

lecture 10 preview

1. Motivation & History

두 가지 답변의 형태

- Finding **documents** that contain an answer - information retrieval(IR)
- Finding an **answer** in a paragraph or a document - machine reading comprehension(MRC)

2. Stanford Question Answering Dataset(SQuAD)

질문의 분류

- factoid type questions [what, which, when, who, how] - SQuAD에서 다루는 형태의 질문
- list type questions [list of facts or answers]
- confirmation questions [Y/N]
- casual questions [why, how]
- hypothetical questions [no specific answers]
- complex questions

SQuAD(1.1)

- QA dataset은 Passage, Question, Answer로 구성된다.
- A는 P에 속한다.
- 데이터의 평가방법
 - exact match - 예측값과 실제값이 정확히 일치하면 1, 아니면 0
 - f1 score - 예측값과 실제값의 중첩토큰을 계산

SQuAD(2.0)

- 1.1과 비교했을 때, added unanswerable question이 추가

- 데이터셋의 unanswerable questions는 온라인의 crowd worker에 의해 생성된다.
- noanswer의 처리
 - 임계값 이상의 결과에 대해서만 답을 예측하고 이하의 결과에 대해서는 noanswer
 - Machine Reading Comprehension with Unanswerable Questions (2019 AAAI)
 1. read then verify
 2. 정답과 noanswer 확률을 계산
 3. 정답이 적절한지 확인한다.

SQuAD의 한계

- span-based answers만 존재한다
- P내에서만 정답을 찾으려 하는 질문 구성
- 동일 지시어 문제를 제외하고 Multifact, 문장추론 문제가 거의 없음

→ **BUT SQuAD is well-targeted , well-structured, clean dataset**

3. QA model

Stanford Attentive Reader - Bi LSTM with attention

a. Question vector 생성

- 임베딩된 단어들을 glove에서 가져와 one-layer BiLSTM에 넣는다.
- BiLSTM의 각방향의 은닉층을 연결하여 question vector를 얻을 수 있다.

b. Passage vector 생성

- BiLSTM의 각방향의 은닉층을 연결하여 문장 내 단어의 개수만큼 passage vector를 생성한다.

c. Attention

- a_i : i 개의 P벡터, 1개의 Q벡터를 이용하여 어텐션 적용, 소프트맥스 취하기
- o_s : a_i 벡터와 p_i 벡터를 모두 더함
- a_s : o_s linear transformation

Stanford Attentive Reader++

- one layer BiLSTM 이 아닌 3 layer BiLSTM을 사용
- Question vector 구성할 때 BiLSTM state를 포지션 별로 연결 후 weighted sum 을 구한다.
- Passage vector 구성할 때 embedding에 extra feature를 추가한다. (word embedding, linguistic features, term frequency, exact match, aligned question embedding...)

BiDAF : Bi-Directional Attention Flow for Machine Comprehension

SAR은 $Q \rightarrow P$ 단방향으로 진행되는 반면 이건 양방향으로 진행된다.

1. Character Embedding Layer

- Character Embedding Layer : charCNN을 이용해 단어를 벡터공간에 매핑

2. Word Embedding Layer

- Word Embedding Layer : pre-trained word embedding model 이용하여 단어를 벡터공간에 매핑

3. Contextual Embedding Layer

- 이 둘을 연결하여 d-dimension vector 생성, BiLSTM을 거쳐 2d-dimension vector 생성

4. Attention Flow Layer

- Attention Layer : Query to Context, Context to Query
 - Context2Query Attention
 - S_t - t번째 context word와 query word의 유사도
 - S_t 에 소프트맥스를 취하여 context word 의 입장에서 유사성이 높은 쿼리의 특성을 알게된다.
 - Query2Context Attention
 - S 에서 maxcol을 뽑아 softmax를 취한다. 쿼리와 유사도가 높은 context word만 남게 된다.

5. Modeling Layer - 2-layer-bidirectional LSTMs

6. Output Layer

