

Contextualized Word Embeddings

ELMo



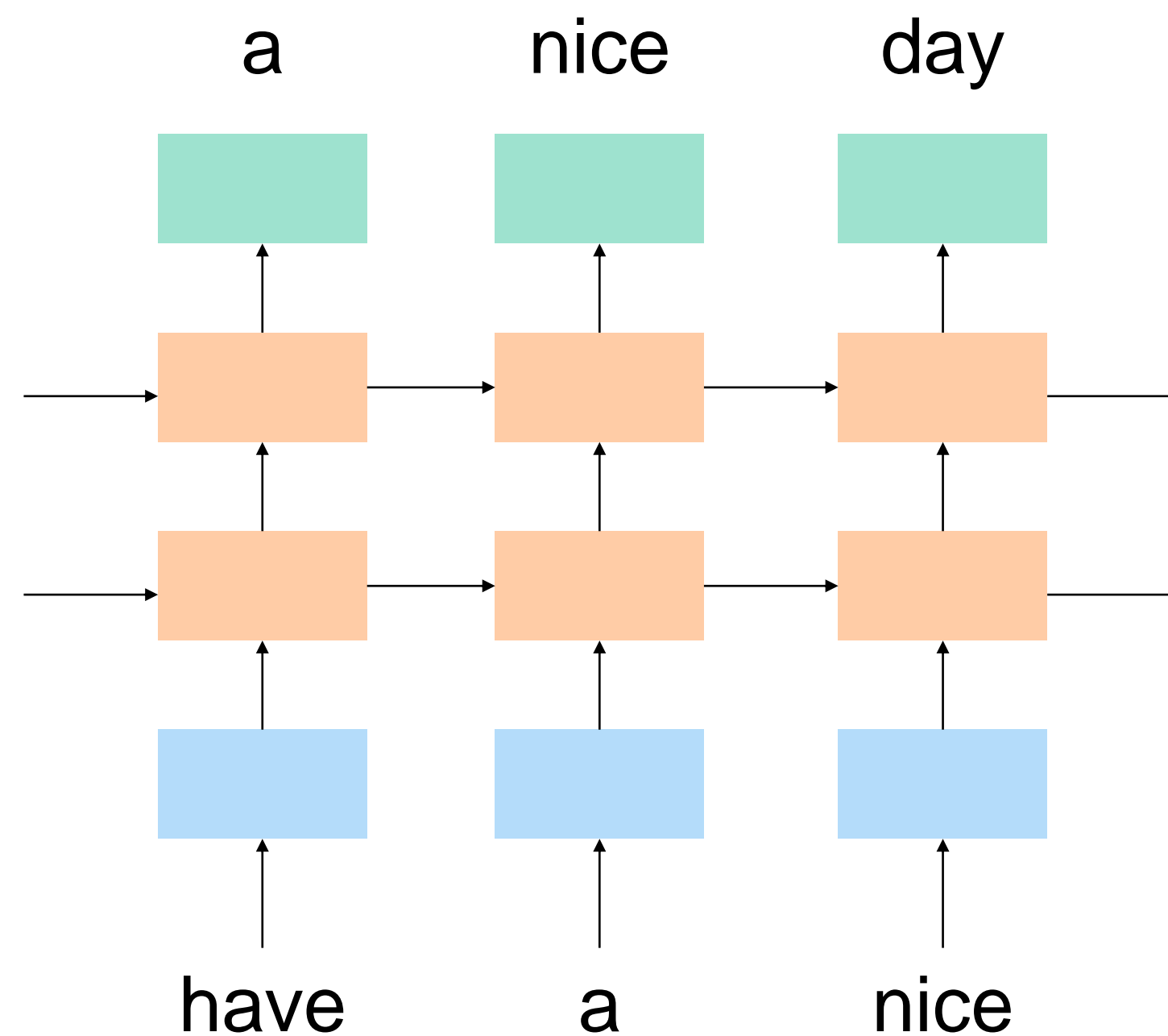
國立臺灣大學 資訊工程學系
陳縉儂 助理教授

<http://vivianchen.idv.tw>



ELMo: Embeddings from Language Models

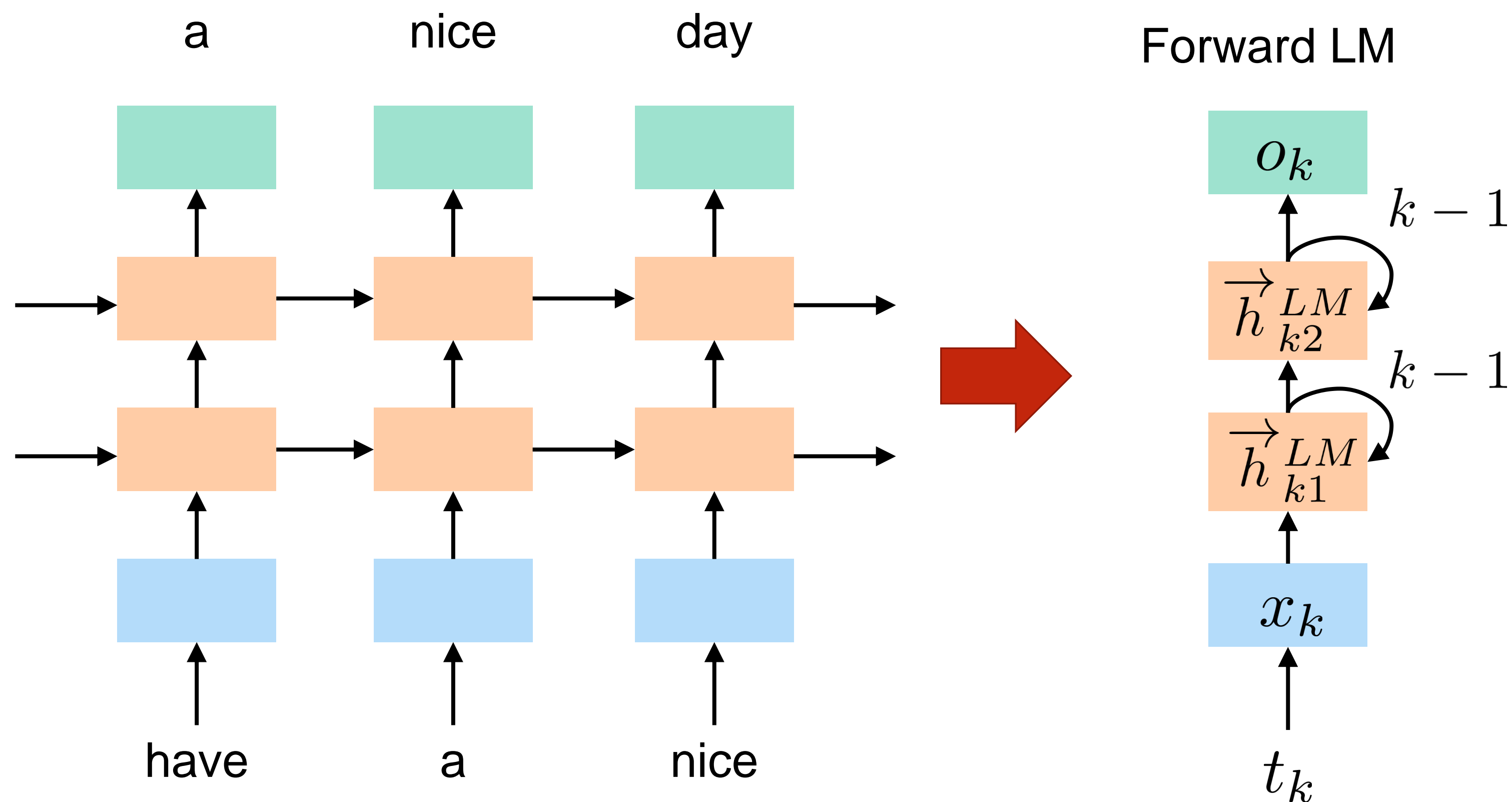
- Idea: contextualized word representations
 - ✓ Learn word vectors using long contexts instead of a context window
 - ✓ Learn a deep Bi-NLM and use all its layers in prediction





