Hierarchical Attention Networks for Document Classification

...

허진규

Abstract & introduction

- 문서를 해석하는 순서적인 방법에 있어서 새로운 방법을 제안한다.

- 문서는 문장, 문장은 단어로 이루어짐
- 문맥에 따라 단어의 의미가 바뀜

- CNN-GRNN, LSTM-GRNN 처럼 Hierarchical 개념 사용
- Attention 개념 사용

2 skills in this Architecture

1. bidirectional RNN

2. 2 encoders , 2 attentions

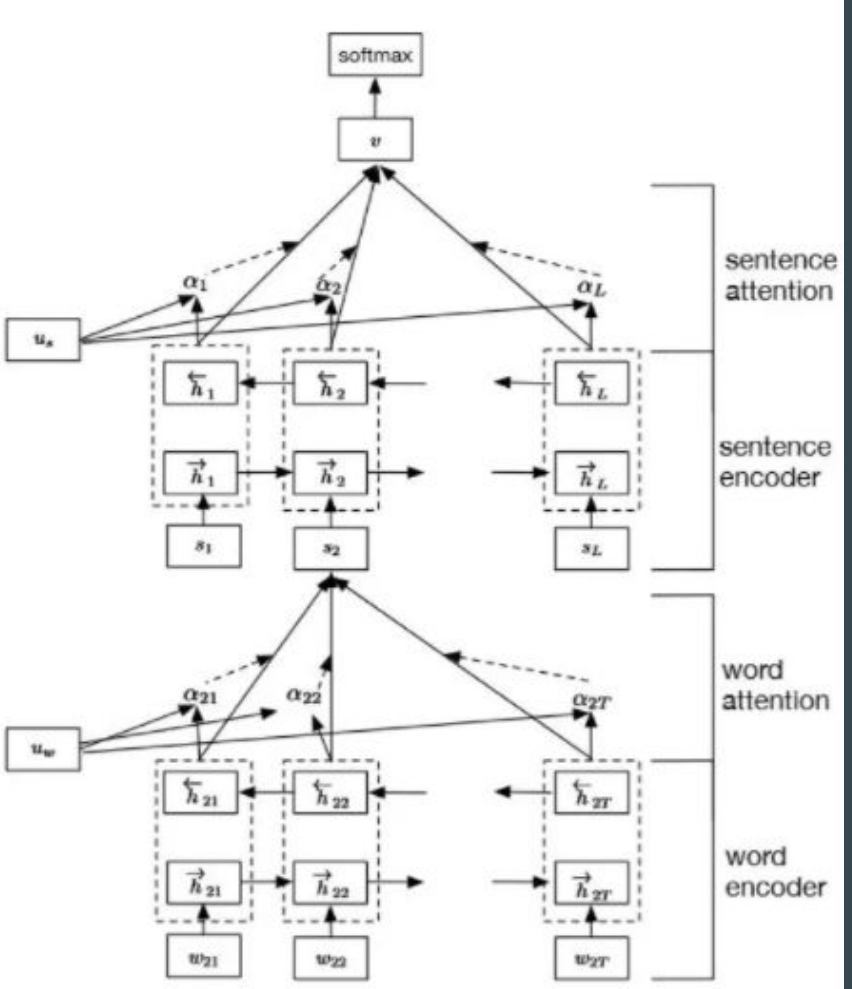

Figure 2: Hierarchical Attention Network.

bidirectional RNN

bi(두개)-RNN 의 의미로 forward RNN, backward RNN으로 구성

- Forward 순차적으로 입력을 읽어서 hidden state 생성
- Backward 역방향으로 입력을 읽어서 hidden state 생성
- 같은 j 번째 hidden state를 연결해서 최종 hidden state 생성
- 가까운 위치에 있는 단어들의 정보를 더 많이 보유하게 됨

Word Encoder

- 단어를 Encoding 하는 구간 (word2vec)

- bidirectional GRU를 이용한 결과를 concat 하여 h를 만든다.

$$
\begin{aligned}
x_{it} &= W_e w_{it}, t \in [1, T], \\
\overrightarrow{h}_{it} &= \overrightarrow{\mathrm{GRU}}(x_{it}), t \in [1, T], \\
\overleftarrow{h}_{it} &= \overleftarrow{\mathrm{GRU}}(x_{it}), t \in [T, 1].
\end{aligned}
$$

Word Attention

- 각 단어 별로 문장의 의도를 파악하는데,
  얼마나 영향을 주는지 파악한다.

