Deep contextualized word representations

Presenter: Longxu Dou

Paper Authors: Matthew E. Peters, Mohit Iyyer, Matt Gardner
Christopher Clark, Kenton Lee, Luke Zettlemoyer

Allen Institute for Artificial Intelligence & University of Washington

NAACL2018 Outstanding paper

May 24, 2018

Overview

Abstract

- Goal: Learning rich word representations
- Solution:
  - Training a deep bidirectional language mode (biLM) on large text corpus.
  - Combining the internal states to get different aspects of word meaning.
- Performance: Improve the state of the art across six challenging NLP problem.

What the **deep contextualized word representation** wants to model?

- Complex characteristics of word use (e.g., syntax and semantics)
- How these uses vary across linguistic contexts (i.e., to model polysemy)

Overview

Bidirectional language models

Given a sequence N tokens, $\{t_1, \ldots, t_N\}$

- Forward language predicts the token t_k given the history:

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

- Backward language predicts the token t_k given the future context:

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

- Jointly maximizes the log likelihood of the forward and backward directions:

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

ELMo

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

$$\begin{aligned} R_k &= \{\mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} \mid j = 1, \ldots, L\} \\ &= \{\mathbf{h}_{k,j}^{LM} \mid j = 0, \ldots, L\}, \end{aligned}$$

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

- Collapses all layers in R into a single vector $\mathbf{ELMo}_k = E(R_k; \mathbf{\Theta}_e)$

ELMo

Then compute a task specific weighting of all biLM layers:

$$\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}. \quad (1)$$

- $\mathbf{s}^{task}$: softmax-normalized weights
- γ^{task} :
  - Scalar parameter allows the task model to scale the entire vector
  - It's practically important to aid the optimization process due to the different distributions.

Using biLMs for supervised NLP tasks

- How to add ELMo into downstream task?
  - Run the biLM and record all of the layer representations for each word.
  - Pass the ELMo enhanced representation $[\mathbf{x}_k; \mathbf{ELMo}_k^{task}]$ into the task RNN.
  - Let the end task model learn a linear combination of these representations
- Regularization:
  - Add a moderate amount of dropout to ELMo.
  - Add $\lambda \|\mathbf{w}\|_2^2$ to the loss.
  - λ imposes an inductive bias on the ELMo weights to stay close to an average of all biLM layers.

Pre-trained bidirectional language model architecture

- Goal: Balance LM perplexity with model size and computational requirements
- Solution: Halve all embedding and hidden dimensions from the single best model `CNN-BIG-LSTM`
- Performance: After training for 10 epochs on the 1B Word, the average perplexities is 39.7, compared to 30.0 for the forward `CNN-BIG-LSTM`.

Fine tuning biLM

Dataset		Before tuning	After tuning
SNLI		72.1	16.8
CoNLL 2012 (coref/SRL)		92.3	-
CoNLL 2003 (NER)		103.2	46.3
SQuAD	Context	99.1	43.5
	Questions	158.2	52.0
SST		131.5	78.6

Figure: Fine tuning the biLM on task specific data typically resulted in significant drops in perplexity.

Overview

Evaluation across 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%

Figure: Test set comparison of ELMo enhanced neural models with state-of-the-art single model baselines across six benchmark NLP tasks.

Overview

Alternate layer weighting schemes

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**

Figure: 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.

Where to include ELMo?

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

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

What information is captured by the biLM's representations?

	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 .

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

What information is captured by the biLM's representations?

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*

Figure: WSD F_1 of Word sense disambiguation

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

Figure: PTB accuracies for POS tagging

- Higher-level LSTM states capture context-dependent word meaning
- Lower-level states model aspects of syntax

