Natural Language Processing Contextualized Embeddings, Pre-Training, Fine-Tuning and Large Language Models

Felipe Bravo-Marquez

June 7, 2023

Representations for a word

- So far, we've basically had one representation of words, the word embeddings we've already learned: Word2vec, GloVe, fastText.[1].
- These embeddings have a useful semi-supervised quality, as they can be learned from unlabeled corpora and used in our downstream task-oriented architectures (LSTM, CNN, Transformer).
- However, they exhibit two problems.
- Problem 1: They always produce the same representation for a word type regardless of the context in which a word token occurs
- We might want very fine-grained word sense disambiguation
- Problem 2: We just have one representation for a word, but words have different aspects, including semantics, syntactic behavior, and register/connotations

---

[1] These slides are based on the Stanford CS224N: Natural Language Processing with Deep Learning course:
http://web.stanford.edu/class/cs224n/

Neural Language Models can produce Contextualized Embeddings

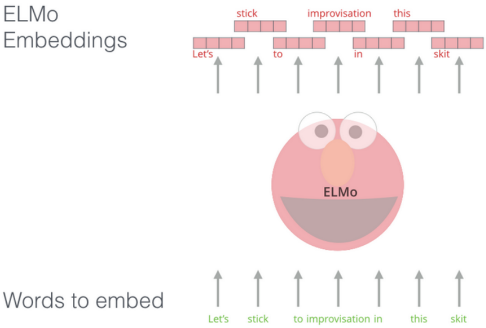
- So far, we've basically had one representation of words, the word embeddings we've already learned: Word2vec, GloVe, fastText.[2].
- These have two problems.
- Problem: Always the same representation for a word type regardless of the context in which a word token occurs
- We might want very fine-grained word sense disambiguation
- Problem 2: We just have one representation for a word, but words have different aspects, including semantics, syntactic behavior, and register/connotations

[2]These slides are based on the Stanford CS224N: Natural Language Processing with Deep Learning course: `http://web.stanford.edu/class/cs224n/`

ELMo: Embeddings from Language Models

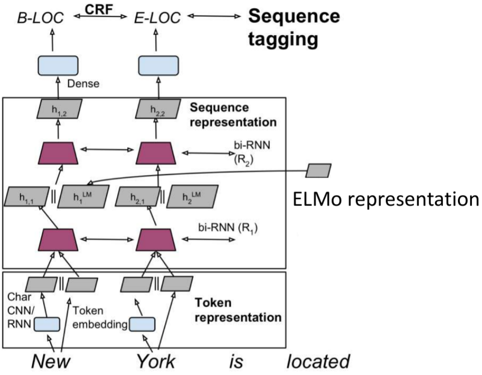
- Idea: train a large language model (LM) with a recurrent neural network and use its hidden states as “contextualized word embeddings” [Peters et al., 2018].
- ELMO is bidirectional LM with 2 biLSTM layers and around 100 million parameters.
- Uses character CNN to build initial word representation (only)
- 2048 char n-gram filters and 2 highway layers, 512 dim projection
- User 4096 dim hidden/cell LSTM states with 512 dim projections to next input
- Uses a residual connection
- Parameters of token input and output (softmax) are tied

ELMo: Embeddings from Language Models



ELMo: Use with a task

- First run biLM to get representations for each word
- Then let (whatever) end-task model use them
- Freeze weights of ELMo for purposes of supervised model
- Concatenate ELMo weights into task-specific model



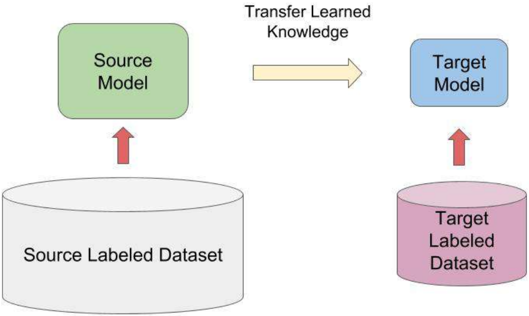
$$\mathbf{h}_{k,1} = [\overrightarrow{\mathbf{h}}_{k,1}; \overleftarrow{\mathbf{h}}_{k,1}; \mathbf{h}_k^{LM}]$$

