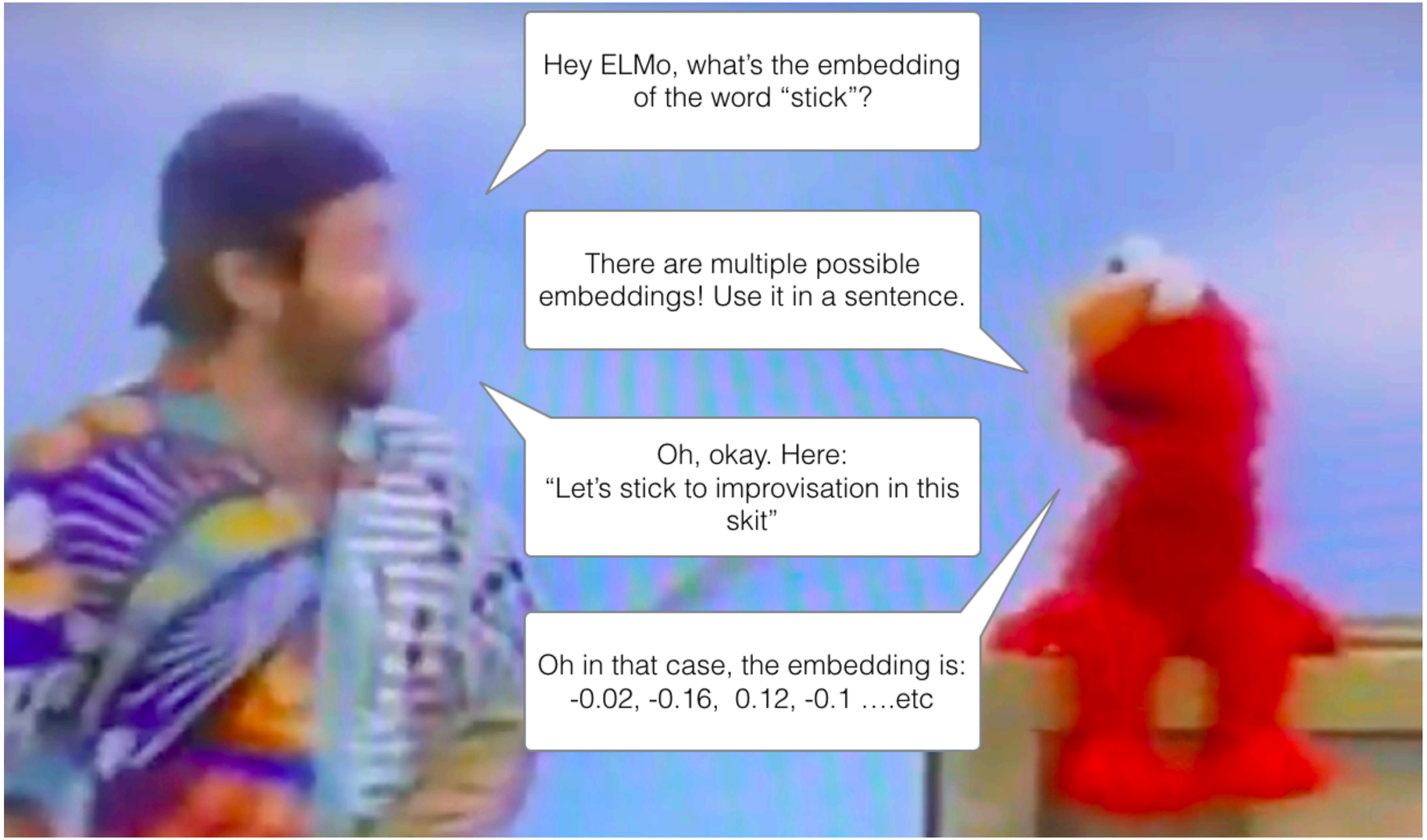
NLP 논문 리뷰

ELMo - Deep contextualized word representations

ELMo : Embeddings from Language Models



ELMo : Embeddings from Language Models

- Pre-trained word representations

  A key component in many neural language understanding models

  Pre-trained 된 단어 representations 은 많은 자연어 이해 모델에서 가장 중요한 구성요소입니다.

- High quality representations

  1) complex characteristics of word use (syntax and semantics)

     단어 사용의 복잡한 특성(문법과 의미)

  2) how these uses vary across linguistic contexts (to model polysemy)

     단어들이 언어적 문맥(동의어)에 따라 어떻게 사용되는지

ELMo : Embeddings from Language Models

- Differ from traditional word type embeddings

  Each token is assigned a representation that is a function of the entire input sentence.
  각각의 토큰들은 전체 입력 문장의 함수인 representation 을 할당 받습니다.

  Use vector derived from a bidirectional LSTM that is trained with a coupled language model (LM) objective on a large text corpus.
  많은 양의 말뭉치로에서 두개의 언어모델 목적을 가지고 학습된 bi-LSTM으로부터 얻어진 벡터를 사용합니다.

