Deep contextualized word representations

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer

NAACL-HLT 2018

- **Overview**
- Method
- Evaluation
- Analysis
- Comments

Overview

- Propose a new type of deep contextualised word representations (**ELMo**) that model:
  - Complex characteristics of word use (e.g., syntax and semantics)
  - How these uses vary across linguistic contexts (i.e., to model polysemy)
- Show that ELMo can improve existing neural models in various NLP tasks
- Argue that ELMo can capture more abstract linguistic characteristics in the higher level of layers

Example

	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.

GloVe mostly learns *sport*-related context

ELMo can distinguish the word sense based on the context

Method

- Overview
- Method
- Evaluation
- Analysis
- Comments

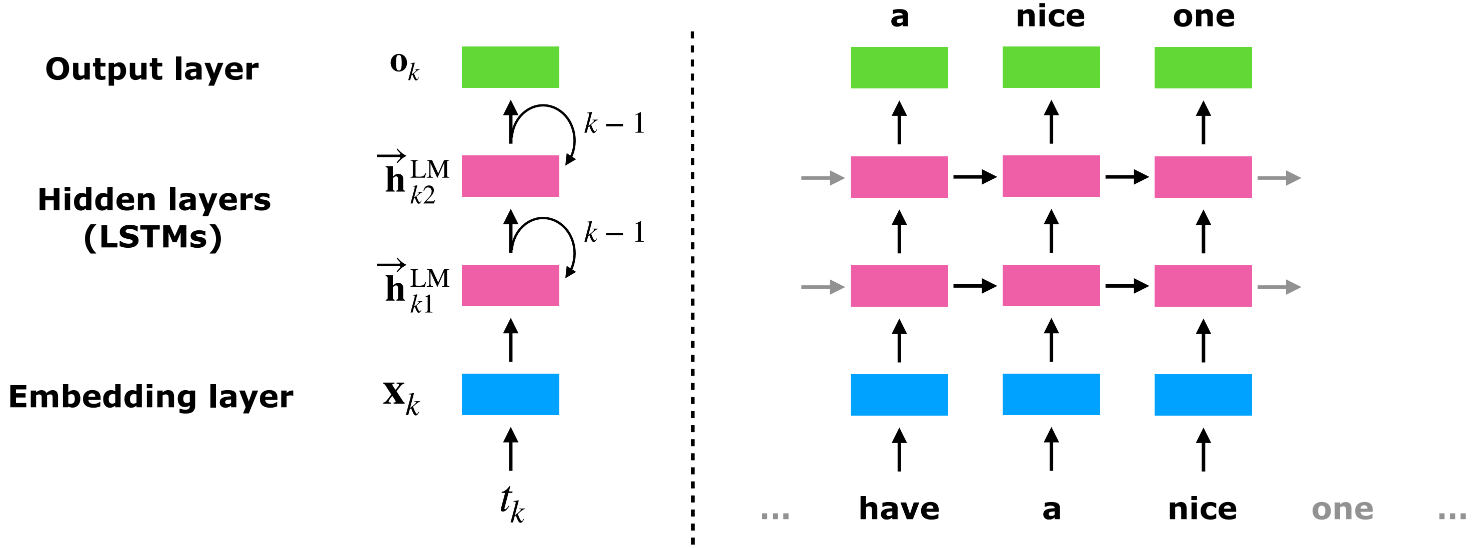
- Embeddings from Language Models: **ELMo**
- Learn word embeddings through building *bidirectional language models* (biLMs)
  - biLMs consist of forward and backward LMs
    - Forward: $$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k \mid t_1, t_2, \ldots, t_{k-1})$$
    - Backward: $$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k \mid t_{k+1}, t_{k+2}, \ldots, t_N)$$

Method

- Overview
- **Method**
- Evaluation
- Analysis
- Comments

With long short term memory (LSTM) network, predicting the next words in both directions to build biLMs

