

ELMo (Embeddings from Language Model)

자연어처리 텍스트마이닝

(<https://arxiv.org/pdf/1802.05365.pdf>)

ELMo

- ELMo(Embeddings from Language Model)는 Allen NLP에서 2018년에 발표한 워드 임베딩
- ELMo는 세서미 스트리트라는 미국 인형극의 캐릭터
- 사전 훈련된 언어 모델(Pre-trained language model)을 사용
- 활용
 - 기계 번역 (Machine Translation)
 - 언어 모델링 (Language Modeling)
 - 텍스트 요약 (Text Summarization)
 - 개체명 인식 (NER, Named Entity Recognition)
 - 질문-답변 시스템 (QA, Question - Answering)



ELMo

My watch was a birthday gift

→ 선물

He has a gift for music

→ 재능

- 전통적 단어 임베딩(Word2Vec, FastText, GloVe) 사용시
“gift”는 유일한 벡터로 표현 ⇒ 문맥이 고려되지 않음
- ELMo는 문맥에 따라 임베딩

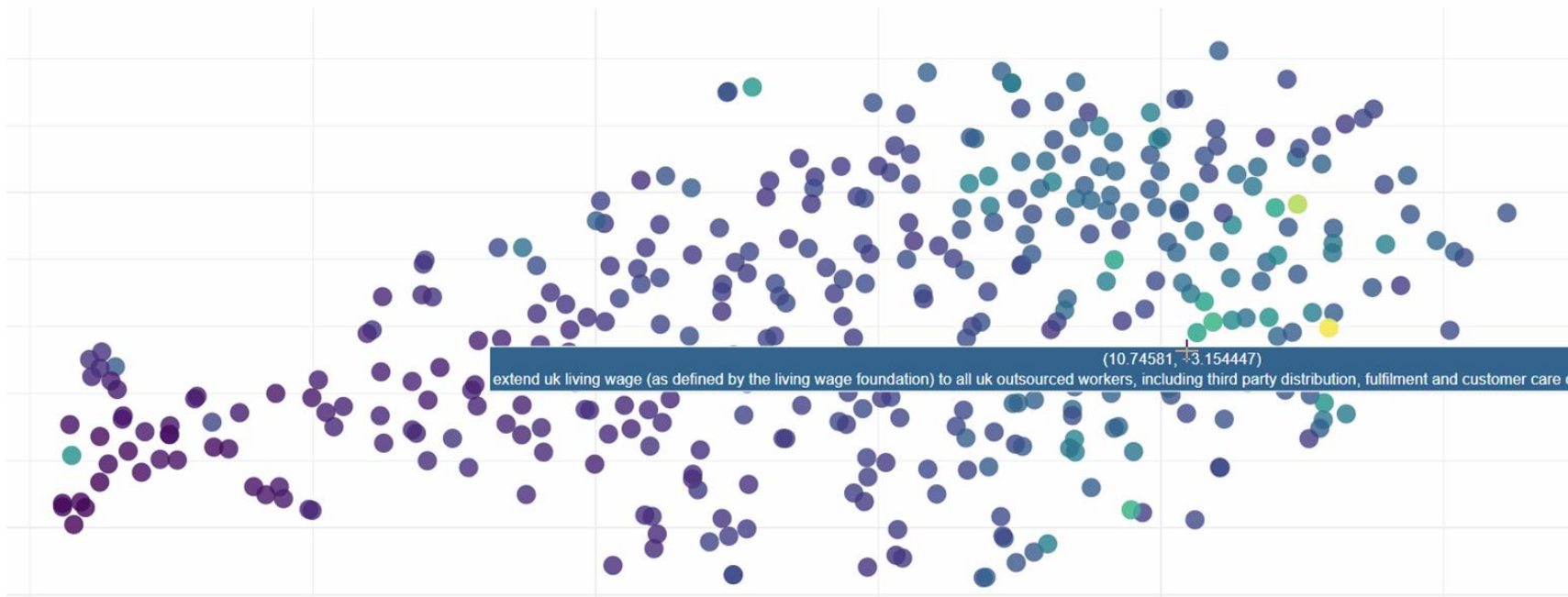


ELMo 성능

2018년 주요 NLP task에서 SOTA를 달성

TASK	PREVIOUS SOTA		OUR BASELINE	ELMo + BASELINE	INCREASE (ABSOLUTE/ RELATIVE)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 \pm 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 \pm 0.19	90.15	92.22 \pm 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 \pm 0.5	3.3 / 6.8%