- ELMo representations are deep, in the sense that they are a function of all of the internal layers of the biLM.

  a linear combination of the vectors stacked above each input word for each end task, which markedly improves performance over just using the top LSTM layer.
  각각의 입력 단어들 위에 쌓여진 end task를 위한 벡터들의 선형결합을 통해 가장 top LSTM 레이어를 사용한 것보다 눈에 띄는 성능향상을 이루어냈습니다.

  - higher-level LSTM states capture context-dependent aspects of word meaning
    높은 레벨의 LSTM 은 단어 의미의 문맥의존적 정보를 포착하고
  - Lower-level states model aspects of syntax
    낮은 레벨의 LSTM 은 구문, 문법적인 정보를 포착했습니다.

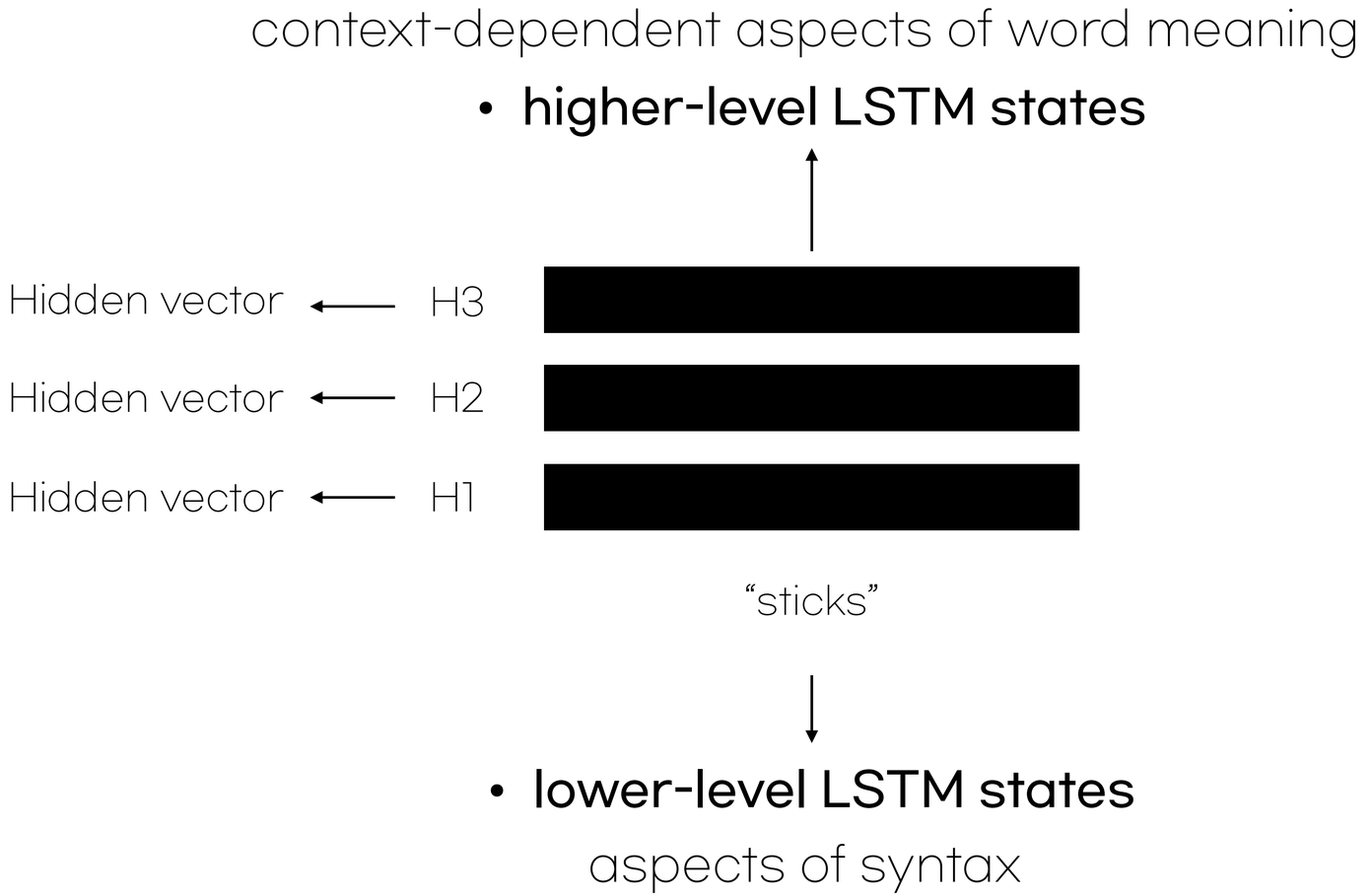
ELMo : Embeddings from Language Models

h_1, h_2, h_3 에서 학습된 특정한 벡터들

+

선형결합

"sticks" 라는 단어의 위치에 있는
여러 layer 에 있는 정보들을 결합



ELMo : Embeddings from Language Models

- Related work

  Pretained word vectors are a standard component of most state-of-the-art NLP architectures

  However these approaches for learning word vectors only allow a single context-independent representation for each word

