Giovanni Luca
Favuzzi

# Deep Contextualized Word Representations

TU-Berlin @ MAR 4.033
03-07-2019

## Abstract

This paper introduces a novel way to represent words: ELMo word embeddings. These embeddings are deep contextualized (they are function of all the internal layers of the neural architecture) and can represent both semantic and syntactic meaning of any word but, most importantly, they can model polysemy as well. Polysemy is explicitly handled since the embedding depends on the word as well as the the context (the full sentence). The embeddings are computed as a weighted average of all internal representations of a deep bi-directional Recurrent Neural Network with LSTM cells. ELMo embeddings defines new state of the art performances on six different Natural Language Processing tasks.

## Introduction

Word Representation are a powerful tool to enhance NLP models, nonetheless the enhancement highly depends on the quality of the representations: they should represent both syntactic and semantic features as well as polysemy: the ability of a word to get different meanings depending on the context it is put into.
Previous works struggle representing decently polysemy, moreover, their word representation is function of only the last layer of the deep neural network.
This paper shows how each layer outputs a representation that is either more syntactic (lower levels) than semantic (higher levels) or vice-versa, depending on its deepness.
ELMo embeddings depend not only on the word to be represented but on the surrounding context too, moreover, they are function of the internal representations of all layers of the deep neural model. The employed neural model is a bi-directional Long Short Term Memory, trained on a coupled language model objective.
It is shown how ELMo embeddings can be easily incorporated in several different models and enhance each one of them.

## Methodology

### Bi-LM

ELMo word embeddings depend on the full sentence and they are computed on top of a two-layer biLMs with character convolutions as a function of the internal states of the network.
Considering a N-tokens sentence, the forward bi-directional Language Model computes the probability of the sentence from left to right and from right to left.

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

The convolutional parameters: $\Theta_x$ and the softmax parameters $\Theta_s$ are tied together for both directions, instead the parameters for the bi-directional LSTM are kept separated.

### Embedding Interpolation

Each layer of the biLM returns 2L+1 embeddings:

$$R_k = \left\{ \mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} \mid j = 1...L \right\} = \left\{ \mathbf{h}_{k,j}^{LM} \mid j = 0...L \right\}$$

Having: the convolutional token representation as ***x*** and the biLSTM embedding as ***h***.
Finally, we can represent the ELMo embeddings as a task specific weighting of all the biLM layers:

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

Having: gamma as a task specific constant and **s** as softmax weights vector.

### Task Integration

Integrating ELMo in a specific task is simple: the pre-trained biLM runs over the tokens and records each inter-layer representation **h**. Finally the end-task model tunes the interpolation parameters between the intermediate embeddings stacking the ELMo representations to the input.

### biLM Architecture

The pre-trained bidirectional Language Model architecture is a bi-directional Recurrent Neural Network enhanced with LSTMs and projection layers, interleaved by residual connections.

## Evaluation

ELMo embeddings show how well they perform in the table below: they are used to enhance previous models, pushing them to a new state of the art, for six different NLP tasks.
For each previous SOTA, the employed model remains more or less untouched, while the ELMo embeddings are fine tuned on the task.

| Task | Previous SOTA | | Our baseline | ELMo + baseline | Increase (absolute/ relative) |
|---|---|---|---|---|---|
| SQuAD | Liu et al. (2017) [1] | 84.4 | 81.1 | 85.8 | 4.7 / 24.9% |
| SNLI | Chen et al. (2017) [2] | 88.6 | 88.0 | $88.7 \pm 0.17$ | 0.7 / 5.8% |
| SRL | He et al. (2017) [3] | 81.7 | 81.4 | 84.6 | 3.2 / 17.2% |
| Coref | Lee et al. (2017) [4] | 67.2 | 67.2 | 70.4 | 3.2 / 9.8% |
| NER | Peters et al. (2017) [5] | $91.93 \pm 0.19$ | 90.15 | $92.22 \pm 0.10$ | 2.06 / 21% |
| SST-5 | McCann et al. (2017) [6] | 53.7 | 51.4 | $54.7 \pm 0.5$ | 3.3 / 6.8% |

1. Question Answering: an attention layer is added to the base model and GRUs are switched to LSTMs.
2. Textual Entailment: the base model remains untouched.
3. Semantic Role Labelling: the base model remains untouched.
4. Coreference Resolution: the base model remains untouched.
5. Name Entity Recognition: the key difference is the interpolation between embeddings at different layers. The base model only uses the top-layer representation.
6. Sentiment Analysis: ELMo completely replaces the CoVe representations.

## Conclusion

ELMo embeddings have hold SOTA performances up until the introduction of BERT embeddings. Nonetheless, they are the first representations to explicitly represent polysemy and show how different layers in the neural architecture can output embeddings with different meanings.

[1]
Stochastic answer networks
for machine reading comprehension
[2]
Enhanced lstm for natural language inference
[3]
End-to-end neural coreference resolution
[4]
Deep semantic role
labeling: What works and what's next
[5]
Semi-supervised sequence tagging
with bidirectional language models
[6]
Learned in translation: Contextualized word vectors

SS19_HOT