

# Ask Me Anything : Dynamic Memory Networks for Natural Language Processing



link

<https://open.spotify.com/playlist/6BWeQjYssdDcaxyzCLAIJ7?si=116d020bf2324d4b>

## 00 Abstract

natural language processing → question answering

introduce the dynamic memory network, DMN

process input seq → form episodic memory → generate answer

attention process (on the input, the result of previous iteration)

⇒ QA(Facebook's bAbI), text classification(Stanford Sentiment Treebank), sequence modeling (WSJ-PTB)

각 학습은 오직 훈련된 벡터 표현 & triplet (input-question-answer) 에 의존

## 01 Introduce

QA = 복잡한 자연어 처리 문제로 치환 가능

(문맥에 의미와 관련 정보의 신뢰성을 판단하는 능력이 필요)

번역 = 이 문장을 프랑스어로 번역하면 어떻게 돼?

### Dynamic Memory Network, DMN

원본 triplet으로 학습, 문답 문제 해결을 위한 신경망 베이스의 프레임워크

답변 메커니즘은 각 입력 단어마다 유발된다. (instead of only at the end)

sequence tagging, sequence-to-sequence, question answering (transitive reasoning : 이전 정보 바탕으로 다음 단계의 문제 해결)

- 입력, 질의에 대해 표현(representation) 계산
- attention process : 입력과 연관된 정보 검색
- 검색된 정보 바탕으로 vector representation 생성

## 02 Dynamic Memory Networks

- input module : raw text input → distributed vector representation
- question module : question → distributed vector representation. Episodic module 로 이어져 기초/초기 상태를 이룬다.
- episodic memory module : 입력 중 어텐션으로 집중할 부분을 결정한다. memory vector representation 생성. 입력에서 기존의 정보와 다른 새로운 정보를 찾아낼 능력이 있다.

### Input module

sentence  $T$ , word  $w$

- encoding : RNN
  - RNN 을 활용하는 방법 :  $t$  마다 update hidden state →  $h_t = \text{RNN}(L[w_t], h_{t-1})$
  - 입력 시퀀스가 하나인 경우 →  $h$  출력
  - 입력 시퀀스가 문장 리스트인 경우 → 문장의 끝을 나타내는 end-of-sentence token 삽입하여 하나의 단어 리스트로 합친다. end-of-sentence token이 있을 때마다 출력되는  $h$  는 입력 모듈의 final representation
  - $T_c$  는 fact representation  $c$  일 때의 output ,  $c_t$ 는 output sequence의  $t$ 번째 요소를 표현한다.

- choice of recurrent network : Gated Recurrent Network(GRU)
  - complex LSTM : 비슷한 성능, 높은 계산 비용
  - standard tanh RNN : 낮은 성능
  - (GRU) gated → 그래디언트 소실 문제를 완화
  - GRU의 internal mechanics

$$z_t = \sigma \left( W^{(z)} x_t + U^{(z)} h_{t-1} + b^{(z)} \right) \quad (1)$$

$$r_t = \sigma \left( W^{(r)} x_t + U^{(r)} h_{t-1} + b^{(r)} \right) \quad (2)$$

$$\tilde{h}_t = \tanh \left( W x_t + r_t \circ U h_{t-1} + b^{(h)} \right) \quad (3)$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t \quad (4)$$

where  $\circ$  is an element-wise product,  $W^{(z)}, W^{(r)}, W \in \mathbb{R}^{n_H \times n_I}$  and  $U^{(z)}, U^{(r)}, U \in \mathbb{R}^{n_H \times n_H}$ . The dimensions  $n$  are hyperparameters. We abbreviate the above computation with  $h_t = GRU(x_t, h_{t-1})$ .

## Question Module

input과 마찬가지로, 질의는 자연어 문장으로 주어진다.

input module과 같이, RNN을 통해 인코딩을 수행한다.

- share the word embedding matrix
  - 입력 모듈, 질문 모듈 사이에서 워드 임베딩 행렬을 공유
  - 입력, 질문 모듈에서 단어를 같은 방식으로 변환한다.
- output ← the final hidden state
  - 입력 모듈에서는 여러 문장이 입력될 수 있고, 문장마다 hidden state 업데이트
  - question module 에서는 최종 hidden state만 사용하여 출력 생성
  - $q = q(TQ)$

# Episodic Memory Module

internal episodic memory 업데이트 해가며, 입력 모듈의 출력에 대해 반복적으로 작동한다.

attention mechanism으로 구성되어 있고, 메모리를 업데이트 하는 과정에서 recurrent network가 포함된다.