  Enriching with subword information, separate vectors for each word

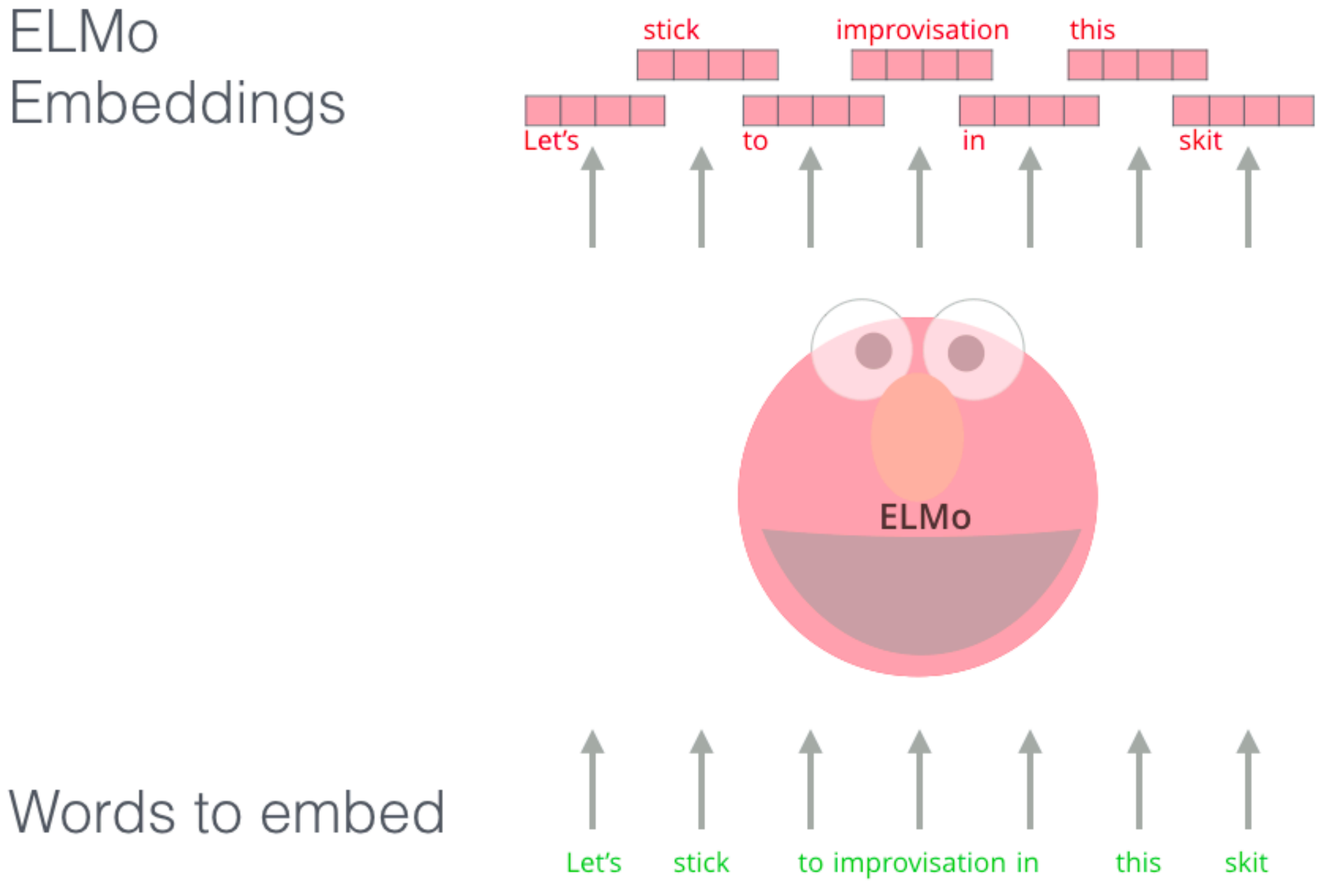
  - context-independent representation

    bi-LSTM, context2vec (Melamud et al., 2016)

    Pivot word, CoVe (McCann et al., 2017)