The forward LM architecture | Expanded in the forward direction of *k*

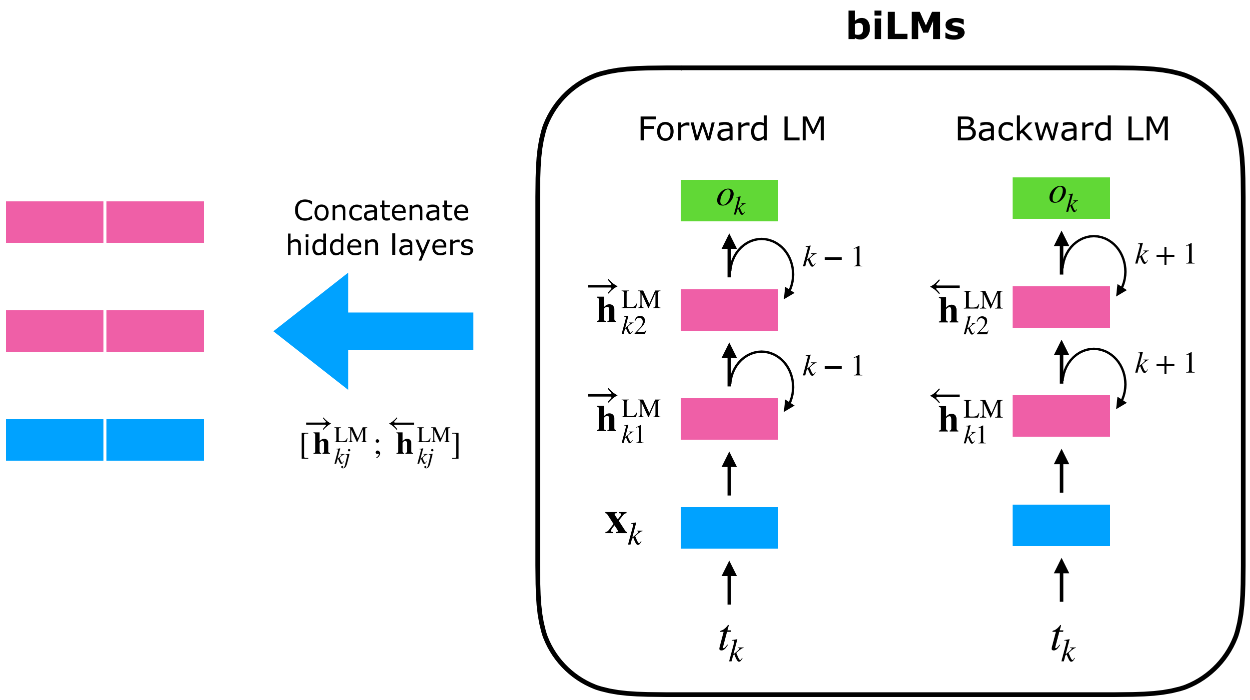


Method

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

$$\mathbf{ELMo}_k^{\text{task}} = \gamma^{\text{task}} \times \sum \left\{ \begin{array}{l} s_2^{\text{task}} \times \mathbf{h}_{k2}^{\text{LM}} \\ s_1^{\text{task}} \times \mathbf{h}_{k1}^{\text{LM}} \\ s_0^{\text{task}} \times \mathbf{h}_{k0}^{\text{LM}} \\ \quad ([\mathbf{x}_k ; \mathbf{x}_k]) \end{array} \right.$$