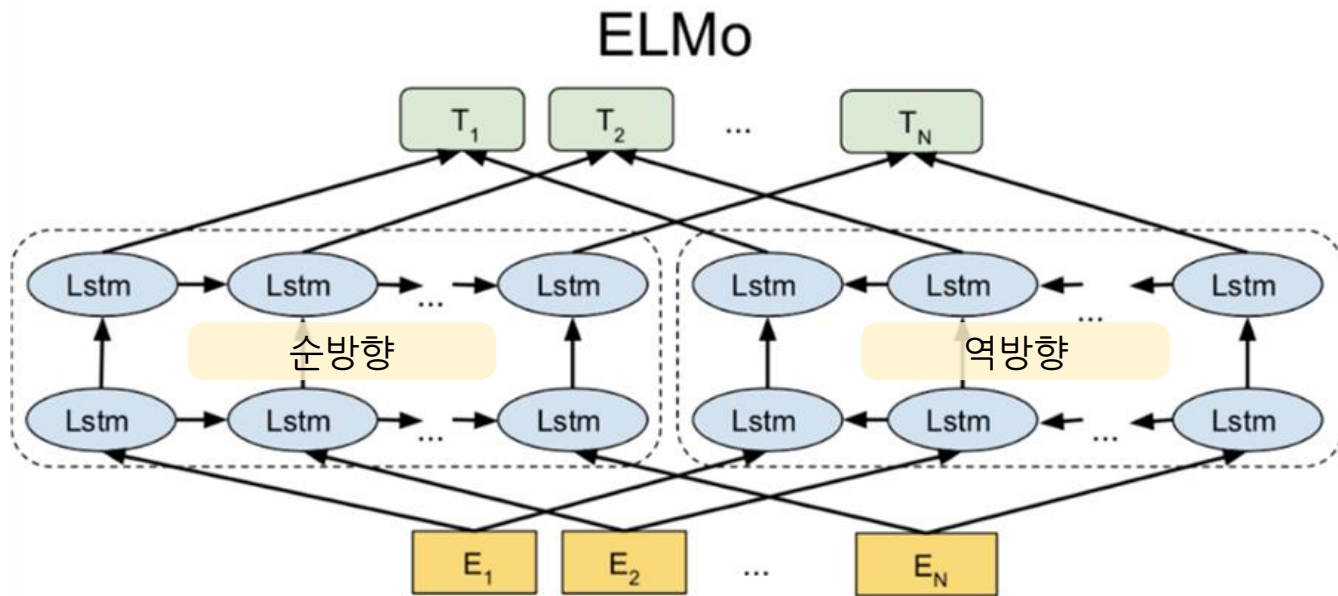
ELMo - 문장 임베딩



<https://towardsdatascience.com/elmo-contextual-language-embedding-335de2268604>

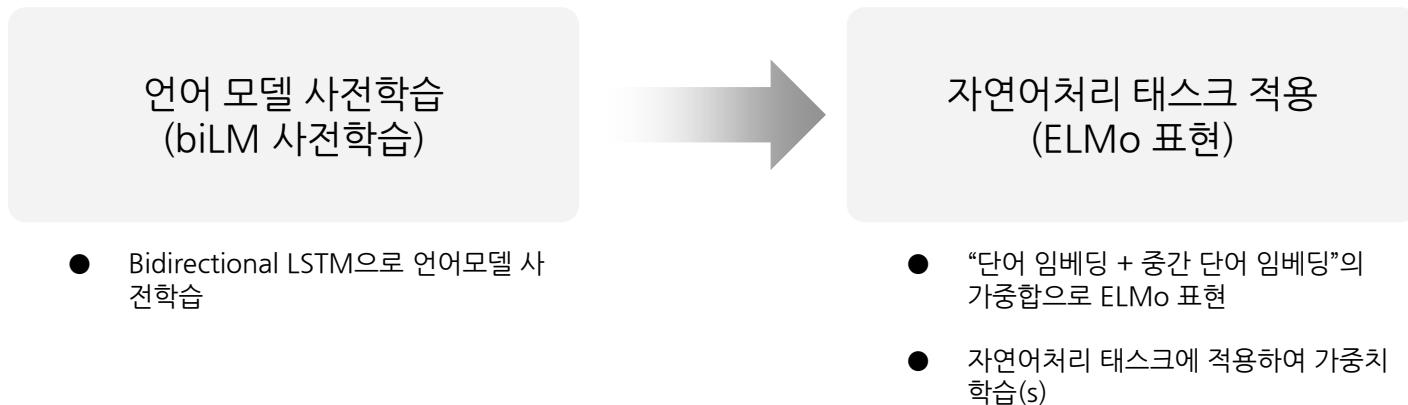
ELMo의 구조

- RNN으로 단어를 예측하는 것은 문맥을 고려한 단어 예측
- ELMo는 순방향 / 역방향으로 예측하는 biLM으로 사전 훈련