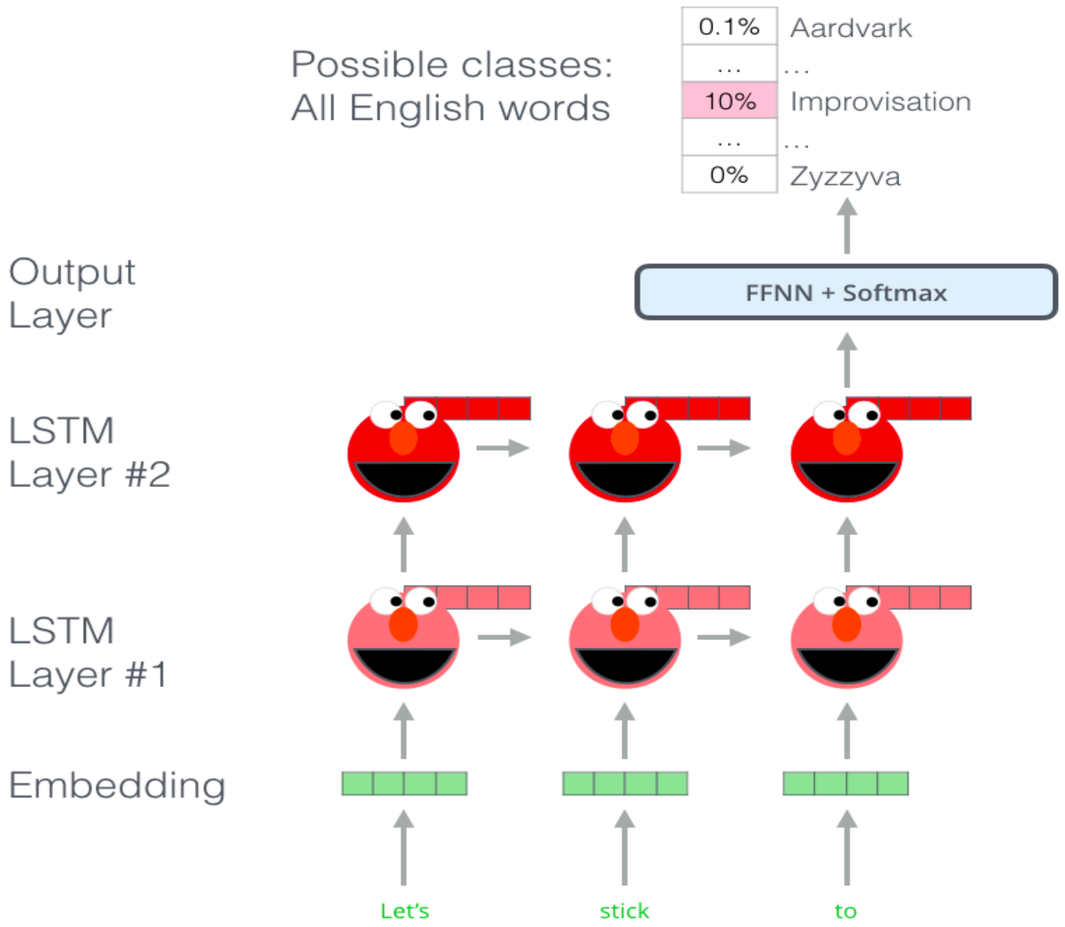
The addition of ELMo representations alone significantly improves the state of the art in every case, including up to 20% relative error reduction

ELMo : Embeddings from Language Models



https://jalammar.github.io/illustrated-bert/

ELMo : Embeddings from Language Models

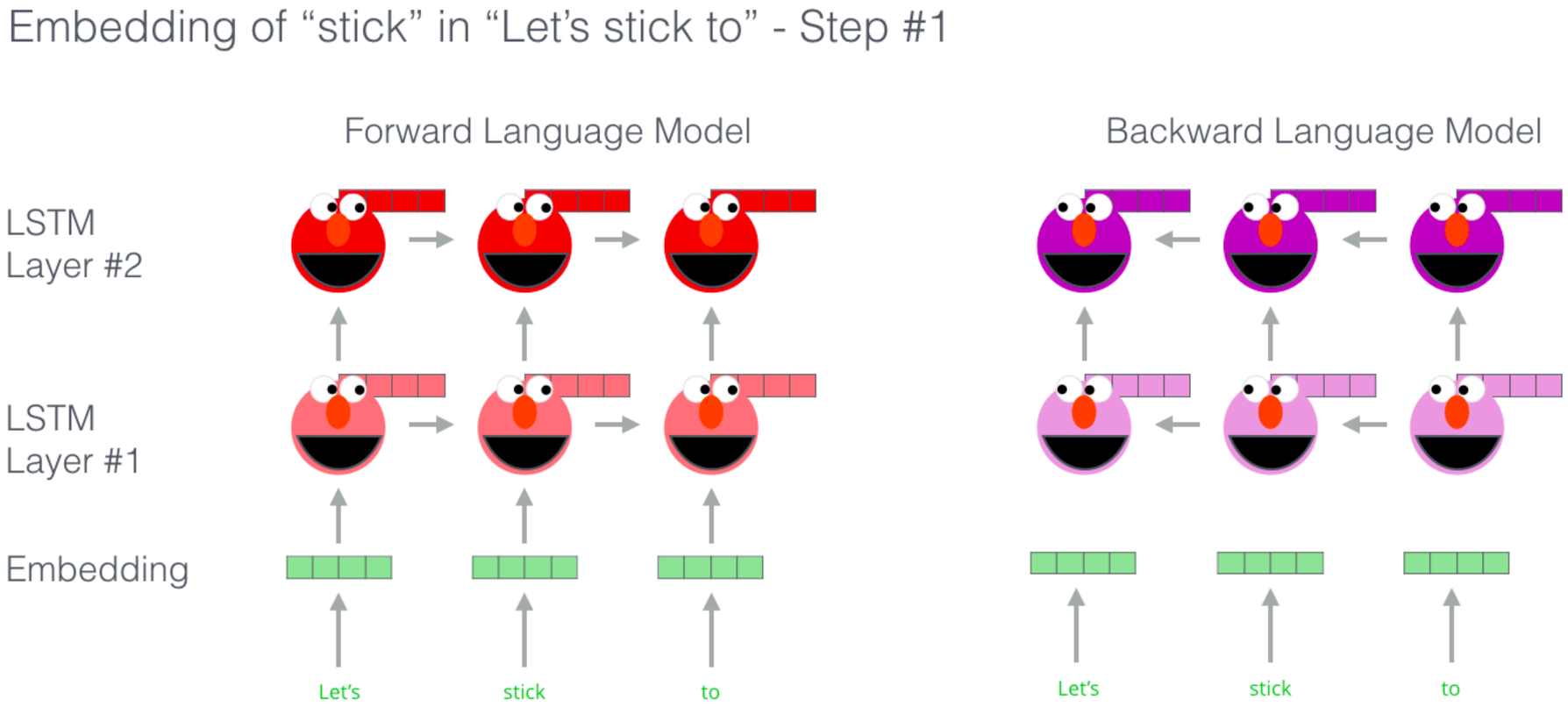


ELMo gained its language understanding from being trained to predict the next word in a sequence of words - a task called Language Modeling.

