Encoding Syntactic Knowledge in Neural Networks for Sentiment Classification

Minlie Huang, Qiao Qian, Xiaoyan Zhu

Tsinghua University

http://coai.cs.tsinghua.edu.cn/hml

TOIS paper presented at SIGIR 2018

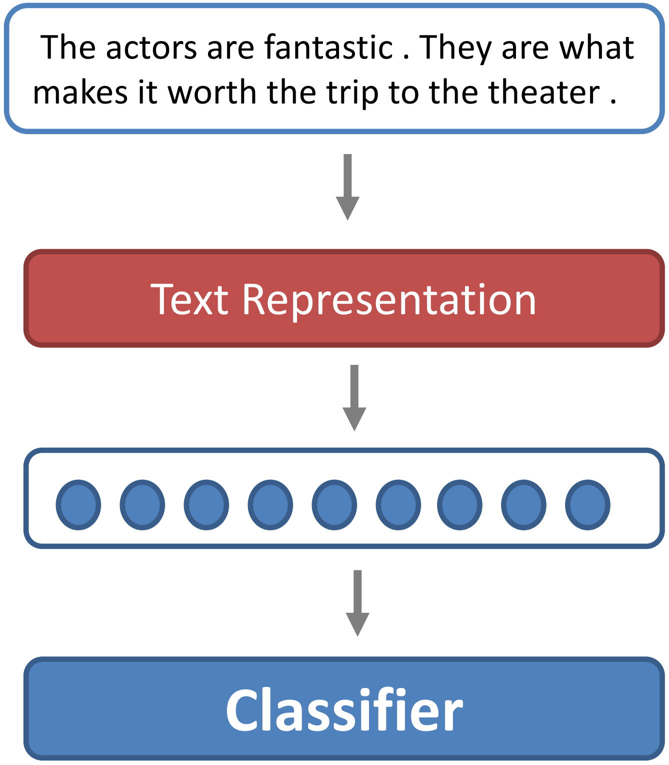
Outline

- Problem & Motivation
- Syntactic Knowledge in Recursive Autoencoders
- Syntactic Knowledge in Tree-structured LSTM
- Linguistically Regularized LSTM
- Summary

Minlie Huang, Qiao Qian, Xiaoyan Zhu. Encoding Syntactic Knowledge in Neural Networks for Sentiment Classification. **ACM Trans. Inf. Syst**. 35, 3, Article 26 (June 2017), 27 pages.

Motivation

- **Non-structure model**
  - Sequence model: CNN, RNN, LSTM
  - Bag-of-words models (BM、AE)
- **Using parsing structures**
  - Recursive autoencoders
  - Tree-structured LSTM
- **Auto-learned structure**
  - Binary tree, overly deep (Yogatama et al., 2017)
  - Hierarchical structure (Chung, et al., 2017; Zhang et al., 2018)



The actors are fantastic . They are what makes it worth the trip to the theater .

Text Representation

Classifier

Motivation

- Text representation is **fundamental** for downstream tasks
- Research problem: does **syntactic (linguistic) knowledge** help sentiment classification?
  - **Part-of-speech tags**: nouns, verbs, adverbs
  - **Lexicons**: sentiment words (awesome, interesting), negators (not, never), and intensifiers (very, quite)

This is **not** a/dt **very/adv** **interesting/adj** movie/nn.