ELMo: Embeddings from Language Models

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





ELMo: Embeddings from Language Models

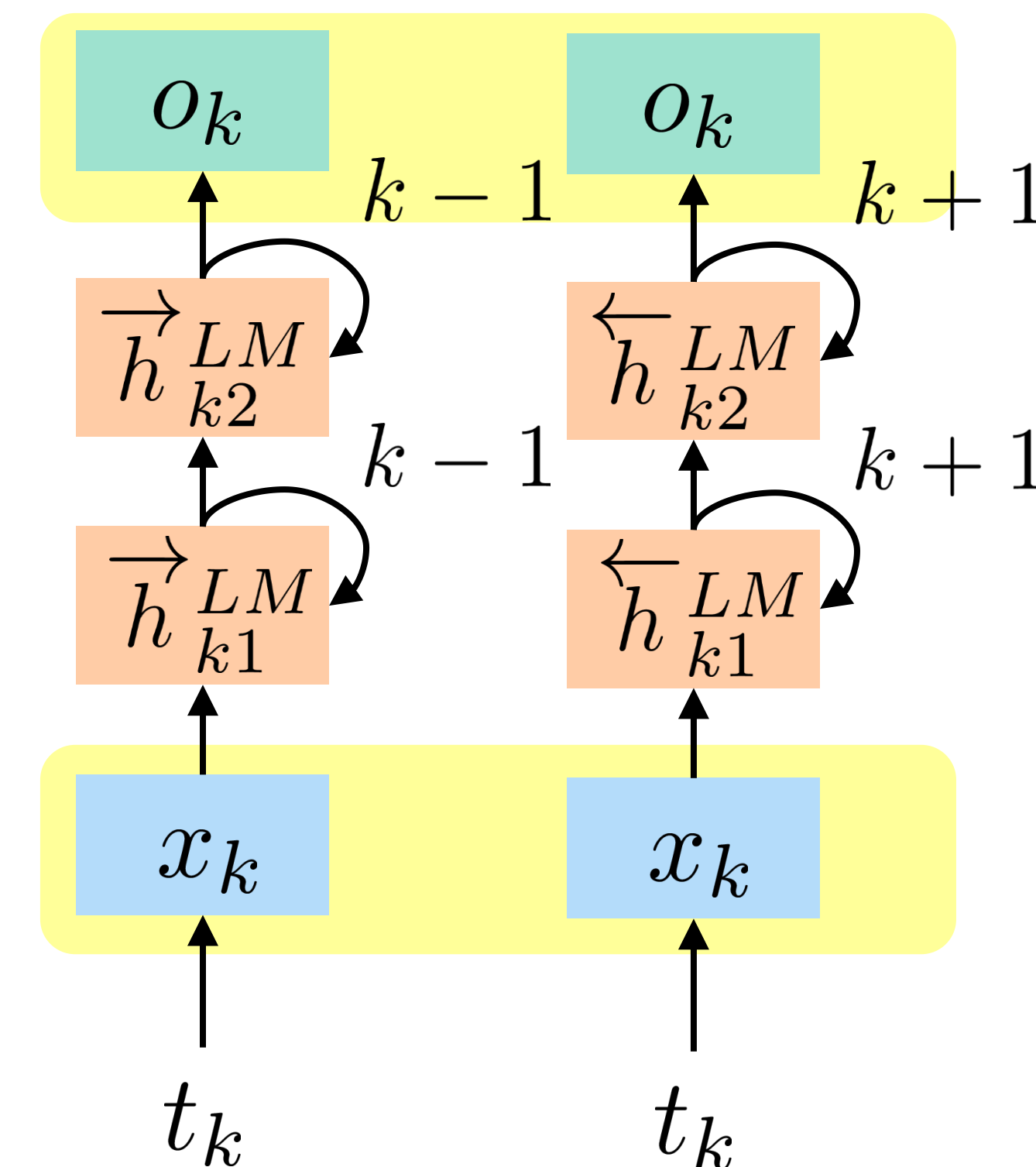
1) Bidirectional LM

$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_1, \dots, t_{k-1})$$
$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_{k+1}, \dots, t_N)$$

- Character CNN for initial word embeddings
2048 n-gram filters, 2 highway layers, 512 dim projection
- 2 BLSTM layers