This is convenient because we have vast amounts of text data that such a model can learn from without needing labels.

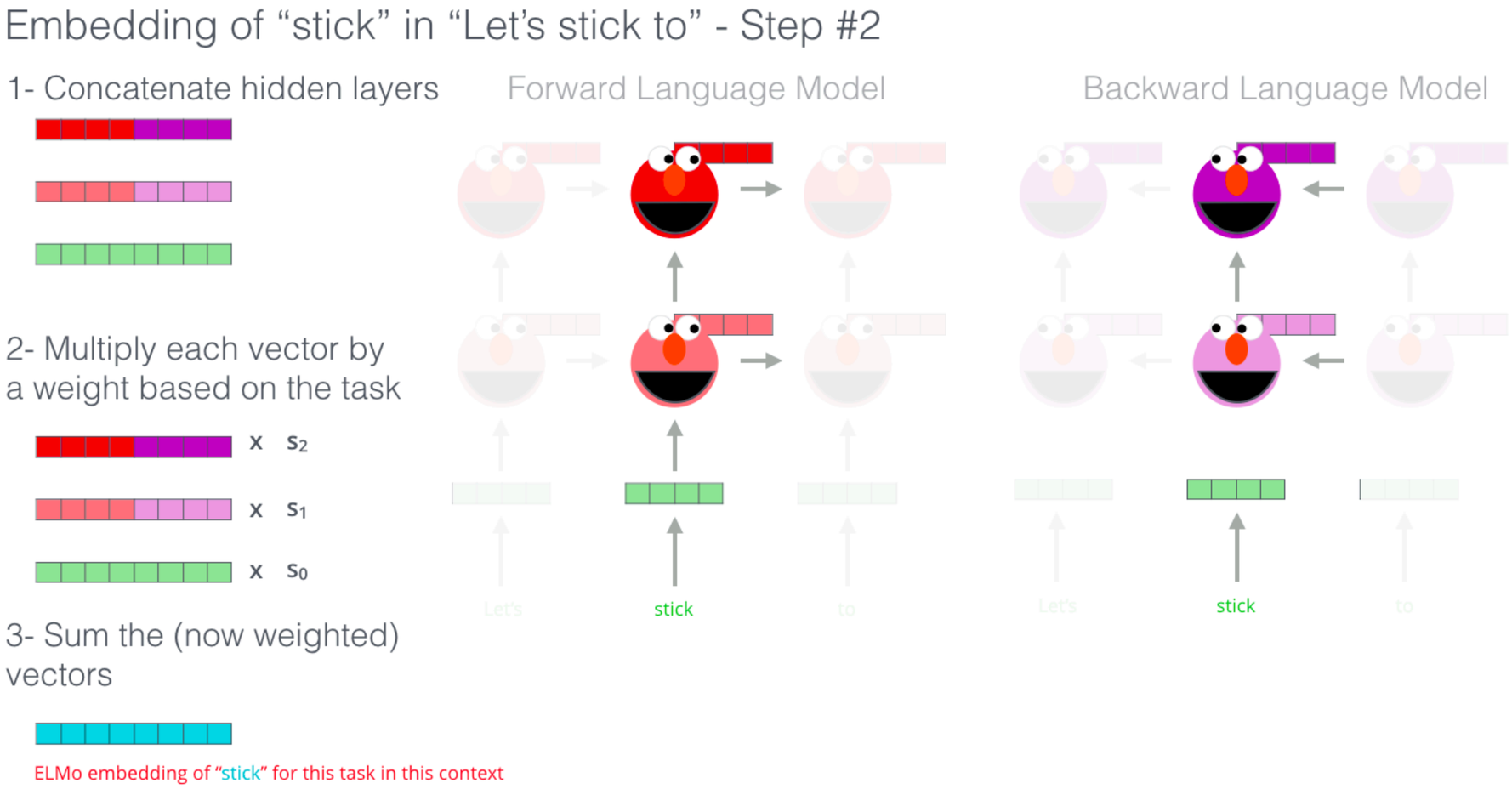
ELMo : Embeddings from Language Models

ELMo actually goes a step further and trains a bi-directional LSTM - so that its language model doesn't only have a sense of the next word, but also the previous word.



ELMo : Embeddings from Language Models

ELMo comes up with the contextualized embedding through grouping together the hidden states (and initial embedding) in a certain way (concatenation followed by weighted summation).



