ELMO

ELMo : Embeddings from Language Models 개요

ELMo는 언어모델로, 입력 단어 시퀀스 다움에 어떤 단어가 올지 맞추는 과정에서 학습.

ELMo는 총 3가지로 구성

1. **문자 단위 CNN 레이어** : 각 단어 내 문자들 사이의 의미적, 문법적 관계 도출
2. **양방향 LSTM 레이어** : 단어들 사이의 의미적, 문법적 관계를 도출
3. **ELMo 레이어**

→ 문자단위 CNN레이어와 양방향 LSTM은 ELMo를 프리트레인하는 과정에서 학습

→ ELMo 레이어는 프리트레인이 끝난 후, 구체적인 다운스트림 테스크를 수행하는 과정에서 학습

문자단위 CNN 레이어

- CNN 입력
  example : 밥 —유니코드—> <BOW>, 235, 176, 165, <EOW>, <PAD>, <PAD>
  (사용자가 정한 max_characters_per_token보다 작을 경우 그 차이만큼 스페셜 토큰인 <PAD>에 해당하는 행벡터로 채워줌)
- 문자단위 CNN 레이어
  앞서만든 입력 벡터를 계산해 문자사이의 의미적, 문법적 관계를 도출하고 최종적으로 단어(ex : 밥)의 벡터를 만들어 내는 역활 (한국어 임베딩 책 그림 5-16 참조)

양방향 LSTM 레이어

- LSTM 입력
  문자단위 CNN을 거쳐 만든 풀링벡터를 이어 붙인 뒤 (각 필터마다 풀링 벡터가 다름), 하이웨이 네트워크와 차원 조정하여, 최종적인 단어 벡터를 만듦. (한국어 임베딩 책 그림 5-17참조) → 하이웨이 네트워크와 차원 조정을 양방향 LSTM 레이어 입력적에 사용한

이유는 문자 단위 CNN 레이어를 통과한 단어 임베딩의 차원수가 지나치게 커지면 레이어 학습에 방해가 될 수 있어서로 예상됨.

- 양방향 LSTM, 스코어 레이어
  ELMo 모델이 프리트레인할때는 단어 시퀀스가 주여졌을 때 그 다음 단어가 무엇인지 맞춰야함 → 거대한 말뭉치를 단어 하나씩 슬라이딩해 가면서 그 다음 단어가 무엇인지 맞추는 과정을 반복하다 보면 문장 내에 속한 단어들 사이의 의미적, 문법적 관계들을 ELMo 모델이 이해할 수 있게 됨

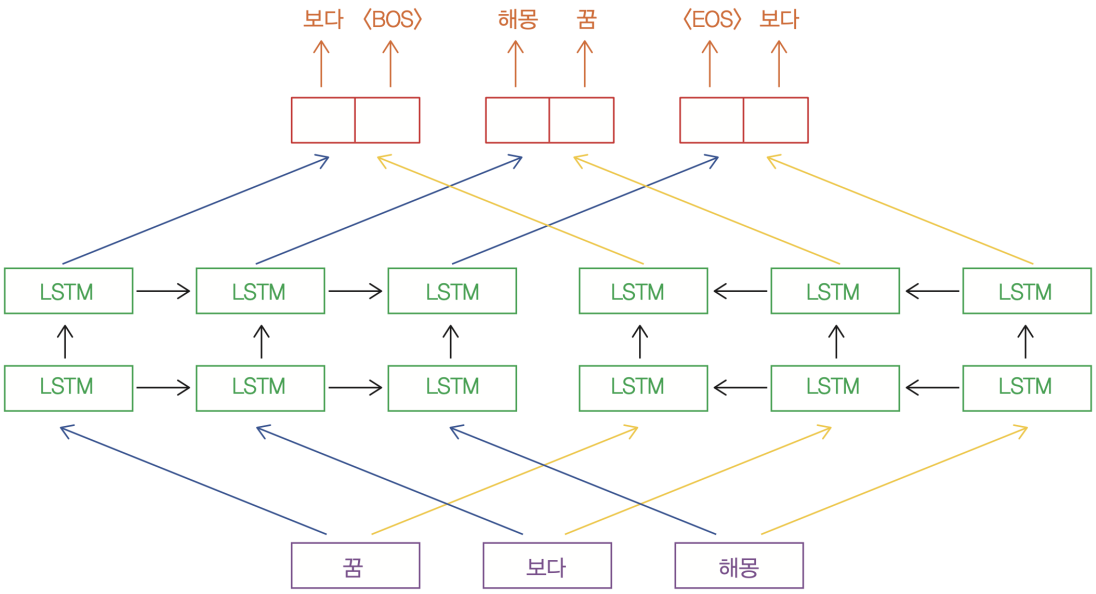

그림 5–18 ELMo 양방향 LSTM 및 출력 레이어

- loss layer
  LSTM 레이어의 최상단 셀의 출력 히든 벡터를 선형변환한 뒤, 소프트맥스를 취함 →이 확률 벡터와 정답 단어에 해당하는 인덱스가 1인 원핫벡터 크로스 엔트로피 계산
- 참조 사항
  1. softmax 확률을 구할때, 일부 단어들만 샘플링해서 구함 (즉, 오답 단어 (네거티브 샘플)를 전체 단어내에서 일부 샘플링하고 이를 정답 단어 (positive sample)고 함께 softmax 확률 계산) ← 전체 어휘집합 갯수만큼 softmax확률을 구하는 것은 비효율적이고, GPU 메모리를 많이 소모 하기에

2. ELMo loss layer에서는 순방향, 양방향 LSTM 출력 히든 벡터를 더하거나 합치지 않고, 각각의 히든 벡터로 각각의 label (순방향 단어 시퀀스, 역방향 단어 시퀀스)를 맞추는 것을 독립적으로 학습함 ← 순방향 네트워커를 학습할 때 뒤쪽에 있는 단어들을 모델에 알려주는 것은 반칙이기에, (loss layer는 프리트레인에는 사용되지만, 파인 튜닝할 때는 사용되지 않음.)

ELMo 레이어

프리트레인이 끝나고, 구체적인 다운 스트림 테스크 학습할 때 사용됨. 입력 문장 k번째 토큰에 대한 ELMo임베딩 수식은 아래와 같다. (한국어 임베딩 그림 5-19 참조)

$$ELMo_k^{task} = r^{task} \sum_{j=0}^{L} s_j^{task} h_{k,j}^{LM}$$

- $h_{k,j}^{LM}$: k번째 토큰의 j번째 레이어의 양방향 LSTM 히든 벡터를 이어 붙인 벡터
- S_j^{task} : j번째 레이어가 해당 테스크 수행에 얼마나 중요한지를 가리키는 스칼라 값.
- r_{task} : 벡터의 크기를 스케일링해 해당 task 수행을 돕는 역활
- L : 양방향 LSTM 레이어의 수, 보통 2로 설정.

출처

- 한국어 임베딩 책