Recursive Autoencoders

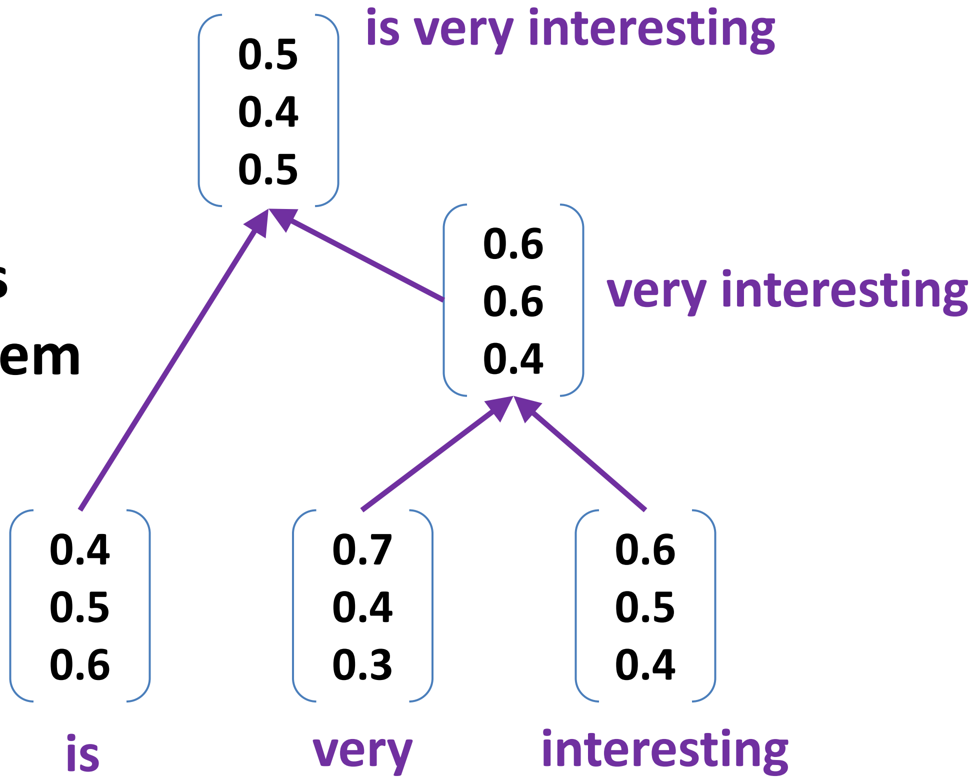
Rules of Compositionality

The meaning (vector) of a sentence is determined by

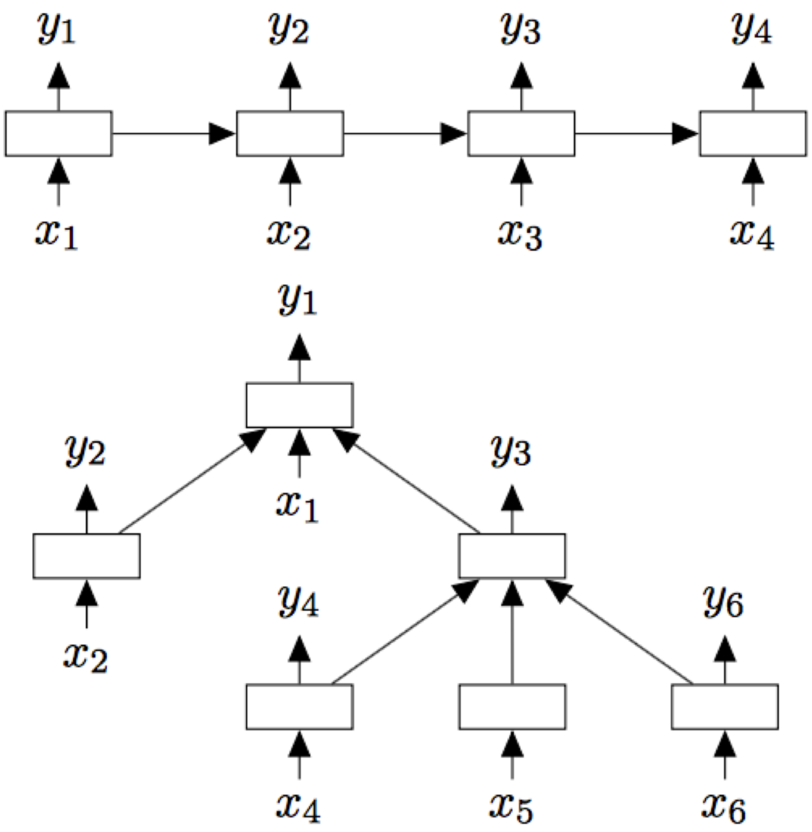
(1) **The meanings of its words**

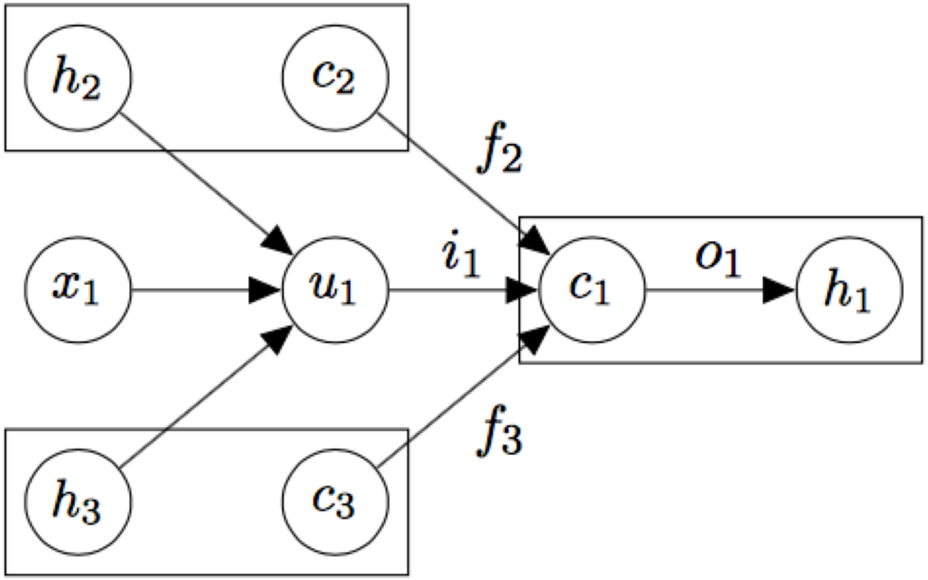
(2) **The rules that combine them**