ELMo : Embeddings from Language Models

ELMo represents a word t_k as a linear combination of corresponding hidden layers (inc. its embedding)

ELMo is a task specific representation. A down-stream task learns weighting parameters

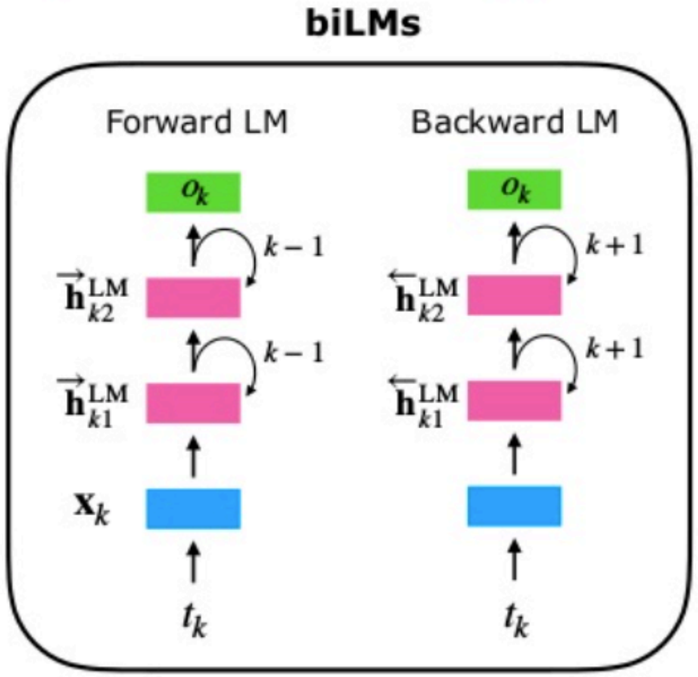
$$\mathbf{ELMo}_k^{\text{task}} = \gamma^{\text{task}} \times \sum \left\{ s_2^{\text{task}} \times \mathbf{h}_{k2}^{\text{LM}},\ s_1^{\text{task}} \times \mathbf{h}_{k1}^{\text{LM}},\ s_0^{\text{task}} \times \mathbf{h}_{k0}^{\text{LM}} \right.$$

$$\left. ([\mathbf{x}_k;\ \mathbf{x}_k]) \right.$$

Concatenate hidden layers

$$[\overrightarrow{\mathbf{h}}_{kj}^{\text{LM}};\ \overleftarrow{\mathbf{h}}_{kj}^{\text{LM}}]$$

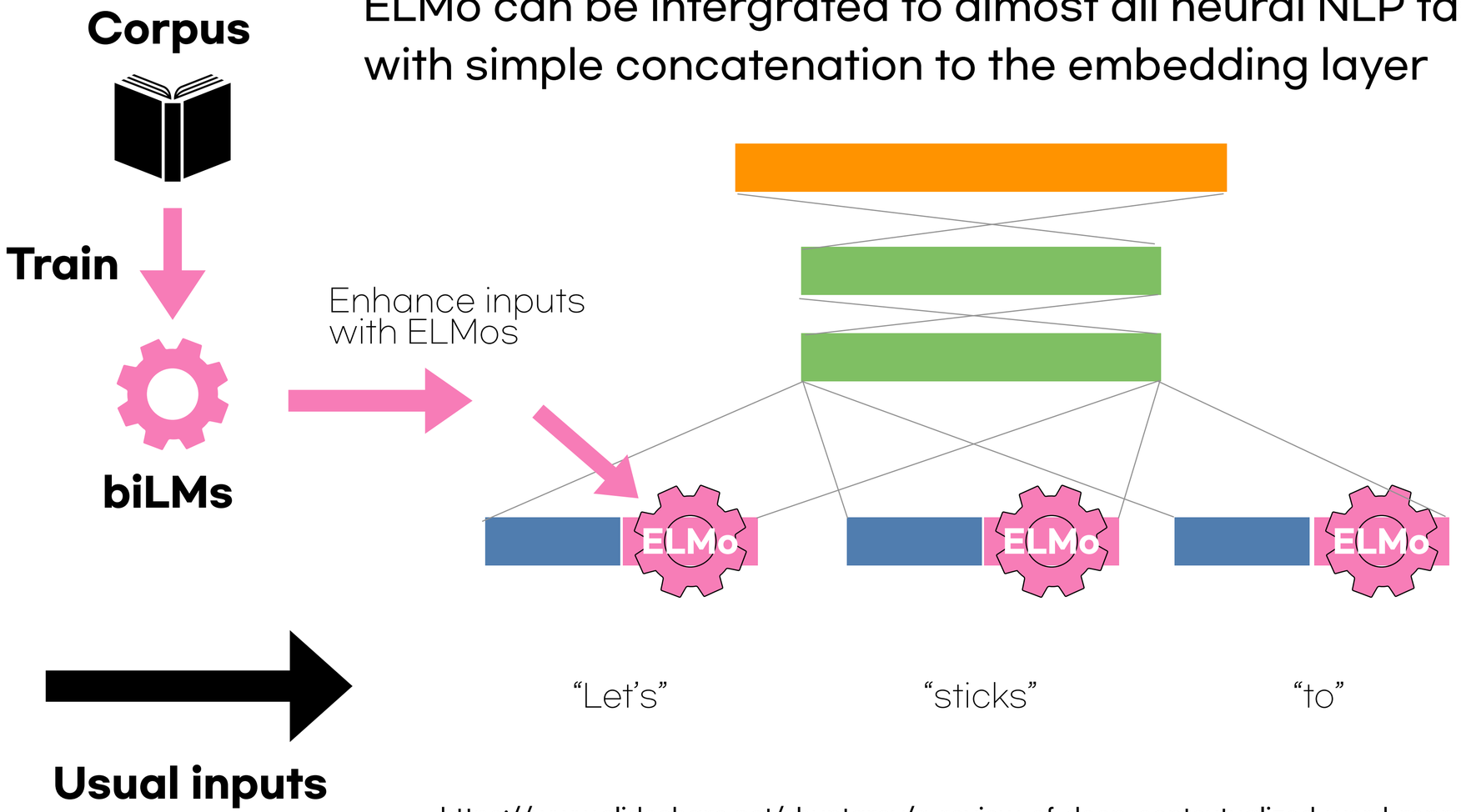
Unlike usual word embeddings, ELMo is assigned to every *token* instead of a *type*