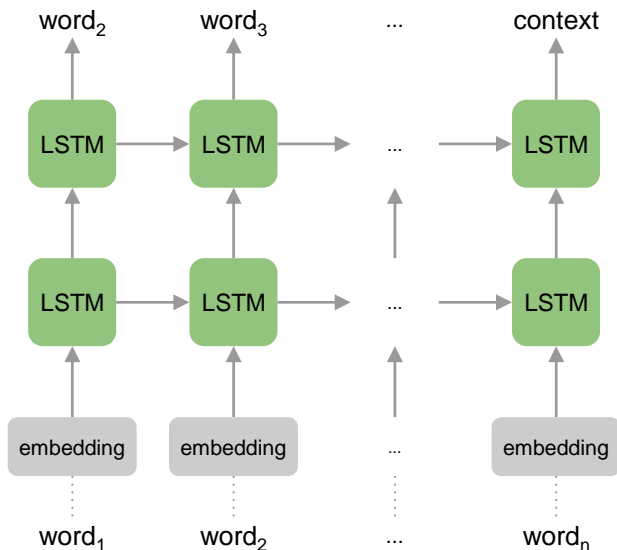
ELMo 사전학습과 활용

- ELMo는 “언어모델 사전학습”과 “자연어처리 Task 적용” 2단계로 수행



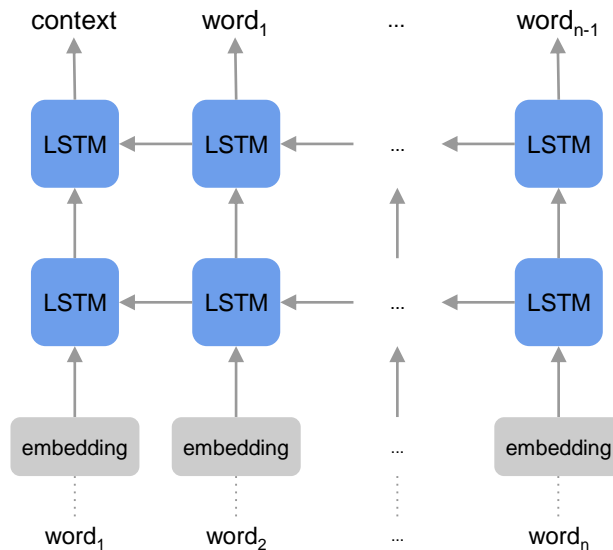
biLM(Bidirectional Language Model)의 사전 훈련

순방향 LSTM



$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_1, t_2, \dots, t_{k-1}).$$

