Contextualized Word Representations

Viraj Nadkarni

ECE 594, UIUC

2022

Static Representations

- Word2Vec, GloVe, etc.
- Idea : represent each word by a vector in $\mathbb{R}^d$.
- Enforce structure via a loss function: e.g. In GloVe, we enforce that inner product between two words should be a good estimate of co-occurence counts
- Still gave us surprisingly meaningful results :

$$v_{King} - v_{Man} + v_{Woman} = v_{Queen}$$

$$v_{Children} - v_{Child} + v_{Bird} = v_{Birds}$$

- Use these vectors as inputs to neural networks (like RNNs) which perform specific tasks. Provides advantages because of the underlying structure.

Shortcomings

- How do you deal with polysemy?
  - **Semantic :** Words differ in meaning based on context.
  - **Syntactic :** Words differ in part-of-speech based on context.
- e.g.
  - *The manufacturing* **plant** *was shut down after the strikes.*
  - *The coffee* **plant** *is native to South America.*
  - *The spy used her skills to* **plant** *a bug in the secret meeting room.*
- To distinguish word sense, we need contextual representations
- i.e. $v_{word} = F(word, context)$

ELMo

Method proposed in this paper : **ELMo** (Embeddings from Language Models)

- Embeddings = Vector representation
- Language Model = Model trained on the task of representing the probability distribution over the entire vocabulary, given previous text

$$p_\theta(w_0, w_1, \ldots, w_n) = \prod_{i=1}^{n} p_\theta(w_i | w_0, \ldots, w_{i-1})$$

- Here the conditional distribution p_θ is modelled using Recurrent Neural Networks
- But how do you take care of context both before *and* after a word?

ELMo : Theory

Uses forward and backward language models

- Forward : Next word prediction

$$p_\theta(w_0, w_1, \ldots, w_n) = \prod_{i=1}^{n} p_\theta(w_i | w_0, \ldots, w_{i-1})$$

- Backward : Previous word prediction

$$p_{\theta'}(w_0, w_1, \ldots, w_n) = \prod_{i=0}^{n-1} p_{\theta'}(w_i | w_{i+1}, \ldots, w_n)$$

Idea : Train θ, θ' to maximize

$$LL(\theta, \theta') = \sum_{i=1}^{n} log(p_\theta(w_i | w_0, \ldots, w_{i-1})) + log(p_{\theta'}(w_i | w_{i+1}, \ldots, w_n))$$

Once the model has been trained, freeze parameters and simply use the hidden states of the model as word vectors!

ELMo : Implementation

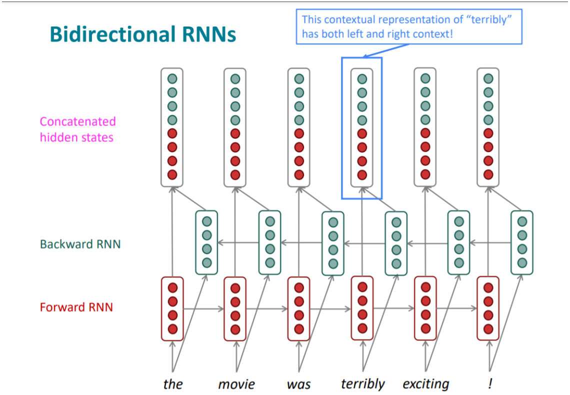

Bidirectional RNNs
This contextual representation of "terribly" has both left and right context!
Concatenated hidden states
Backward RNN
Forward RNN
the
movie
was
terribly
exciting
!

Figure 1: Bi-RNNs

For each word w_k in a sentence, the above model constructs hidden states $\overrightarrow{h_k}$ and $\overleftarrow{h_k}$. Their concatenation $h_k = [\overrightarrow{h_k} | \overleftarrow{h_k}]$ is used as the "net" hidden state.

ELMo : Implementation

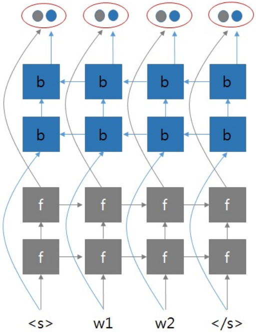


Figure 2: Multilayered Bi-RNNs

- In practice we have multiple layers in the RNN as well.
- Therefore each layer would have a separate hidden state $h_{k,l}$ where $l \in \{1, 2, \ldots, L\}$

ELMo : Implementation

- Given the set of all the hidden states for word w_k, we get the ELMo representation as :

$$ELMo_k = \gamma \sum_{l=0}^{L} s_l h_{k,l}$$

- Here s_l are weights chosen so that they sum to 1. The choice of both γ and s_l depends on the task.
- We shall see an interesting heuristic later to guide the choice of s_l

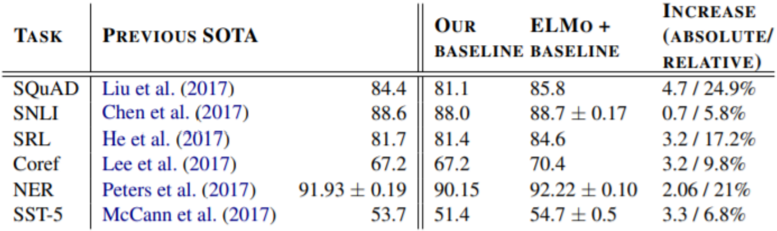
ELMo : How do you use these contextual word vectors?

