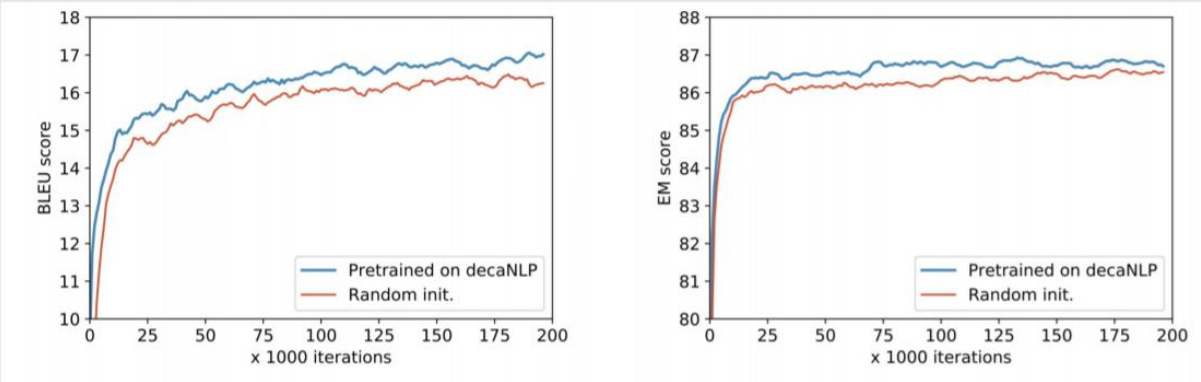
Ⅲ. Training Strategies

1. Fully joint : 각각의 batch에 task를 하나씩 넣음.
2. Anti-curriculum Pre-training : single-task setting에서 수렴하기 위한 얼마나 많은 반복이

필요한지 모른다. Reddish Tasks(pretraining phrase에 포함되는 tasks)

IV. decaNLP

여러 개의 NLP tasks에 대해 하나의 question answering model을 학습. 더 일반적인 언어 이해가 가능하고 multitask learning이 가능하다. Domain adaptation과 transfer learning, weight sharing, pre-training, fine-tuning, zero-shot learning이 가능하다.



decaNLP로 pretrained한 모델의 성능이 좋음을 알 수 있다.