- Layer의 결과는 GRU의 각 단어 vector에
  attention layer의 단어별 가중치를 적용해서 더한
  값이다.
- Si 값은 가중치가 표현된 문장을 나타낸다.

$$u_{it} = \tanh(W_w h_{it} + b_w)$$
$$\alpha_{it} = \frac{\exp(u_{it}^\top u_w)}{\sum_t \exp(u_{it}^\top u_w)}$$
$$s_i = \sum_t \alpha_{it} h_{it}.$$

Sentence Encoder

- Word Encoder와 하는 일은 동일하지만,

  input 값이 sentence 이다.

$$\overrightarrow{h}_i = \overrightarrow{\mathrm{GRU}}(s_i), i \in [1, L],$$
$$\overleftarrow{h}_i = \overleftarrow{\mathrm{GRU}}(s_i), t \in [L, 1].$$

Sentence Attention

- 문장 별 가중치를 학습하는 layer를 통과 시킨다.
- 단어별 가중치를 구한다.
- 더한 후, 가중치가 반영된 documentation을 구한다.

$$u_i = \tanh(W_s h_i + b_s),$$
$$\alpha_i = \frac{\exp(u_i^\top u_s)}{\sum_i \exp(u_i^\top u_s)},$$
$$v = \sum_i \alpha_i h_i,$$

Document Classification

- softmax를 통해 클래스 별 가중치를 확률로 변환시킨다.

- Loss 는 Negative Log likelihood를 사용한다.

attention weight가 반영된 good

- b~f는 1점부터 5점까지 별점 리뷰

- 안좋은 경우, 왼쪽으로 쏠려있고,
  좋은 경우, 오른쪽으로 쏠려있다.

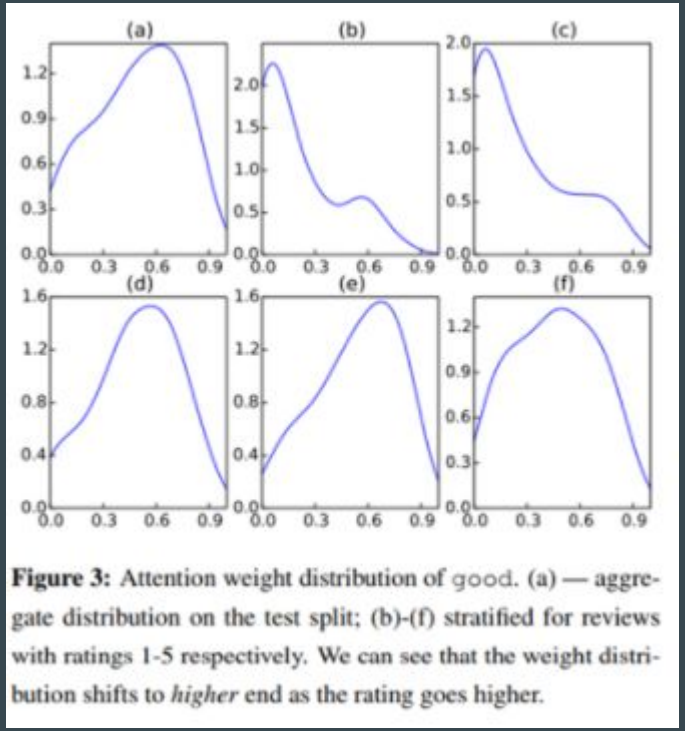

Figure 3: Attention weight distribution of good. (a) — aggregate distribution on the test split; (b)-(f) stratified for reviews with ratings 1-5 respectively. We can see that the weight distribution shifts to *higher* end as the rating goes higher.

attention weight가 반영된 bad

- good과 반대로

  좋은 리뷰의 weight는 적게 나오고,

  나쁜 리뷰의 weight는 크게 나왔다.

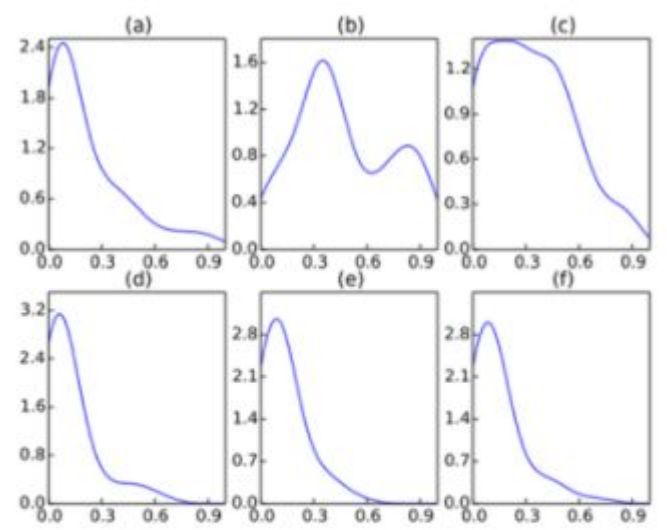

Figure 4: Attention weight distribution of the word `bad`. The setup is as above: (a) contains the aggregate distribution, while (b)-(f) contain stratifications to reviews with ratings 1-5 respectively. Contrary to before, the word `bad` is considered important for poor ratings and less so for good ones.

Thanks for listening