Deep contextualized word representations

2018.12.22
김보섭

Agenda

1. Abstract
2. Introduction
3. ELMo: Embeddings from Language Models
4. Evaluation
5. Analysis
6. Conclusion

Abstract

본 논문에서는 pre-trained deep bidirectional language model을 이용, context, syntactic, semantic 등을 고려하여 word를 representation하는 방식을 제시

Deep contextualized word representations

Matthew E. Peters[†], **Mark Neumann**[†], **Mohit Iyyer**[†], **Matt Gardner**[†],
{matthewp,markn,mohiti,mattg}@allenai.org

Christopher Clark[*], **Kenton Lee**[*], **Luke Zettlemoyer**[†*]
{csquared,kentonl,lsz}@cs.washington.edu

[†]Allen Institute for Artificial Intelligence
[*]Paul G. Allen School of Computer Science & Engineering, University of Washington

Abstract

We introduce a new type of *deep contextualized* word representation that models both (1) complex characteristics of word use (e.g., syntax and semantics), and (2) how these uses vary across linguistic contexts (i.e., to model polysemy). Our word vectors are learned functions of the internal states of a deep bidirectional language model (biLM), which is pre-trained on a large text corpus. We show that these representations can be easily added to existing models and significantly improve the state of the art across six challenging NLP problems, including question answering, textual entailment and sentiment analysis. We also present an analysis showing that exposing the deep internals of the pre-trained network is crucial, allowing downstream models to mix different types of semi-supervision signals.

Introduction (1/3)

기존의 word representation (eg. skip-gram, sisg, etc.)은 context-independent representation으로 아래의 두 가지 잘 modeling 하기가 어려움

High quality word representations?

- modeling complex characteristics of word use (eg. syntax and semantics)

→ subword information skip-gram (sisg, aka "FastText")가 특히 syntax를 modeling하는 측면에서 아주 좋은 성능을 보임

- modeling how these uses vary across linguistic contexts (i.e. to model polysemy)

→ sisg와 같은 context-independent representation 방법이 아래의 상황에 대처할 수 있는가?

Eg.

Chico Ruiz made a spectacular play on Alusiks's grounder...
Olivia De Havilland signed to do a Broadway play for Garson....

결국 sentence를 보고 word를 representation 해야!

sisg : subword information skip gram aka "FastText"

Introduction (2/3)

context-dependent한 word representation은 sentence를 봐야하며, sentence encoder로써 deep bidirectional rnn은 다음과 같은 특징을 갖고 있음