- Parameter tying for input/output layers

$$O = \sum_{k=1}^N \left(\log p(t_k | t_1, \dots, t_{k-1}; \Theta_x, \vec{\Theta}_{LSTM}, \Theta_s) \right. \\ \left. + \log p(t_k | t_{k+1}, \dots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s) \right)$$

Forward LM Backward LM





ELMo: Embeddings from Language Models

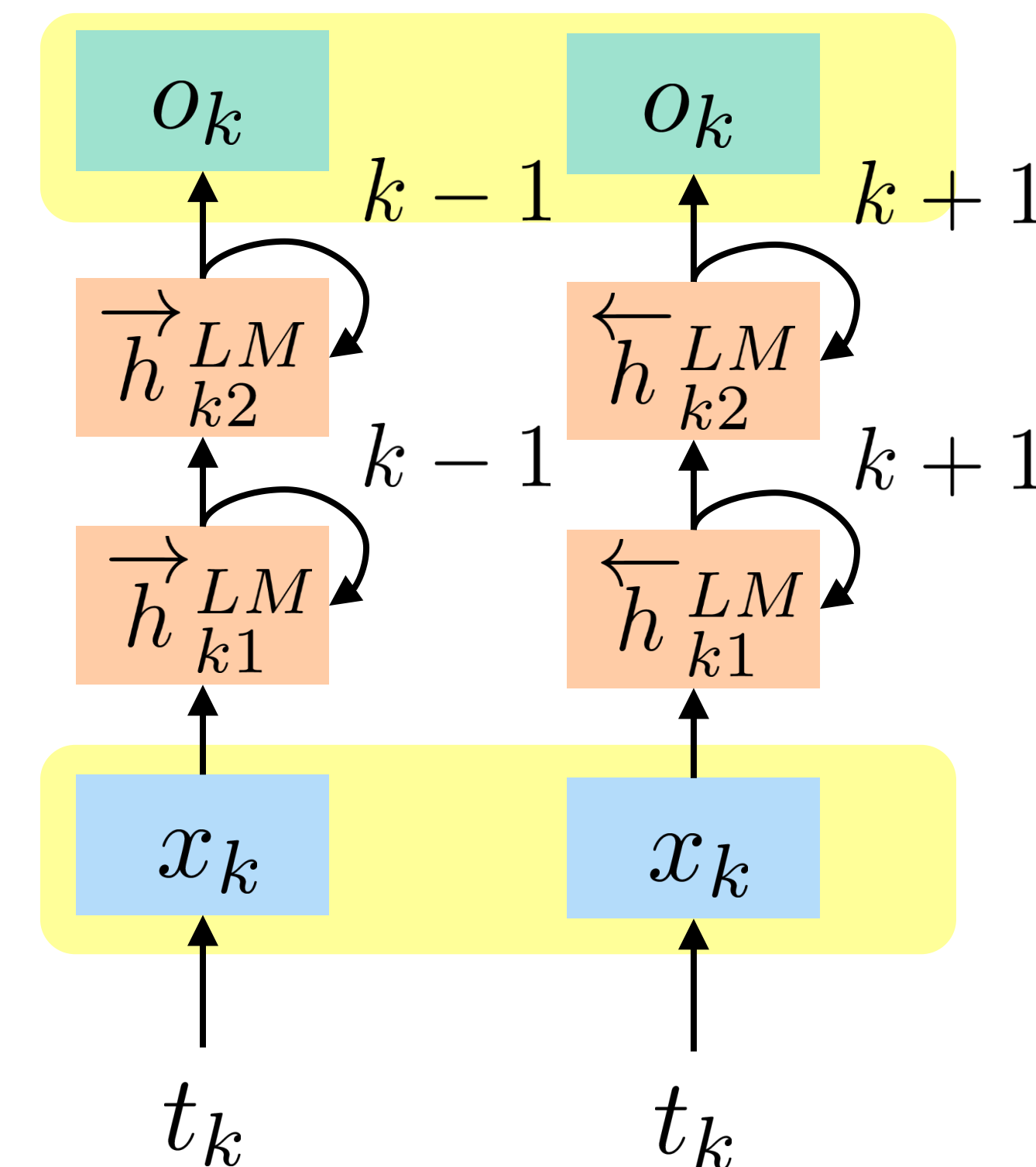
2) ELMo

- Learn the task-specific linear combination of LM embeddings
- Use multiple layers in LSTM instead of top one