Sample efficiency

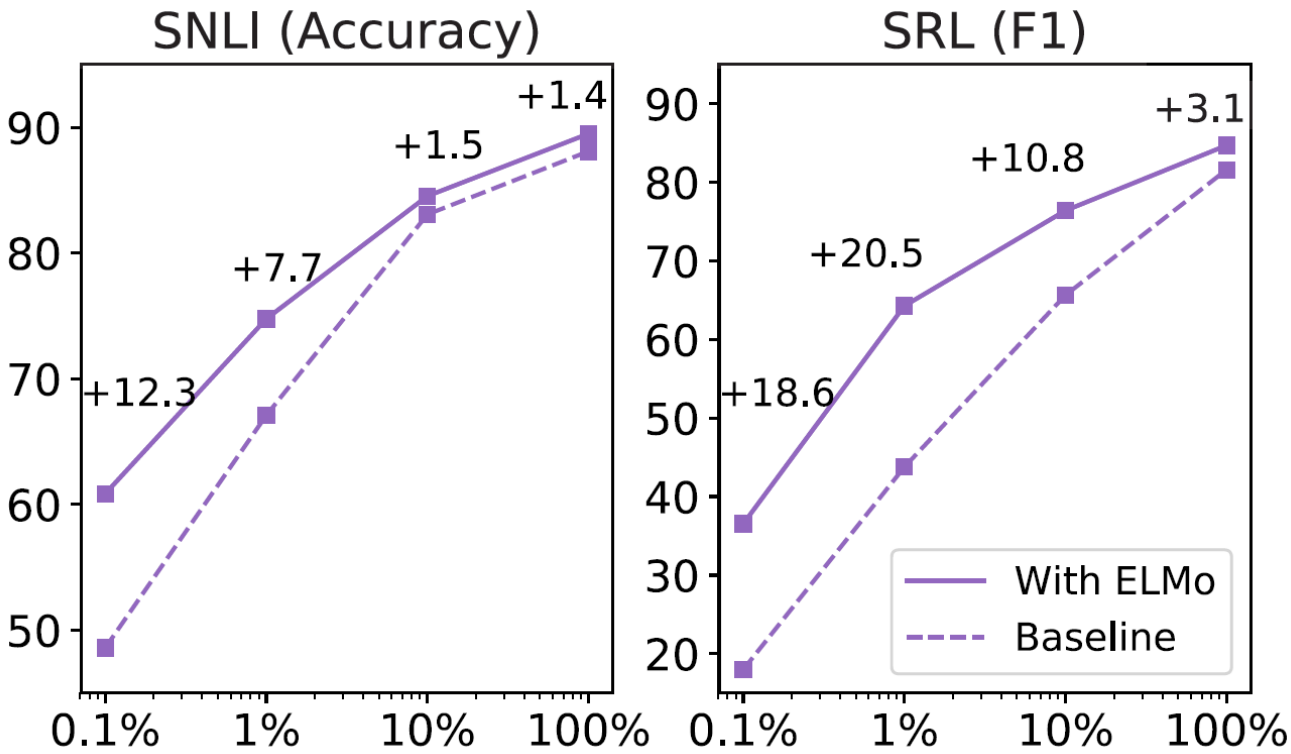

Figure: Comparison of baseline vs. ELMo performance for SNLI and SRL as the training set size is varied from 0.1% to 10%.

Visualization of learned weights

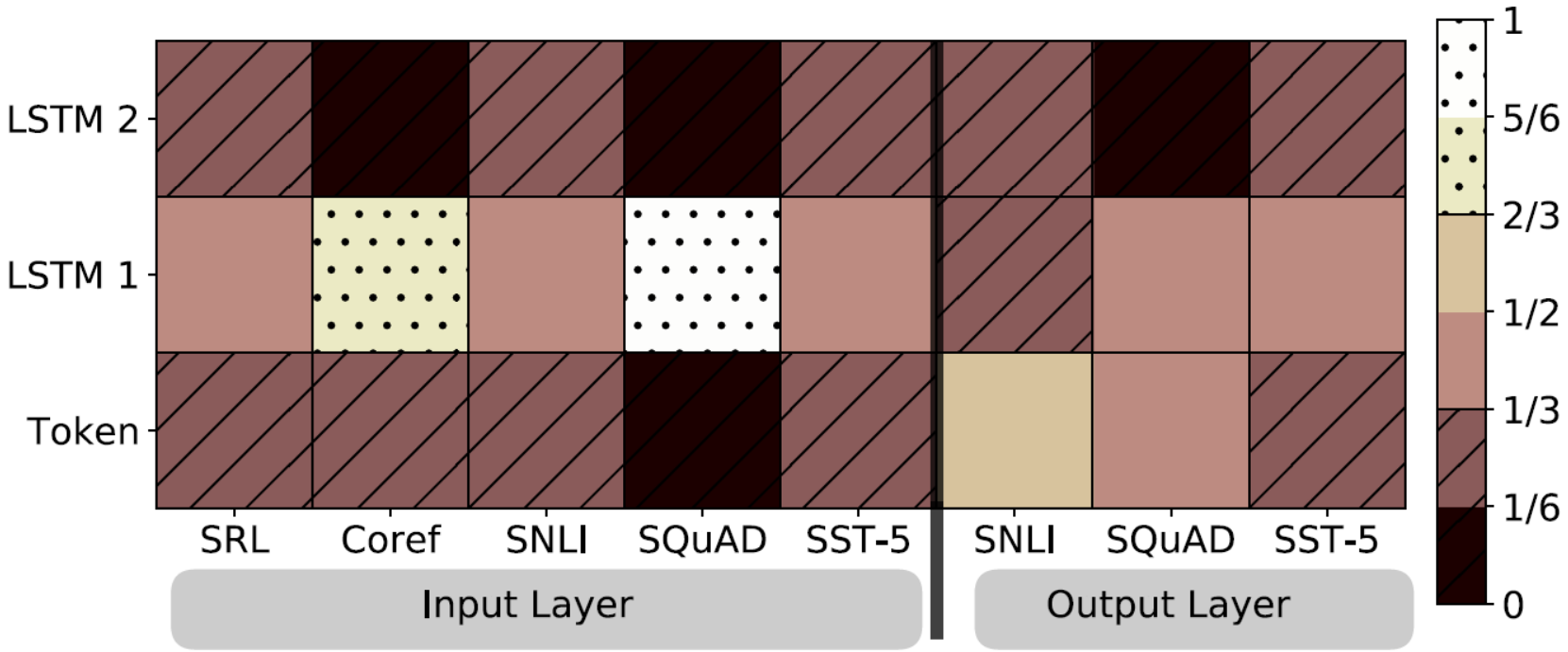

Figure: Visualization of softmax normalized biLM layer weights across tasks and ELMo locations. Normalized weights less than 1/3 are hatched with horizontal lines and those greater than 2/3 are speckled.

Overview

Salient Features

- Contextual : The word representation depends on the entire context in which it is used.
- Deep : The word representations combine all layers of a deep pre-trained neural network.
- Character based : Allow the network to use morphological clues to form robust representations for out-of-vocabulary tokens unseen in training.