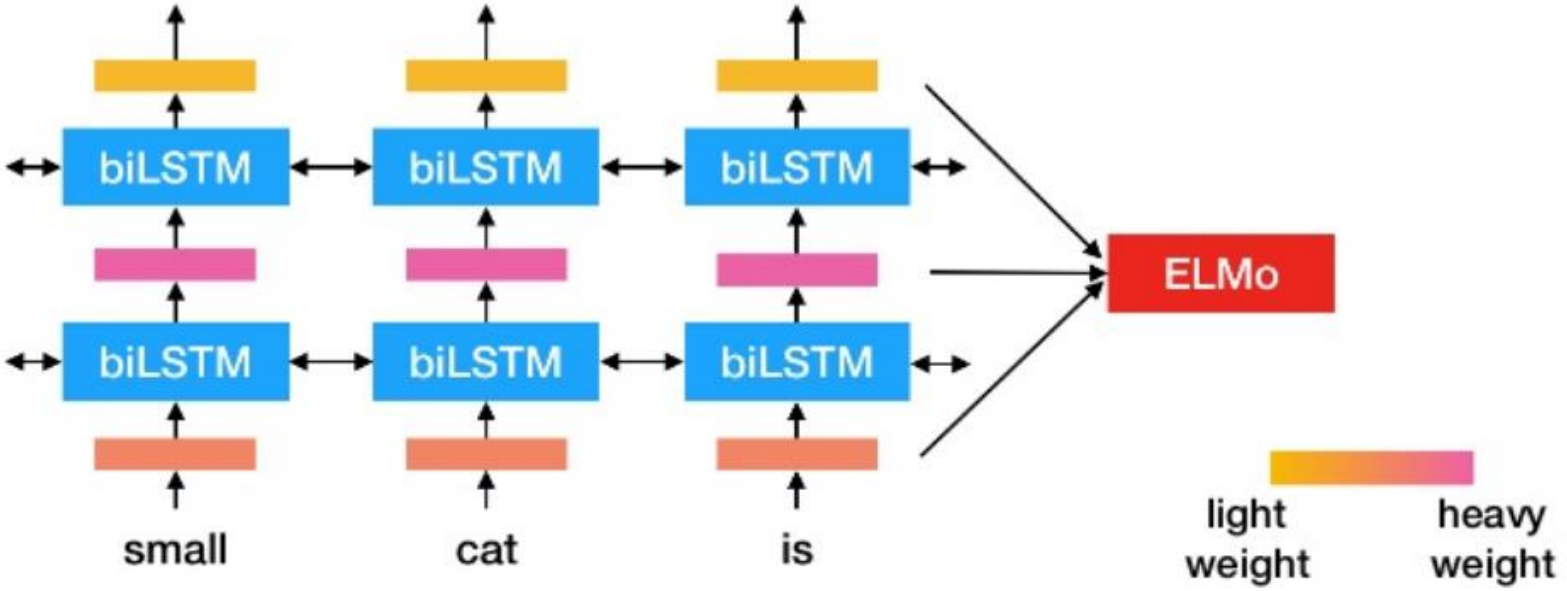
Previous work has also shown that different layers of deep biRNNs encode different types of information. For example, introducing multi-task syntactic supervision (e.g., part-of-speech tags) at the lower levels of a deep LSTM can improve overall performance of higher level tasks such as dependency parsing (Hashimoto et al., 2017) or CCG super tagging (Søgaard and Goldberg, 2016). In an RNN-based encoder-decoder machine translation system, Belinkov et al. (2017) showed that the representations learned at the first layer in a 2-layer LSTM encoder are better at predicting POS tags then second layer. Finally, the top layer of an LSTM for encoding word context (Melamud et al., 2016) has been shown to learn representations of word sense.

deep bidirectional rnn의 layer 별로 서로 다른 information을 잘 encoding 한다!

Introduction (3/3)

"Embeddings from Language Model (ELMo)"는 deep bidirectional rnn 기반의 context-dependent 방식으로 두 가지 요소를 잘 modeling, 아래의 특징들이 존재

- each token is assigned a representation that is a function of entire input sentence.
- using vectors derived from a bidirectional LSTM that is trained with a coupled language model objective on a large text corpus (biLM)
- easily being integrated into existing model (eg. textual entailment, question answering, sentiment analysis, etc.)



https://www.slideshare.net/JunyaKamura/deep-contextualized-word-representations-92582770

ELMo - Bidirectional language models

제안하는 방법론의 기반이 되는 bidirectional language model (biLM)은 아래와 같은 구조를 지니며, 대용량의 monolingual data (300m sentences)에 training됨

Given a sequence of N tokens, $(t_1, t_2, \ldots, t_N)$

1. computing a context-independent token representation $\mathbf{x}_k$

   (via token embeddings or a CNN over characters)

2. passing it through forward L-layers LSTM and backward L-layers LSTM

$$p(t_1, t_2, .., t_N) = \prod_{k=1}^{N} p(t_k | t_1, t_2, \ldots, t_{k-1}) \rightarrow \vec{\mathrm{h}}_{k,j}^{\mathrm{LM}} \; of \; t_k \; given \; (t_1, \ldots, t_{k-1})$$

$$p(t_1, t_2, .., t_N) = \prod_{k=1}^{N} p(t_k | t_{k+1}, t_{k+2}, \ldots, t_N) \rightarrow \overleftarrow{\mathrm{h}}_{k,j}^{LM} \; of \; t_k \; given \; (t_{k+1}, \ldots, t_N)$$