$$\text{ELMo}_k^{\text{task}} = \gamma^{\text{task}} \times \sum \left\{ \begin{array}{l} s_2^{\text{task}} \times h_{k2}^{LM} \\ s_1^{\text{task}} \times h_{k1}^{LM} \\ s_0^{\text{task}} \times h_{k0}^{LM} \end{array} \right. \begin{array}{|c|c|} \hline \overrightarrow{h}_{k2}^{LM} & \overleftarrow{h}_{k2}^{LM} \\ \hline \overrightarrow{h}_{k1}^{LM} & \overleftarrow{h}_{k1}^{LM} \\ \hline x_k & x_k \\ \hline \end{array}$$

- γ^{task} scales overall usefulness of ELMo to task
- s^{task} are softmax-normalized weights
- optional layer normalization

Forward LM Backward LM



A task-specific embedding with combining weights learned from a downstream task



ELMo: Embeddings from Language Models

3) Use ELMo in Supervised NLP Tasks

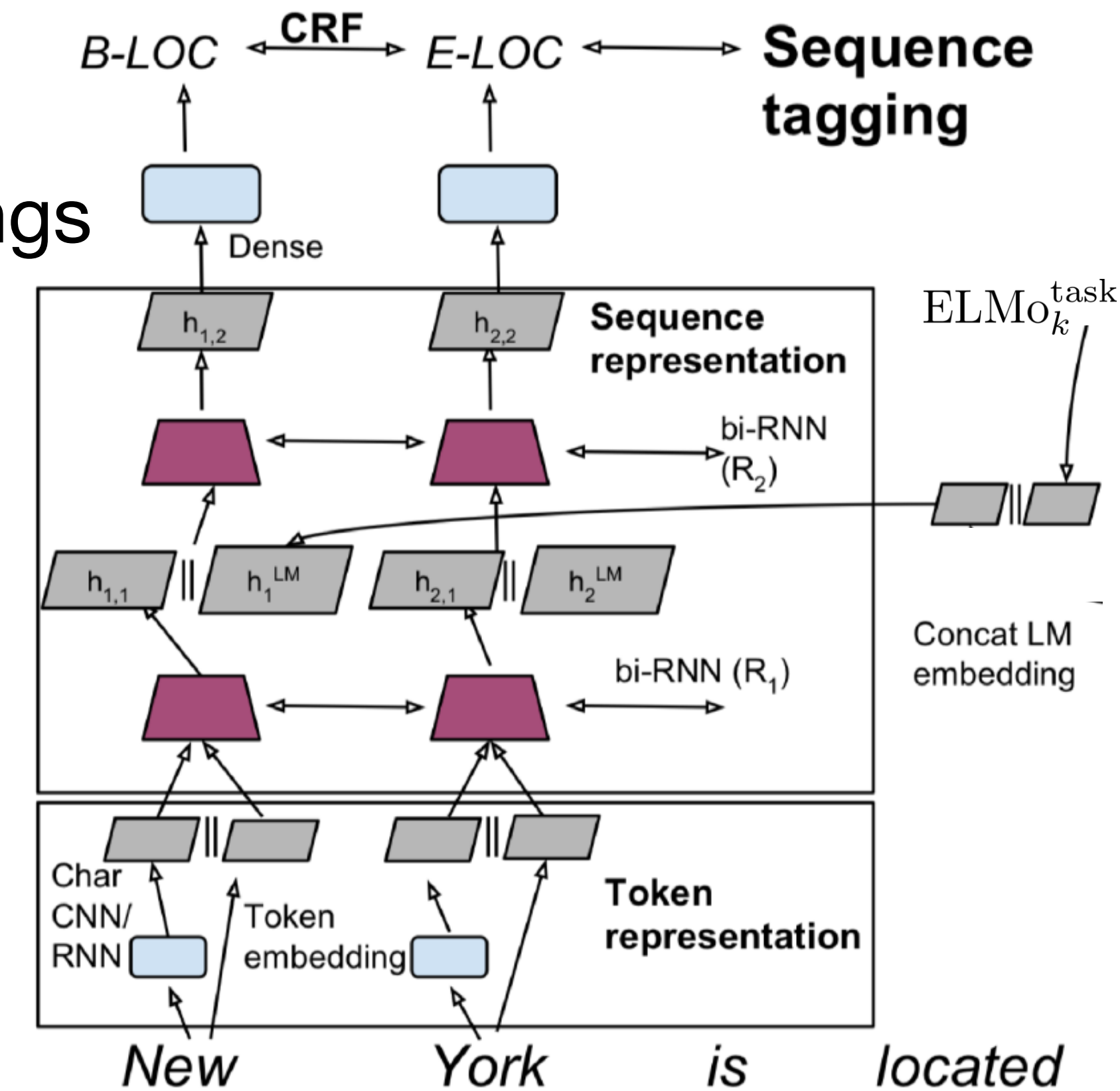
- Get LM embedding for each word
- Freeze LM weights to form ELMo enhanced embeddings