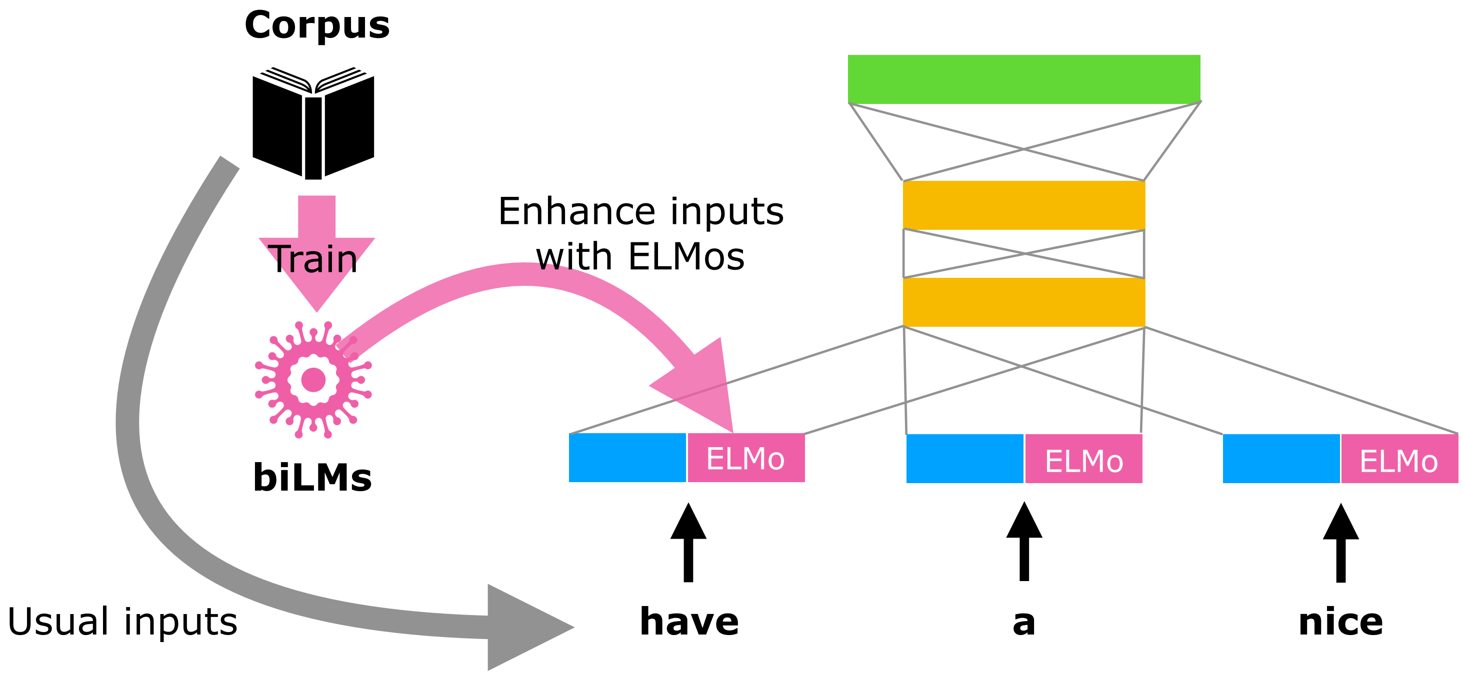


Unlike usual word embeddings, ELMo is assigned to every *token* instead of a *type*

Method

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

- Overview
- Method
- Evaluation
- Analysis
- Comments



Evaluation

Many linguistic tasks are improved by using ELMo

- Overview
- Method
- **Evaluation**
- Analysis
- Comments

	Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
Q&A	SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
Textual entailment	SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
Semantic role labelling	SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coreference resolution	Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
Named entity recognition	NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
Sentiment analysis	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

- Overview
- Method
- Evaluation
- **Analysis**
- Comments

The higher layer seemed to learn semantics while the lower layer probably captured syntactic features

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.

Analysis

The higher layer seemed to learn semantics while the lower layer probably captured syntactic features**???**

Most models preferred "syntactic (probably)" features

Even in sentiment analysis

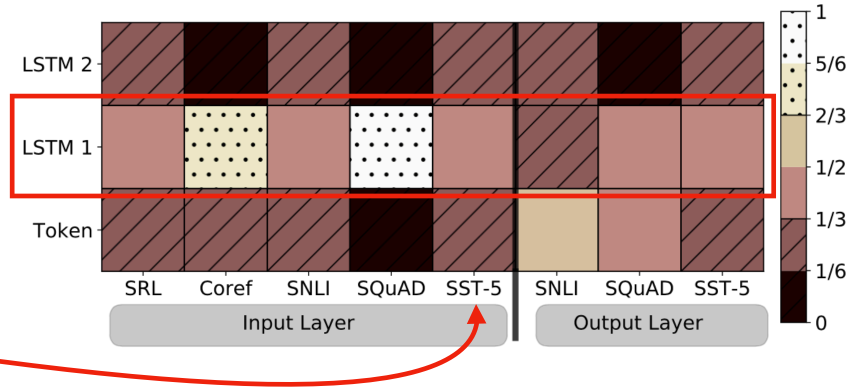


Figure 2: Visualization of softmax normalized biLM layer weights across tasks and ELMo locations. Normalized weights less then $1/3$ are hatched with horizontal lines and those greater then $2/3$ are speckled.

Analysis

ELMo-enhanced models can make use of small datasets more efficiently

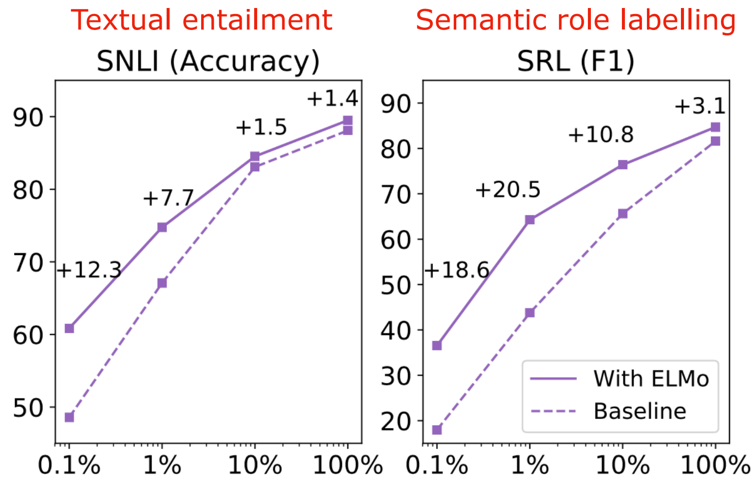


Figure 1: Comparison of baseline vs. ELMo performance for SNLI and SRL as the training set size is varied from 0.1% to 100%.

- Overview
- Method
- Evaluation
- **Analysis**
- Comments

- Overview
- Method
- Evaluation
- Analysis
- **Comments**

Comments

- Pre-trained ELMo models are available at https://allennlp.org/elmo
  - AllenNLP is a deep NLP library on top of PyTorch
  - AllenNLP is a product of AI2 (Allen Institute for Artificial Intelligence) which works on other interesting projects like Semantic Scholar
- ELMo can process character-level inputs
  - Japanese (Chinese, Korean, …) ELMo models likely to be possible