3. jointly maximizing the log likelihood of the forward and backward directions

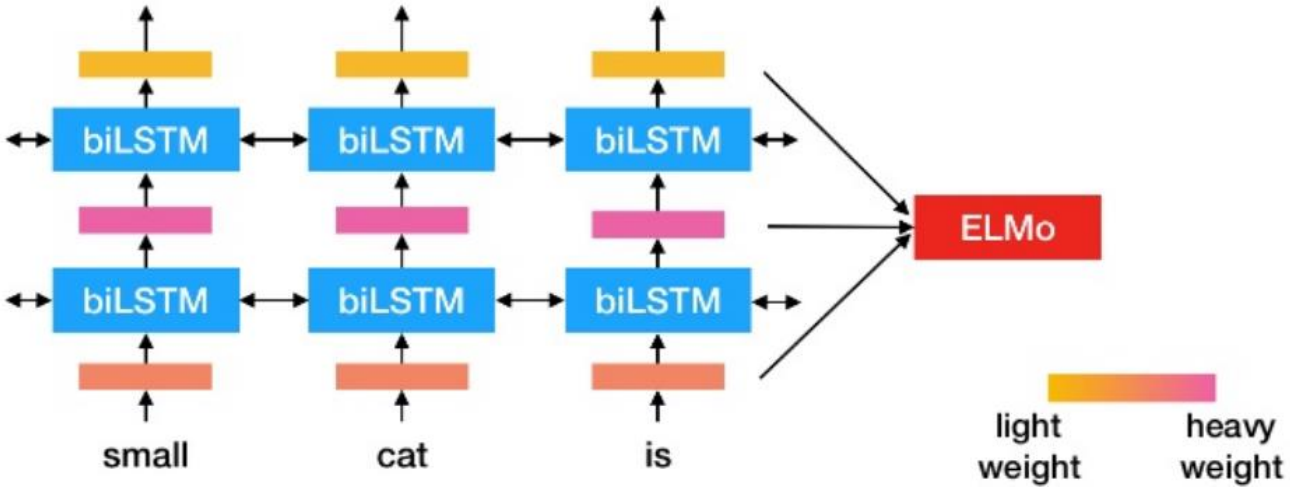
$$\sum_{k=1}^{N} \left(\log p\left(t_k | t_1, \ldots, t_{k-1}; \theta_{\mathrm{x}}, \vec{\theta}_{\mathrm{LSTM}}, \theta_{\mathrm{s}}\right) + \log \left(p\left(t_k | t_{k+1}, \ldots, t_N; \theta_{\mathrm{x}}, \overleftarrow{\theta}_{\mathrm{LSTM}}, \theta_{\mathrm{s}}\right)\right)\right)$$

$\theta_{\mathrm{x}} : weights \; of \; token \; representation \; (forward, backward \; are \; tied)$

$\theta_{\mathrm{s}} : weights \; of \; softmax \; layer \; (forward, backward \; are \; tied)$

ELMo - ELMo

ELMo는 sentence를 input으로 받아, deep biLM이 encoding한 token 별 internal layer representation을 linear combination하여 encoding 하는 것



For a each token t_k, a L-layer biLM computes a set of $2L + 1$ representations,

$$R_k = \{x_k, \overrightarrow{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM} \mid j = 1, \dots, L\} = \{h_{k,j}^{LM} \mid j = 0, \dots, L\}$$
$$where\ h_{k,0}^{LM}\ is\ the\ token\ layer\ and\ h_{k,j}^{LM} = [\overrightarrow{h}_{k,j}^{LM}; \overleftarrow{h}_{k,j}^{LM}]$$

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

$$ELMo_k = E(R_k; \theta_e)$$
$$ELMo_k^{task} = E(R_k; \theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} h_{k,j}^{LM}$$
$$s^{task} : softmax - normalized\ weights, \gamma^{task} : scale\ parameter$$