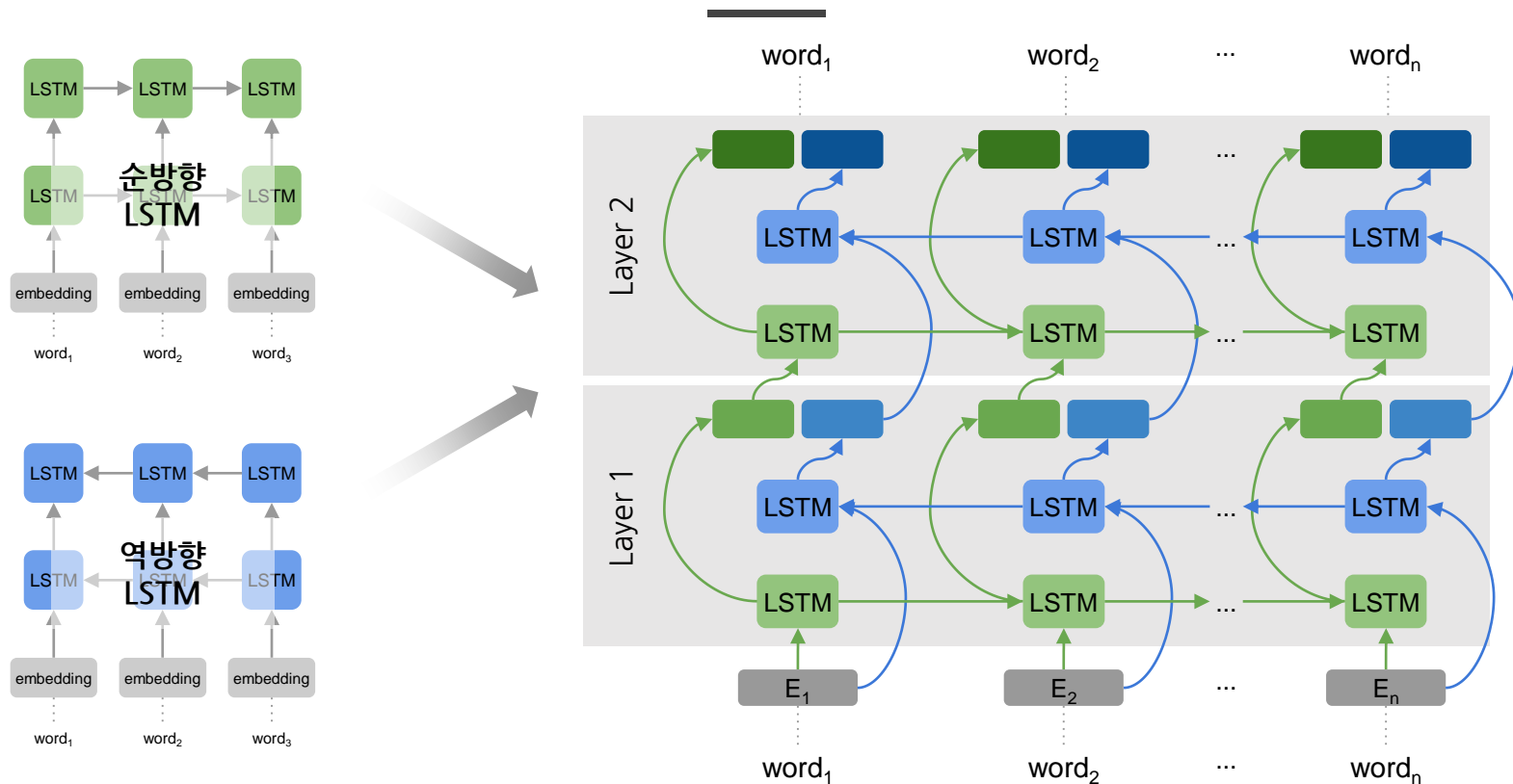
역방향 LSTM



$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_{k+1}, t_{k+2}, \dots, t_N).$$