https://www.slideshare.net/shuntaroy/a-review-of-deep-contextualized-word-representations-peters-2018

ELMo : Embeddings from Language Models

ELMo can be intergrated to almost all neural NLP tasks
with simple concatenation to the embedding layer



https://www.slideshare.net/shuntaroy/a-review-of-deep-contextualized-word-representations-peters-2018

ELMo : Embeddings from Language Models

- Bidirectional language models

  Given a sequence of N tokens, $(t_1, t_2, \ldots, t_N)$, a forward language model computes the probability of the sequence by modeling the probability of token t_k given the history $(t_1, \ldots, t_{k-1})$

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

ELMo : Embeddings from Language Models

확률분포(probability distribution)는 확률변수가 확률함수(probability function)에 의해 사이의 값으로 매핑될 때 확률변수의 모든 값과 그에 대응하는 확률들이 어떻게 분포하고 있는지를 말한다

결합확률분포(Joint probability distribution)는 두 개 이상의 사건이 동시에 일어날 확률에 대한 분포를 말한다.

ELMo : Embeddings from Language Models

Neural language models compute a context-independent token representation X_k^{LM} (via token embeddings or a CNN over characters) the pass it through L-layers of forward LSTMs.

At each position k, each LSTM layer outputs a context-dependent representation $\overrightarrow{h}_{k\,j}^{LM}$ (where $i = 1,\ldots,L$).

The top layer LSTM output, $\overrightarrow{h}_{k,L}^{LM}$, is used to predict the next token t_{k+1} with a Softmax layer.

ELMo : Embeddings from Language Models

- Bidirectional language models

A backward LM is similar to forward LM, except it runs over the sequence in reverse, predicting the previous token given the future context :

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

Each backward LSTM layer j in a L layer deep model producing representations $\overleftarrow{h}_{k,j}^{LM}$ of t_k given $(t_{k+1}, \ldots, t_N)$.

ELMo : Embeddings from Language Models

- Bidirectional language models

  Jointly maximizer the log likelihood of the forward and backward directions:

$$\sum_{k=1}^{N} (\log p(t_k \mid t_1, \ldots, t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) + \log p(t_k \mid t_1, \ldots, t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s))$$

  - Θ_x, Θ_s : token representation , softmax layer parameters
  - Maintaining separate parameters for the LSTMs in each directions.

ELMo : Embeddings from Language Models

- ELMo

Each token t_k, a L-layer biSM computes a set of $2L + 1$ representations

$$R_k = \left\{ x_k^{LM}, \overrightarrow{h}_{l,j}^{LM}, \overleftarrow{h}_{l,j}^{LM} \mid j = 1,\ldots,L \right\} = \left\{ h_{k,j}^{LM} \mid j = 0,\ldots,L \right\}$$

- Where $h_{k,0}^{LM}$ is the token layer and $h_{k,j}^{LM} = [\overrightarrow{h}_{k,j}^{LM}; \overleftarrow{h}_{k,j}^{LM}]$ for each biLSTM layer

For inclusion in a downstream model, ELMo collapses all layers in R into a single vector

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

Hidden state vectors of each layers

allows the tasks model to scale the entire ELMo scale

Softmax-normalized weights

ELMo : Embeddings from Language Models

- Using biLMs for supervised NLP tasks
  - Run the biLM and record all of the layer representations for each word
  - Let the end task model learn a linear combination
    1. consider the lowest layers without biLM
    2. Most NLP model (supervised) share a common architecture at the lowest layers allowing us to add ELMo in a consistent, unified manner.
    3. Given a sequence of tokens $(t_1, \ldots, t_N)$, it is standard to form a context-independent token representation x_k for each token position using pre-trained word embeddings(optionally character-based representations)
    4. Then model forms a context-sensitive representation h_k, typically using either bidirectional RNNs, CNNs, or feed forward networks.