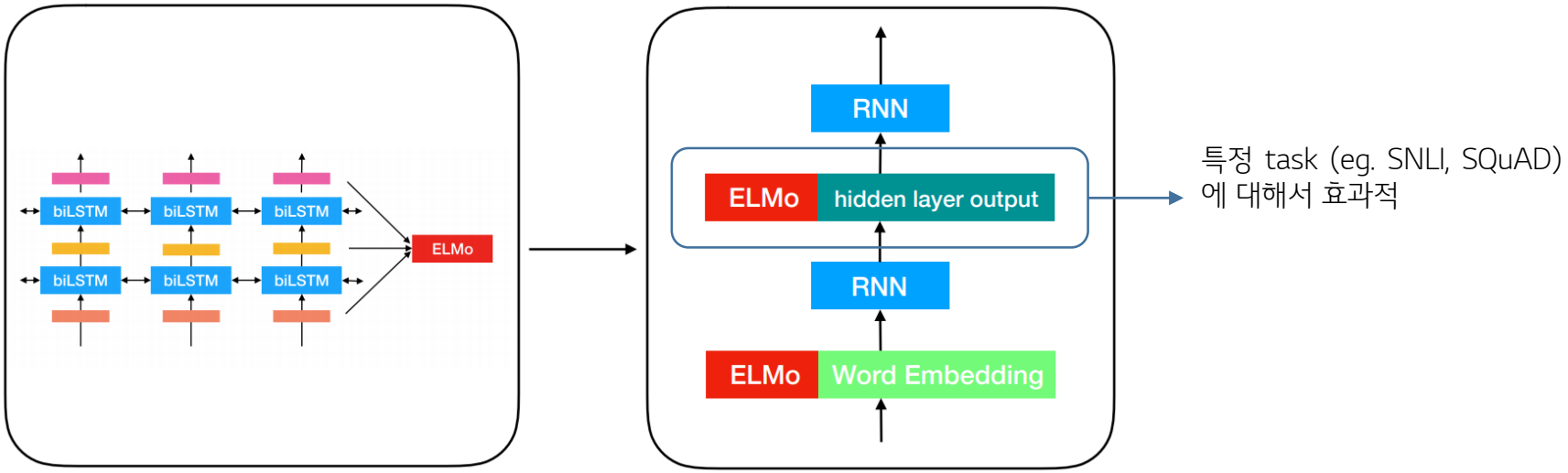
https://www.slideshare.net/JunyaKamura/deep-contextualized-word-representations-92582770

ELMo - Using biLMs for supervised NLP tasks

task model (eg. sentiment analysis, named entity recognition, etc.)은 단순히 token 별로 pre-trained biLM의 결과를 linear combination하는 weight만 학습

To add ELMo to the supervised model,

1. freezing the weights of the biLM
2. concatenating the ELMo vector $\mathbf{ELMo}_k^{task}$ with $\mathbf{x}_k$ → $[\mathbf{x}_k; \mathbf{ELMo}_k^{task}]$
3. passing $[\mathbf{x}_k; \mathbf{ELMo}_k^{task}]$ into the task RNN (or task model) and the remainder of the supervised model remains unchanged.
4. when training the model, adding a moderate amount of dropout orregularizing the ELMo weights by adding $\lambda\|\mathbf{w}\|_2^2$ to the loss



https://www.slideshare.net/JunyaKamura/deep-contextualized-word-representations-92582770

ELMo - pre-trained bidirectional language model architecture

The pre-trained biLMs in this paper are similar to the architectures in Józefowicz et al. (2016) and Kim et al. (2015), but modified to support joint training of both directions and add a residual connection between LSTM layers. We focus on large

⋮

(2016). The final model uses $L = 2$ biLSTM layers with 4096 units and 512 dimension projections and a residual connection from the first to second layer. The context insensitive type representation uses 2048 character n-gram convolutional filters followed by two highway layers (Srivastava et al., 2015) and a linear projection down to a 512 representation. As a result, the biLM provides three layers of representations for each input token, including those outside the training set due to the purely character input. In contrast, traditional word em-

Evaluation

NLP model에 ELMo를 추가하는 것만으로 성능을 크게 개선, 이는 task model이 deep biLM이 encoding한 context를 활용할 수 있기 때문

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%

Table 1: Test set comparison of ELMo enhanced neural models with state-of-the-art single model baselines across six benchmark NLP tasks. The performance metric varies across tasks – accuracy for SNLI and SST-5; F_1 for SQuAD, SRL and NER; average F_1 for Coref. Due to the small test sizes for NER and SST-5, we report the mean and standard deviation across five runs with different random seeds. The "increase" column lists both the absolute and relative improvements over our baseline.

Analysis (1/3)

deep biLM의 top layer만 쓰기보다는 ELMo 방식이 성능이 좋음, 또한 ELMo의 representation을 task model에 결합하는 방식에 따라서도 성능에 차이가 존재

Task	Baseline	Last Only	All layers	
			λ=1	λ=0.001
SQuAD	80.8	84.7	85.0	**85.2**
SNLI	88.1	89.1	89.3	**89.5**
SRL	81.6	84.1	84.6	**84.8**

Table 2: Development set performance for SQuAD, SNLI and SRL comparing using all layers of the biLM (with different choices of regularization strength λ) to just the top layer.