$[h_k; \text{ELMo}_k^{\text{task}}]$: concatenate ELMo into the intermediate layer

$[x_k; \text{ELMo}_k^{\text{task}}]$: concatenate ELMo into the input layer

- Tricks: dropout, regularization

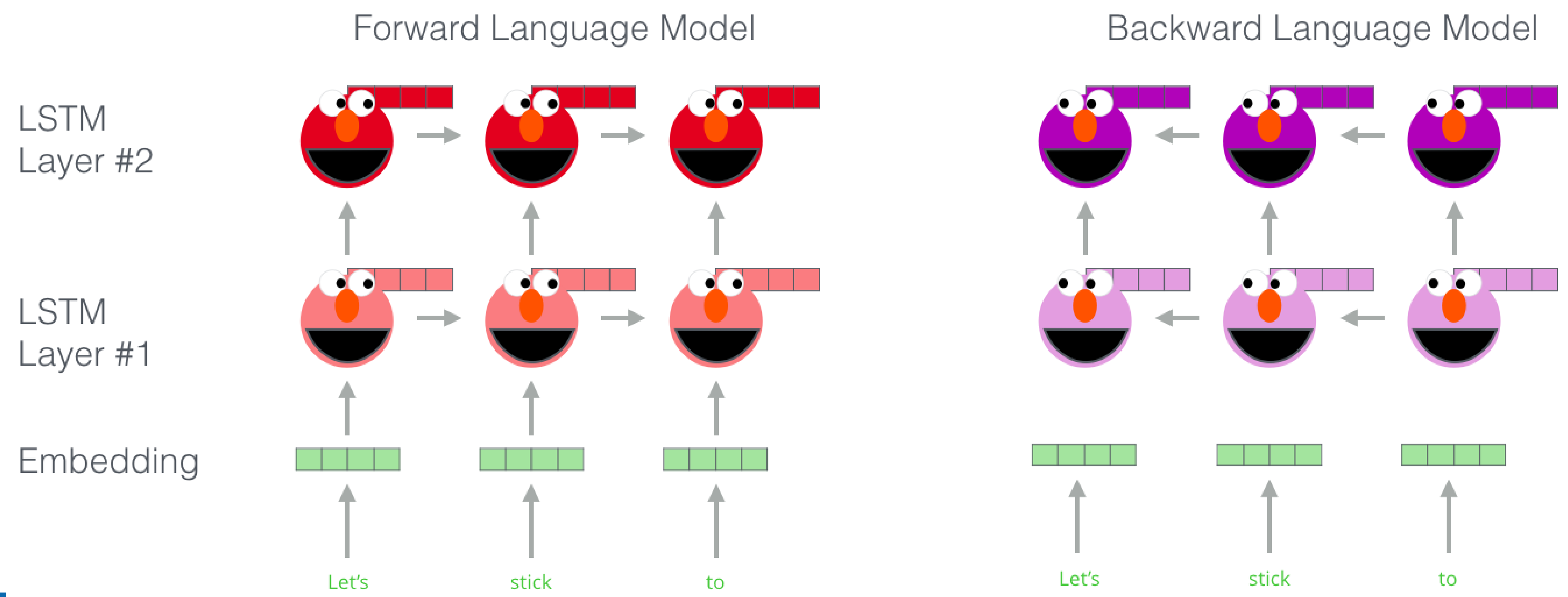
The way for concatenation depends on the task