LSTM의 경우 순방향은 다음단어를 역방향은 이전단어를 예측

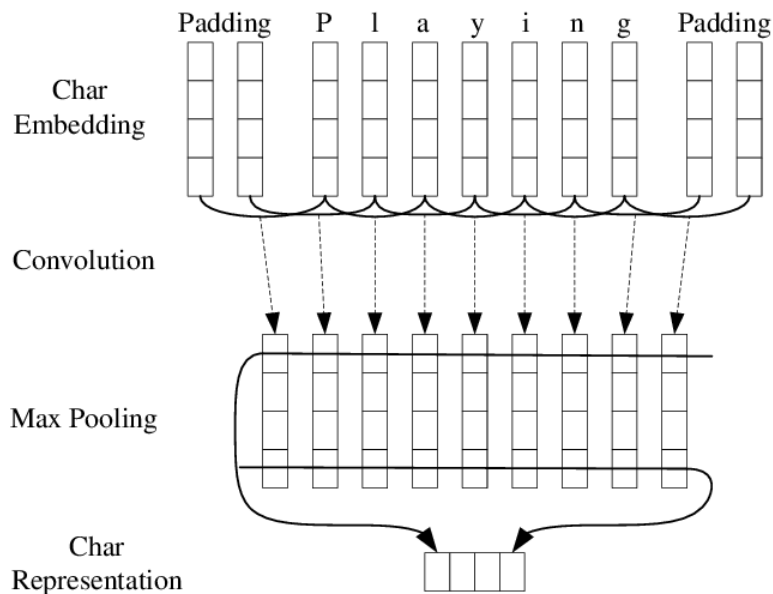
biLM(Bidirectional Language Model)의 사전 훈련



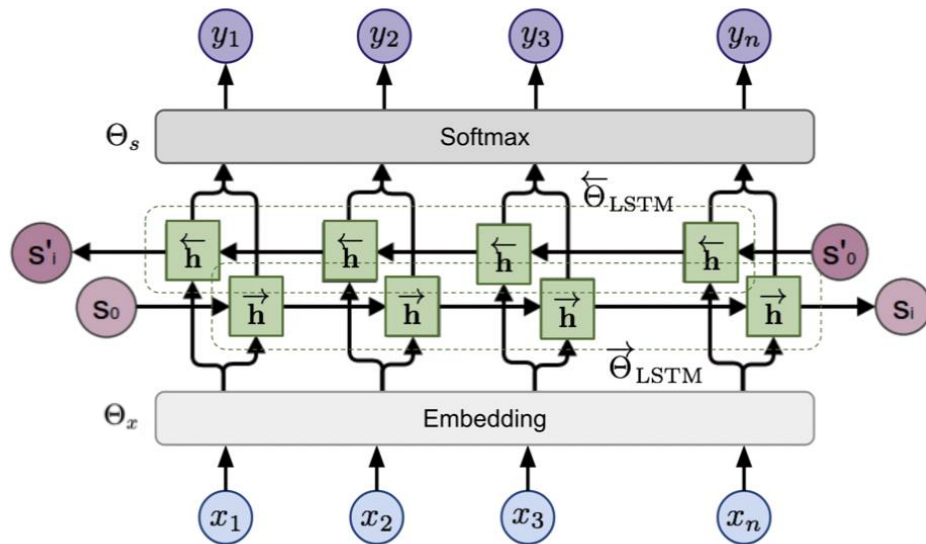
양방향의 컨텍스트를 포함하여 단어를 예측하여 학습

char CNN

- ELMo 입력 단어의 임베딩 방법으로 char CNN를 사용
- 글자(character) 단위로 CNN을 적용하여 임베딩 함으로써, 문맥과 상관없이 단어안에 포함된 서브단어(sub word)도 임베딩 가능
- OOV(Out of Vocabulary)에 견고
- 글자단위로 함으로써 한국어, 일본어에도 적용가능