ELMo: Results

Name	Description	Year	F1
ELMo	ELMo in BiLSTM	2018	92.22
TagLM Peters	LSTM BiLM in BiLSTM tagger	2017	91.93
Ma + Hovy	BiLSTM + char CNN + CRF layer	2016	91.21
Tagger Peters	BiLSTM + char CNN + CRF layer	2017	90.87
Ratinov + Roth	Categorical CRF+Wikipeda+word cls	2009	90.80
Finkel et al.	Categorical feature CRF	2005	86.86
IBM Florian	Linear/softmax/TBL/HMM ensemble, gazettes++	2003	88.76
Stanford	MEMM softmax markov model	2003	86.07

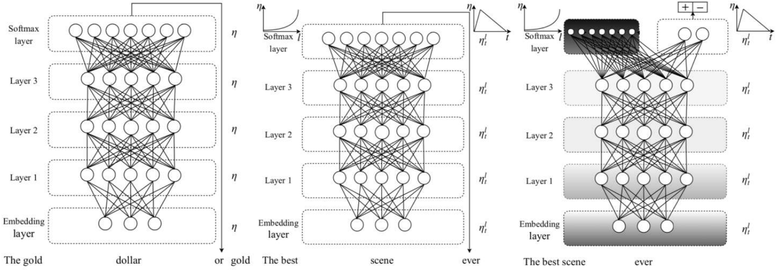
ULMfit

- Howard and Ruder (2018) Universal Language Model Fine-tuning for Text Classification [Howard and Ruder, 2018].
- Same general idea of transferring NLM knowledge
- Here applied to text classification



ULMfit

- Train LM on big general domain corpus (use biLM)
- Tune LM on target task data
- Fine-tune as classifier on target task



(a) LM pre-training (b) LM fine-tuning (c) Classifier fine-tuning

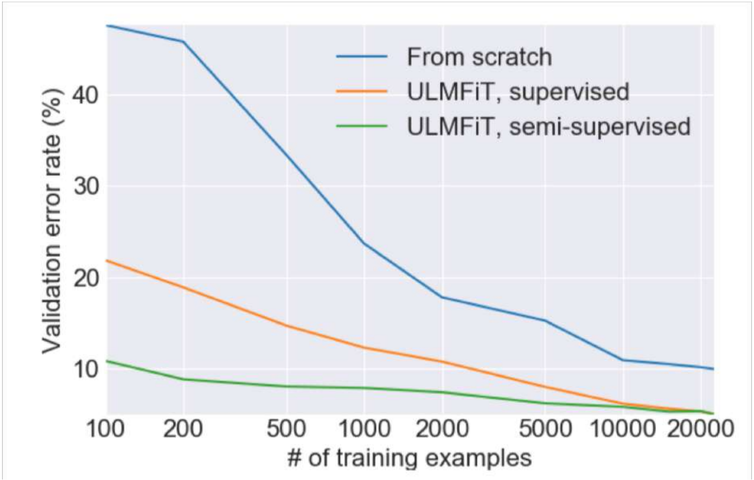
ULMfit emphases

- Use reasonable-size “1 GPU” language model not really huge one
- A lot of care in LM fine-tuning
- Different per-layer learning rates
- Slanted triangular learning rate (STLR) schedule
- Gradual layer unfreezing and STLR when learning classifier
- Classify using concatenation [h_T,maxpool(h),meanpool(h)]

	Model	Test		Model	Test
IMDb	CoVe (McCann et al., 2017)	8.2	TREC-6	CoVe (McCann et al., 2017)	4.2
	oh-LSTM (Johnson and Zhang, 2016)	5.9		TBCNN (Mou et al., 2015)	4.0
	Virtual (Miyato et al., 2016)	5.9		LSTM-CNN (Zhou et al., 2016)	3.9
	ULMFiT (ours)	**4.6**		ULMFiT (ours)	**3.6**

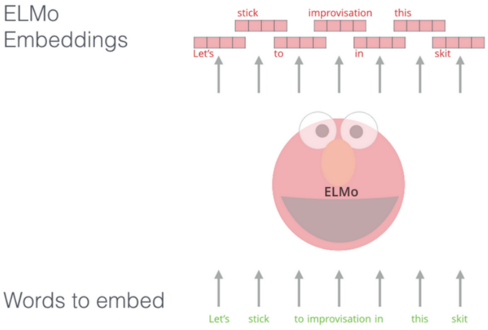
Text classifier error rates

ULMfit transfer learning



BERT

- Idea: train a large language model with a recurrent neural network and use its hidden states as “contextualized word embeddings” [Peters et al., 2018].



Questions?

Thanks for your Attention!

References I

Howard, J. and Ruder, S. (2018).
Universal language model fine-tuning for text classification.
In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018).
Deep contextualized word representations.
In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.