ELMo Illustration

Embedding of “stick” in “Let’s stick to” - Step #1

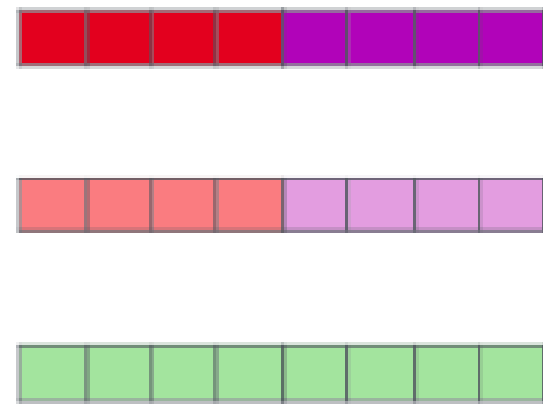




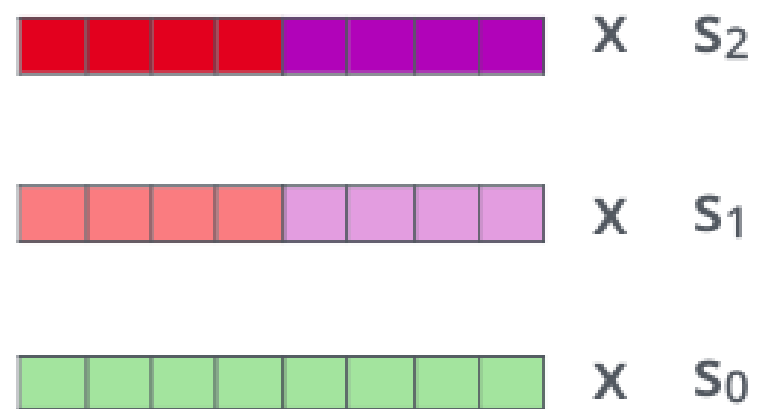
ELMo Illustration

Embedding of “stick” in “Let’s stick to” - Step #2

1- Concatenate hidden layers



2- Multiply each vector by a weight based on the task

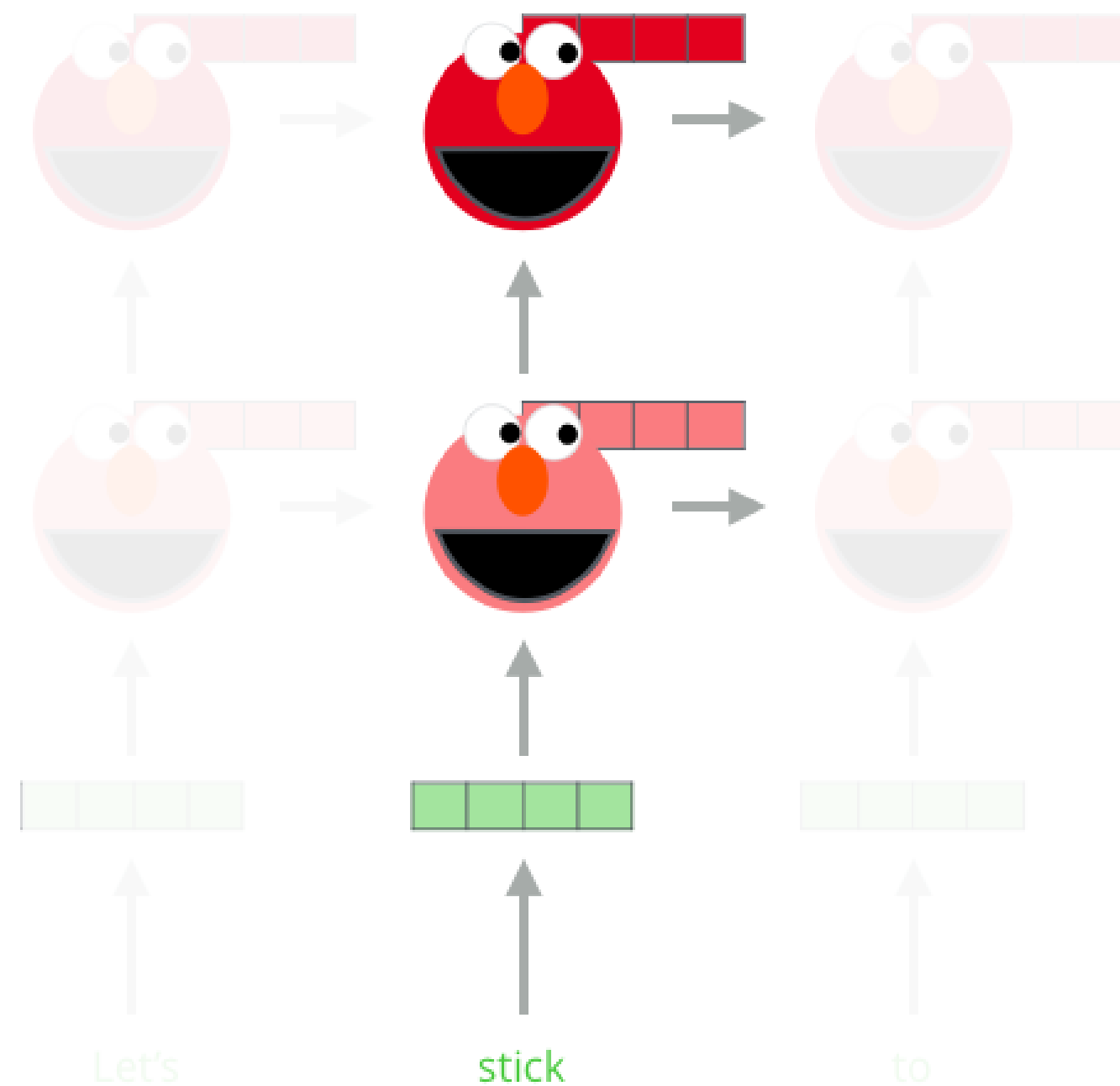


3- Sum the (now weighted) vectors

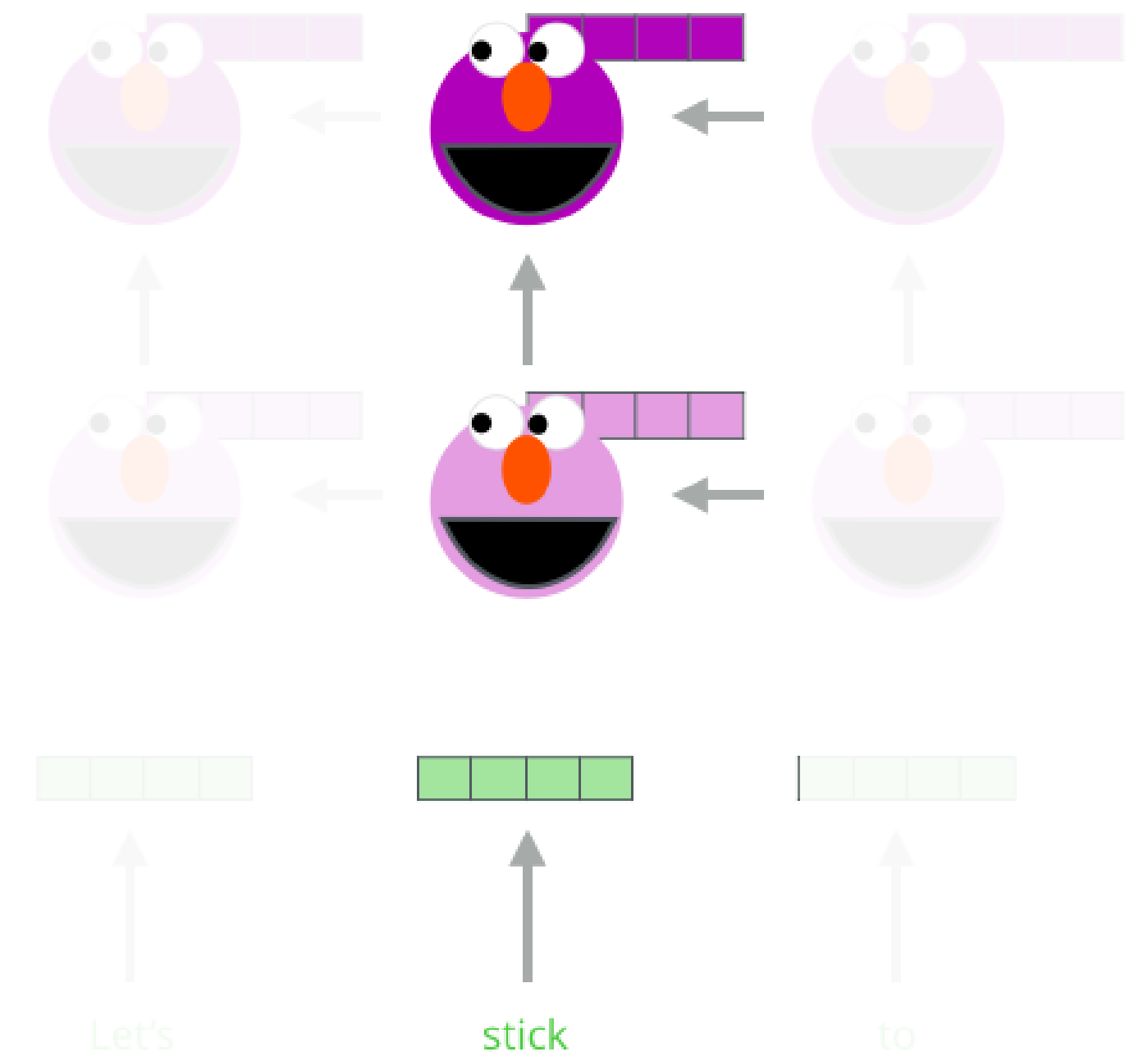


ELMo embedding of “stick” for this task in this context

Forward Language Model



Backward Language Model





ELMo on Name Entity Recognition

Model	Description	CONLL 2003 F1
Klein+, 2003	MEMM softmax markov model	86.07
Florian+, 2003	Linear/softmax/TBL/HMM	88.76
Finkel+, 2005	Categorical feature CRF	86.86
Ratinov and Roth, 2009	CRF+Wiki+Word cls	90.80
Peters+, 2017	BLSTM + char CNN + CRF	90.87
Ma and Hovy, 2016	BLSTM + char CNN + CRF	91.21
TagLM (Peters+, 2017)	LSTM BiLM in BLSTM Tagger	91.93
ELMo (Peters+, 2018)	ELMo in BLSTM	92.22



Peters et al., “Deep Contextualized Word Representations”, in *NAACL-HLT*, 2018.





ELMo Results

- Improvement on various NLP tasks

	TASK	PREVIOUS SOTA		OUR BASELINE	ELMo + BASELINE	INCREASE (ABSOLUTE/ RELATIVE)
Machine Comprehension	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 Labeling	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%
Name 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%

Good transfer learning in NLP (similar to computer vision)





ELMo Analysis

- Word embeddings v.s. contextualized embeddings

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular <u>play</u> 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 <u>play</u> .
	Olivia De Havilland signed to do a Broadway <u>play</u> for Garson {...}	{...} they were actors who had been handed fat roles in a successful <u>play</u> , and had talent enough to fill the roles competently , with nice understatement .

The biLM is able to disambiguate both the PoS and word sense in the source sentence





ELMo Analysis

- The two NLM layers have differentiated uses/meanings
 - ✓ Lower layer is better for lower-level **syntax**, etc. (e.g. Part-of-speech tagging, syntactic dependencies, NER)
 - ✓ Higher layer is better for higher-level **semantics** (e.g. sentiment, semantic role labeling, question answering, SNLI)