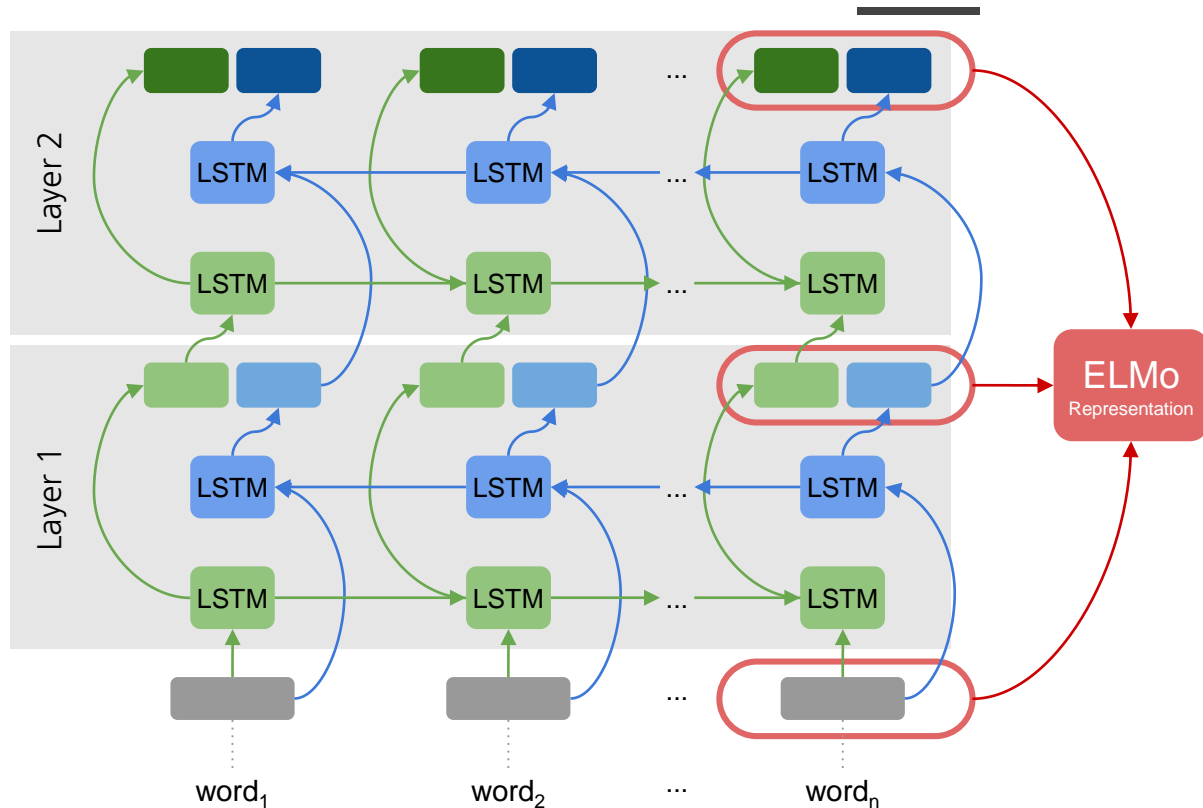
양방향 LSTM 학습



$$\sum_{k=1}^N \log P(t_k \mid t_1, \dots, t_{k-1}; \Theta_x, \Theta_{LSTM}^{\rightarrow}, \Theta_s) \\ + \log P(t_k \mid t_{k+1}, \dots, t_N; \Theta_x, \Theta_{LSTM}^{\leftarrow}, \Theta_s)$$

N	문장내 단어 개수
t_k	k 번째 단어
Θ_x	입력단어 임베딩
Θ_{LSTM}	LSTM 레이어 (순방향, 역방향)
Θ_s	Softmax 레이어