- **Pre-Training :** ELMo embeddings are obtained by pretraining on the language modelling task and then freezing the parameters to given contextual representation of any word in a test sentence.
- These pretrained embeddings are concatenated with static embeddings and given as an input to models performing downstream tasks.

ELMo : Evaluation on downstream tasks

- **Question-Answering** : Find answers to a question from a given passage.
- **Textual Entailment** : Is a hypothesis true given a premise?
- **Semantic role labelling** : Tag the subject, object and predicate in a given sentence.
- **Coreference Resolution** : Given a large passage/sentence, what entity does each pronoun refer to ?
  e.g : The trophy did not fit into the suitcase because **it** was too *big/small*.
- **Named entity extraction** : Extract named entities and classify them as person, location, organisation, etc.
- **Sentiment Analysis** : Given a review of movie classify if positive or negative.

ELMo : Evaluation on downstream tasks



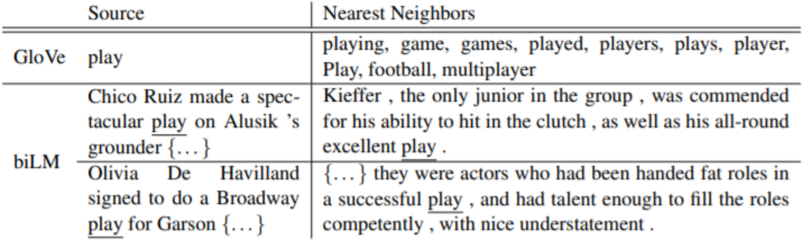
Task	Previous SOTA		Our baseline	ELMo + baseline	Increase (absolute/ relative)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (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 3: SQuAD : QnA task , SNLI : Textual Entailment task, SRL : Semantic Role Labelling, Coref : Coreference resolution, NER : Named entity extraction, SST-5 : Sentiment Analysis

ELMo : Where do you include contextual vectors?

- Depends on architecture of the downstream model
- Authors observed that for models which had attention layers at the output, it was useful to have ELMo embedding at the input as well as the last layer of the downstream model
- For other models, most of the improvements came from including the contextual embeddings only at the input

ELMo : What do contextual vectors tell us?



	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 4: Nearest neighbours : Static vs Contextual embeddings

Nearest neighbours in contextual case change correctly based on context of the polysemous word "play"

ELMo : What do contextual vectors tell us?

Experiments :

- **Word sense disambiguation :** Given a word and context, what is the sense it is being used in?
  Observed that hidden states of **later layers of RNN perform better than initial layers** in this task
- **POS tagging :** Given a word and context, what part-of-speech is it being used as?
  Observed that hidden states of **initial layers of RNN perform better than later layers** in this task

Conclusion :

- Initial layers encode syntactic information
- Final layers encode semantic information

ELMo : Are gains only in performance?

- **Improves the performance of simple baseline models** on a variety of tasks and makes them better than SOTA
- Drastic **decrease in training time** for tasks that had less training data. (e.g. For SRL task, had 98% reduction in number of epochs to reach the same performance)
- This tells us that most of the heavy lifting of encoding semantic and syntactic information is being done by ELMo after being pre-trained
- The downstream model can now focus on optimizing the task rather than also try to embed this information - **Pretrain-Finetune paradigm**

ELMo : Are gains only in performance?

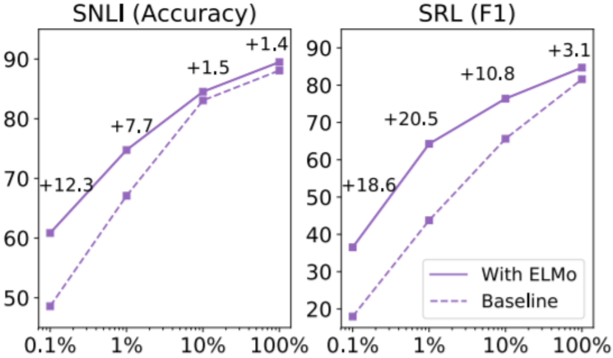


Figure 5: Better performance is reached even at less training data

ELMo : Limitations

- Takes care of the bidirectonal aspect of context, but not long term aspect due to the use of RNNs. Attention Layers improve on that aspect.
- Carries all the limitations that come with using RNNs : exploding/vanishing gradients, hard to parallelize training, etc.
- Contextual embedding weights independent of task?
- Phrase/sentence level embeddings?
- Common embeddings across languages?

THANK YOU!