ELMo : Embeddings from Language Models

- Using biLMs for supervised NLP tasks

  - To add ELMo to the supervised model,
    1. Freeze the weights of the biLM
    2. Concatenate the ELMo vector $ELMo_k^{task}$ with x_k
    3. and pass the ELMo enhanced representation $[x_k; ELMo_k^{task}]$ into task RNN

  - To add ELMo to the supervised model,

Evaluation

Performance of ELMo across a diverse set of six benchmark NLP tasks

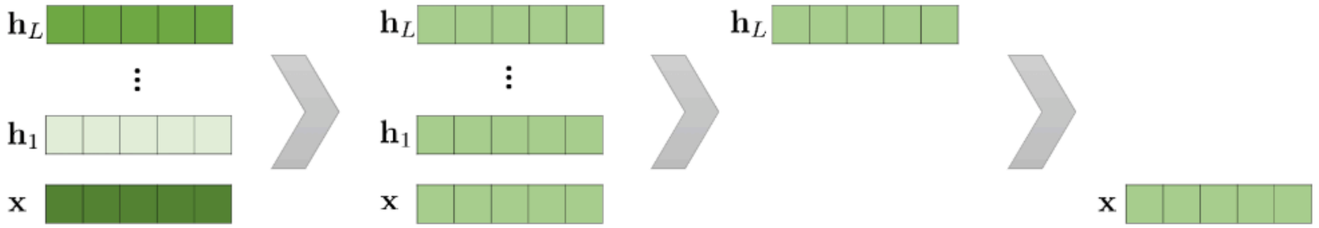
Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

Question Answering - SQuAD(Stanford Question Answering Dataset)
Textual Entailment - SNLI(Stanford Natural Language Inference)
Semantic role labeling - SRL(Semantic Role Labeling)
Coreference resolution - Coref(Coreference resolution)
Named entity extraction - NER(Named Entity Recognition)
Sentiment analysis - SST-5(Stanford Sentiment Tree-bank)

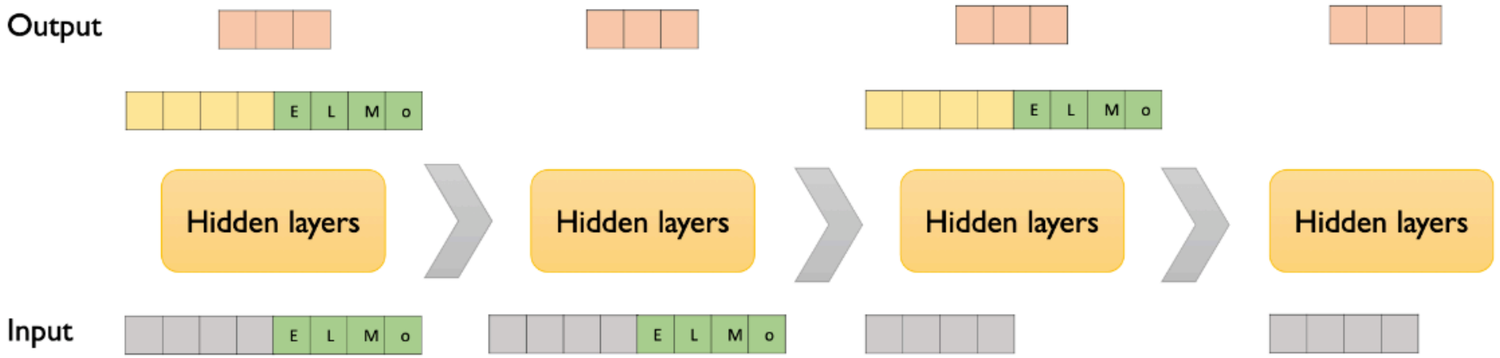
Analysis

Ablation analysis to validate our chief claims and to elucidate some interesting aspects of ELMo representations.

- Alternate layer weighting schemes



- Where to include ELMo?



https://github.com/pilsung-kang/Text-Analytics/blob/master/08%20Seq2Seq%20Learning%20and%20Pre-trained%20Models/08-3_ELMo.pdf

Analysis

- What information is captured by the biLM's representations?
  - The biLM is able to disambiguate both part of speech and word sense in the source sentence

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular play on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play .
	Olivia De Havilland signed to do a Broadway play for Garson {...}	{...} they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement .

Table 4: Nearest neighbors to "play" using GloVe and the context embeddings from a biLM.

스포츠, 놀이와 관련된 이웃벡터

Source 문장과 비슷한 의미로 쓰인 문장이 가깝게 위치하고 있음.

Analysis

- 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*

Table 5: All-words fine grained WSD F_1. For CoVe and the biLM, we report scores for both the first and second layer biLSTMs.

- 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

Table 6: Test set POS tagging accuracies for PTB. For CoVe and the biLM, we report scores for both the first and second layer biLSTMs.

these experiments confirm different layers in the biLM represent different types of information and explain why including all biLM layers is important for the highest performance in down-stream tasks.

Thanks

일러스트 출처

- https://www.slideshare.net/shuntaroy
- https://github.com/pilsung-kang/Text-Analytics/blob/master/08%20Seq2Seq%20Learning%20and%20Pre-trained%20Models/08-3_ELMo.pdf
- https://jalammar.github.io/illustrated-bert/

참고

- https://arxiv.org/abs/1802.05365
- https://youtu.be/zV8klUwH32M
- https://greeksharifa.github.io/