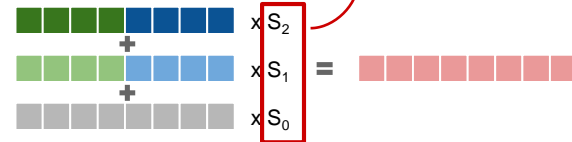
ELMo Representation



1단계 : 레이어 출력값을 연결

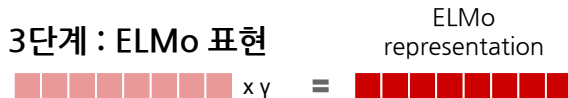


2단계 : 출력값 가중 합



downstream
task에 맞게 학습

3단계 : ELMo 표현



ELMo Representation

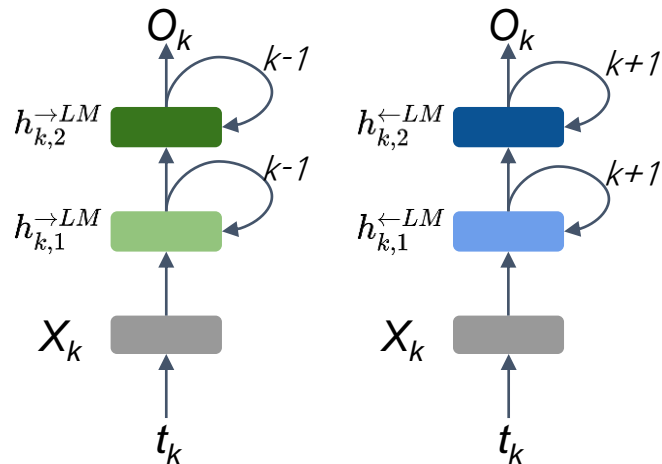
$$\text{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^L s_j^{task} h_{k,j}^{LM}$$

$$\text{ELMo}_k^{task} = \gamma^{task} \cdot \sum \left\{ \begin{array}{l} s_2^{task} \times h_{k,2}^{LM} \\ s_1^{task} \times h_{k,1}^{LM} \\ s_0^{task} \times h_{k,0}^{LM} \end{array} \right\} [X_k; X_k]$$

양방향 언어모델 (biLM)

순방향 언어모델

역방향 언어모델



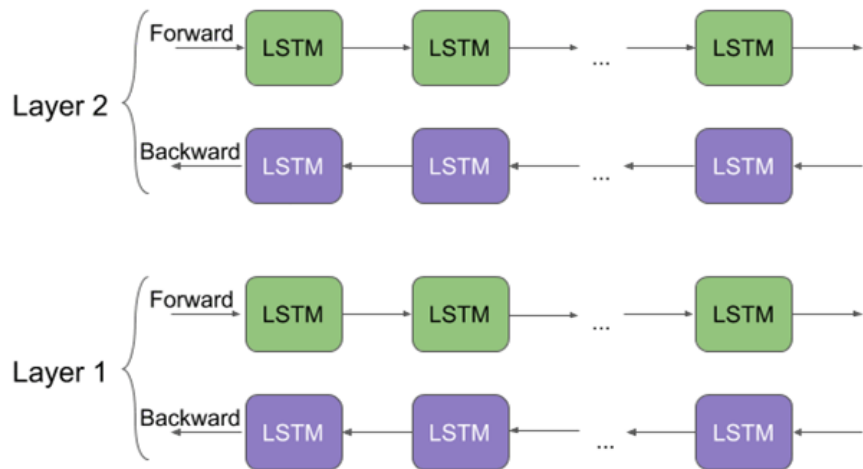
ELMo Representation

$$\sum_{k=1}^N \log P(t_k \mid t_1, \dots, t_{k-1}; \Theta_x, \Theta_{LSTM}^{\rightarrow}, \Theta_s) \\ + \log P(t_k \mid t_{k+1}, \dots, t_N; \Theta_x, \Theta_{LSTM}^{\leftarrow}, \Theta_s)$$

$$R_k = \{x_k^{LM}, h_{k,j}^{\rightarrow LM}, h_{k,j}^{\leftarrow LM} \mid j = 1, \dots, L\} \\ = \{h_{k,j}^{LM} \mid j = 1, \dots, L\}$$