Task	Input Only	Input & Output	Output Only
SQuAD	85.1	**85.6**	84.8
SNLI	88.9	**89.5**	88.7
SRL	**84.7**	84.3	80.9

Table 3: Development set performance for SQuAD, SNLI and SRL when including ELMo at different locations in the supervised model.

Analysis (2/3)

ELMo의 토대가되는 deep biLM은 context-dependent representation이 가능하며, lower layer의 representation의 경우, syntactic information을 잘 encoding

	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.

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.

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.

Analysis (3/3)

ELMo를 활용하면 training data의 efficiency가 올라감을 확인할 수 있으며, 각 NLP task 별로 task weight가 다름을 확인할 수 있음

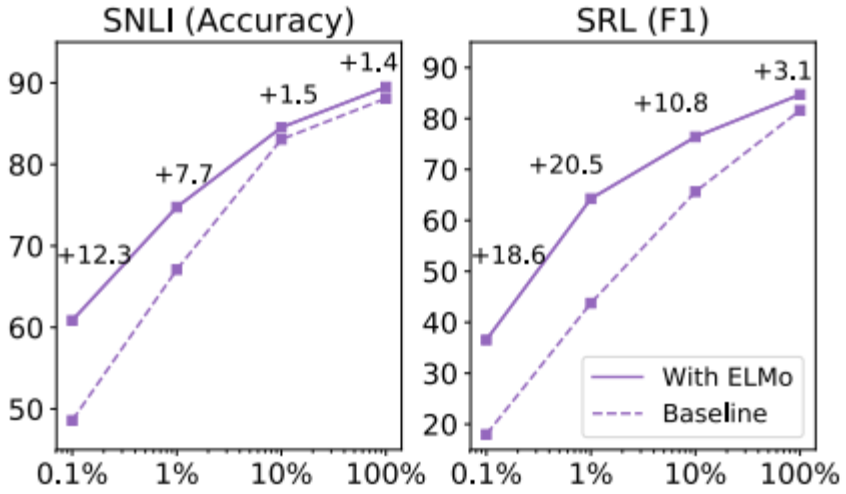


Figure 1: Comparison of baseline vs. ELMo performance for SNLI and SRL as the training set size is varied from 0.1% to 100%.

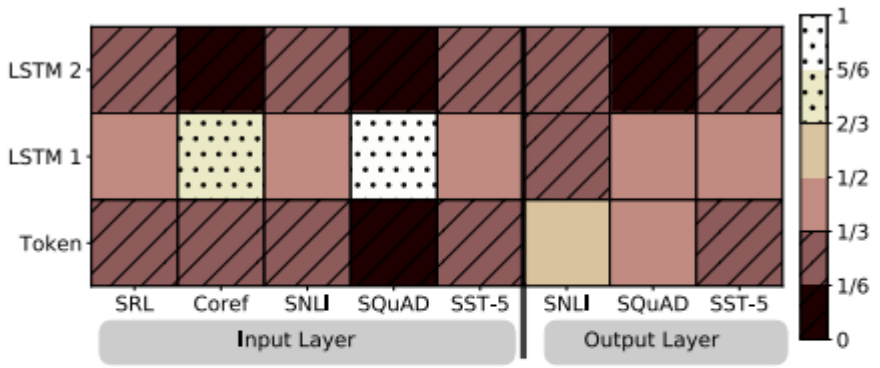


Figure 2: Visualization of softmax normalized biLM layer weights across tasks and ELMo locations. Normalized weights less then 1/3 are hatched with horizontal lines and those greater then 2/3 are speckled.

Conclusion

6 Conclusion

We have introduced a general approach for learning high-quality deep context-dependent representations from biLMs, and shown large improvements when applying ELMo to a broad range of NLP tasks. Through ablations and other controlled experiments, we have also confirmed that the biLM layers efficiently encode different types of syntactic and semantic information about words-in-context, and that using all layers improves overall task performance.

Q & A

감사합니다.