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

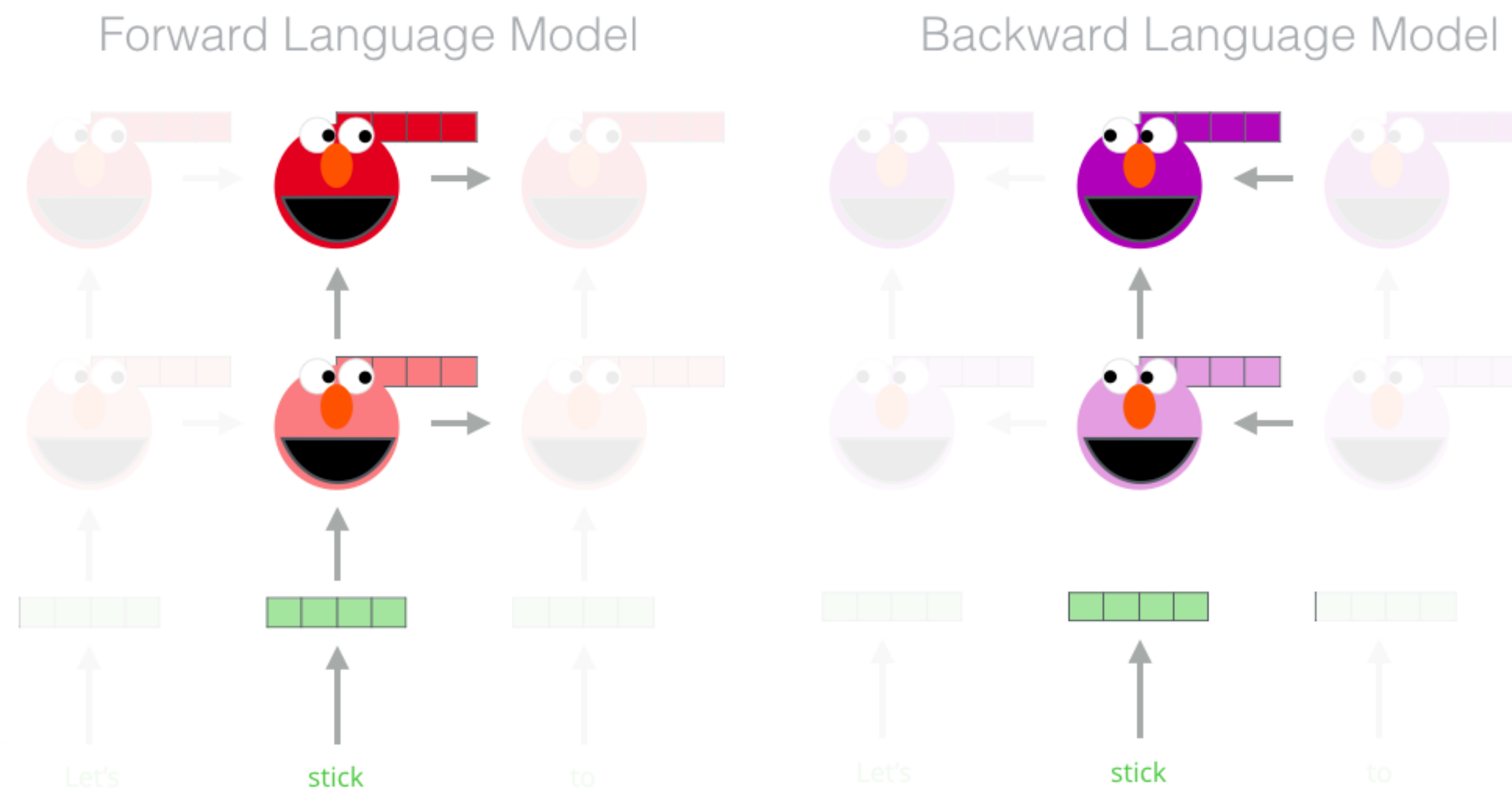
Word Sense Disambiguation

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

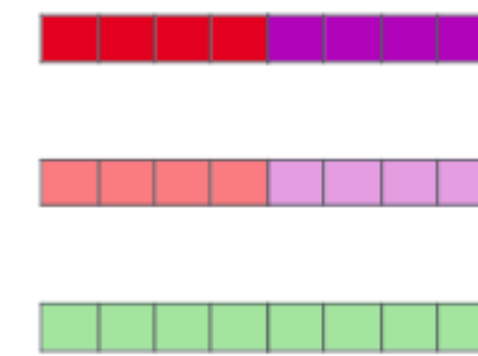


Concluding Remarks

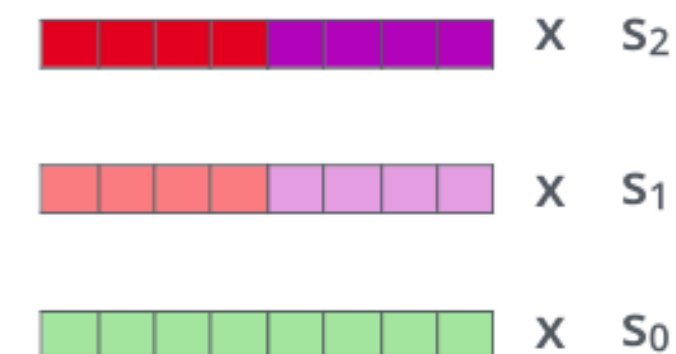
- Contextualized embeddings learned from LM provide informative cues
- ELMo – a general approach for learning high-quality deep context-dependent representations from biLMs
 - ✓ Pre-trained ELMo: <https://allennlp.org/elmo>
 - ✓ ELMo can process the character-level inputs



1- Concatenate hidden layers



2- Multiply each vector by a weight based on the task



3- Sum the (now weighted) vectors