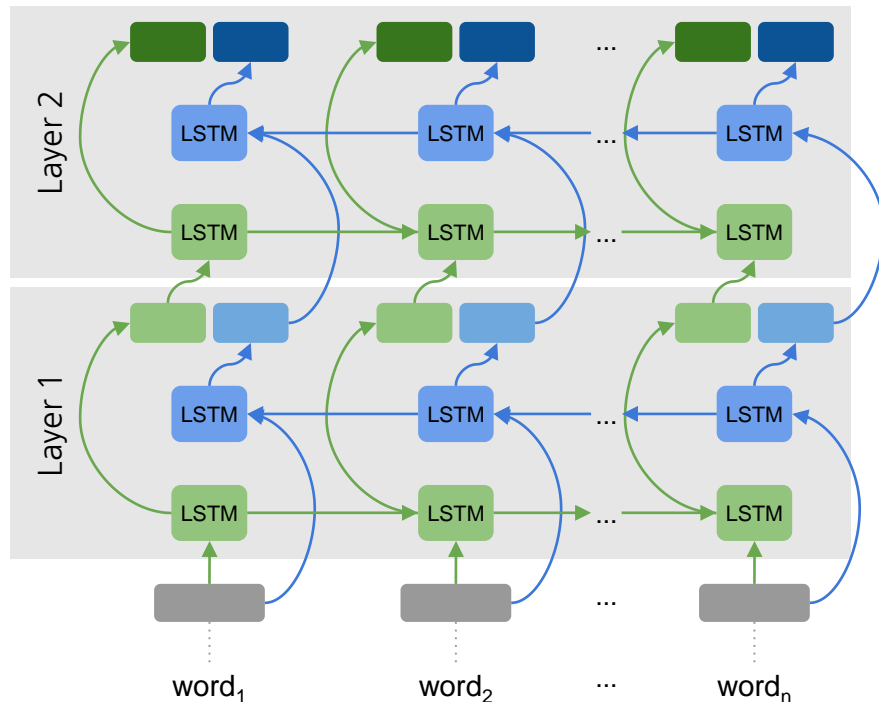
$$\text{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^L s_j^{task} h_{k,j}^{LM}$$

ELMo Representation

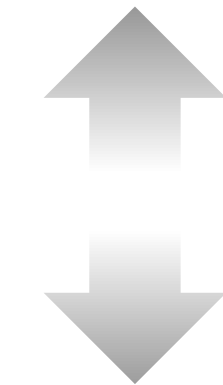


- Char CNN으로 단어 벡터 입력
- 순방향(Forward)는 특정 단어와 이전 단어들의 컨텍스트가 포함
- 역방향(Backward)은 특정 단어와 이후 단어들의 컨텍스트가 포함
- 순방향 및 역방향으로부터 정보 쌓은 중간 단어 벡터를 생성
- 중간 단어 벡터는 다음 레이어의 입력으로 사용
- 최종 단어 벡터는 “Char CNN 단어벡터”와 2개 “중단 단어벡터”의 가중 합

“Stacking” 의 효과



의미적(Semantic)



문법적(Syntactic)

단어의미 모호성
(Word sense disambiguation)

Model	F ₁
WordNet 1st Sense Baseline	65.9
Raganato et al. (2017a)	69.9
Iacobacci et al. (2016)	70.1
CoVe, First Layer	59.4
CoVe, Second Layer	64.7
biLM, First layer	67.4
biLM, Second layer	69.0

품사 태깅
(PoS Tagging)

Model	Acc.
Collobert et al. (2011)	97.3
Ma and Hovy (2016)	97.6
Ling et al. (2015)	97.8
CoVe, First Layer	93.3
CoVe, Second Layer	92.8
biLM, First Layer	97.3
biLM, Second Layer	96.8

레이어별 중간 벡터를 반영하여 단어의 의미적(Semantic), 문법적(Syntactic) 부분을 함께 고려

NLP Task 적용

ELMo 표현을 NLP Task에 적용할 때는 “기존 임베딩 + ELMo표현”을 입력값으로 사용하여 적용