어텐션 메카니즘이 representation  $c$ 에 대해 작동할 때, question representation previous memory를 고려하여 하나의 episode 를 만들어낸다

( $c$ , question representation, previous memory)  $\rightarrow$  episode

- Need for Multiple Episodes
  - iteration  $\rightarrow$  to attend to different inputs
  - allows for a type of transitive inference  $\leftarrow$  uncover the need to retrieve additional fact
    - 반복 처리로 점진적인 정보 확보
    - 축구공이 어디 있는가?  $\rightarrow$  (1) 축구공  $\rightarrow$  (2) 축구공이 어디에?
    - sentiment analysis 에서 활용 가능
- Attention mechanism
  - scoring function  $G$  (candidate fact, previous memory, question )  $\Rightarrow$  주어진 질문에서 어떤 정보(fact) 가 중요한지
  - supervised, standard cross-entropy

$$g_t^i = G(c_t, \hat{m}^{i-1}, q).$$

between input, memory and question vectors:  $z(c, m, q) =$

$$\left[ c, m, q, c \circ q, c \circ m, |c - q|, |c - m|, c^T W^{(b)} q, c^T W^{(b)} m \right] \quad (5)$$

$$G(c, m, q) =$$

$$\sigma \left( W^{(2)} \tanh \left( W^{(1)} z(c, m, q) + b^{(1)} \right) + b^{(2)} \right). \quad (6)$$

- Memory Update Mechanism
  - 에피소드 계산을 위해 modified GRU + 위의  $g_i$ 로 가중 처리된 인풋
  - GRU의 최종 hidden state가 답변 모듈로 전달된다.
  - hidden state 업데이트와, 에피소드 계산 식

$$h_t^i = g_t^i GRU(c_t, h_{t-1}^i) + (1 - g_t^i) h_{t-1}^i \quad (7)$$

$$e^i = h_{T_C}^i \quad (8)$$

- Criteria for Stopping
  - input 에 special end-of-passes representation 적용
  - maximum number 지정
  - The whole module is end-to-end differentiable : 입력 ~ 출력까지 모델 피드백 및 수정이 필요하지 않다.

## Answer Module

에피소드 메모리에서 얻은 정보를 바탕으로 답변 생성

- 에피소드 메모리 마지막에서 trigger / time step 마다 trigger
- 답변 생성
  - GRU : 마지막 메모리 상태로 초기화, 질의 벡터 & last hidden state & previous predicted output 를 입력받아 답변 생성
    - time step 별 다음 단어 예측, GRU 작동 방식

$$y_t = \text{softmax}(W^{(a)} a_t) \quad (9)$$

$$a_t = GRU([y_{t-1}, q], a_{t-1}), \quad (10)$$

- 마지막에 생성된 단어 + 질문 벡터  $\Rightarrow$  input
- cross-entropy 사용하여 정답 예측
- 답변 종료  $\rightarrow$  end-of-sequence

- sequence modeling : 단어 레이블링, 두두 번째 답변부터 서로 다른 에피소드로 인해 게이트의 값 또한 달라진다.

## Training

supervised classification problem (cross-entropy err 줄이기 위함)

각 모듈이 상호작용하기 때문에 역전파, gradient descent 사용

## 03 Related Work

관련 있어 보이는 연구 간략히 정리

### Deep learning

- Recursive neural network(RNN, 재귀 신경망) : 계층 구조 데이터, 트리 구조
  - 문장 분석 sentiment analysis, QA (NLP)
  - DMN의 episodic, question model 없음 - 추론 불가
- chain-structured RNN : 연결된 RNN
  - sequence data 처리
- seq-to-seq : LSTM 기반(enc-dec)
  - 시퀀스 변환, 번역

### Attention and Memory

- Attention Mechanisms : 중요 부분에 집중
  - 메모리, 질문의 특정 정보 선택&처리
- Neural Turing Machines : 신경망 + memory
  - 알고리즘 문제 해결 가능
  - DMN도 비슷하게 메모리 기능이 포함됨

- Memory Network : NLP + memory component
  - 문장간 연관성 파악
  - DMN은 이와 달리 문장들을 연속적으로 (독립X) → 복잡한 추론을 가능하게 함
- Stanford Sentiment Treebank (데이터셋-감성분석)

## Neuroscience

- The episodic memory in humans stores specific experiences
  - 인간은 시간적, 공간적인 맥락에서 경험을 기억
  - 영감 - 실제 인간 뇌의 작동 방식

# 04 Experiments

세가지 NLP task 수행

- Question Answering
- Part-of-Speech Tagging
- Sentiment Analysis

전체 DMN구조 동일, 작업에 따라 입력/ 답변 처리 방식이 달라진다. (단어, 문장에 따라)

- 역전파
- Adam
- GloVe embedding

## 실험 결과

- Question Answering
  - 페이스북 - bAbI
  - 메모리 네트워크(MemNN) 능가 ← 메모리와 에피소드의 반복
- Part-of-Speech Tagging
  - WSJ-PTB
  - SVMTool, Soegaard, Suzuki 등 보다 뛰어난 성능
  - 단일 순차 모델, 다른 작업과 같은 구조로도 높은 성능
- Sentiment Analysis
  - Stanford Sentiment Treebank
  - binary 감성 분석, fine-grained 감정 분석
  - DMN : 다중(1-3) 패스 로 중요 정보 반복 분석. 패스가 추가되었을 때 더 높은 성능
- Episodic memory module
  - 에피소드의 수가 증가할수록 좋은 추론, 복잡한 추론이 필요할 때 좋은 성능을 보인다.

## 05 Conclusion

- **DMN → NLP의 문제에 적용 가능한 모델 제안**
  - trained end-to-end
  - → multi task model, multi modal input&question