Socher et al., 2011b;
Socher et al., 2012;
Socher et al., 2013b;



Tree-structured LSTM

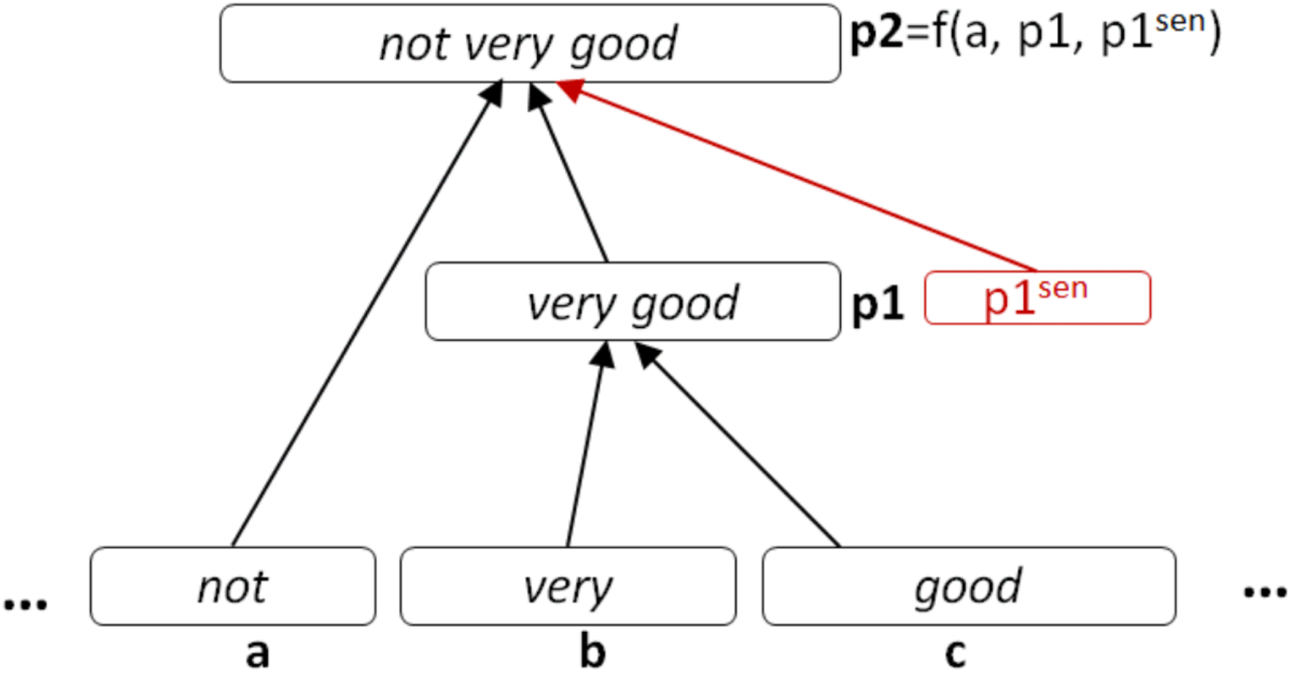




x: input word
h: hidden state
c: memory state
i,f,o: input, forget, output gates, resp.

Tai et al., 2015
Zhu et al., 2015

Negation Effect in Sent. Class.

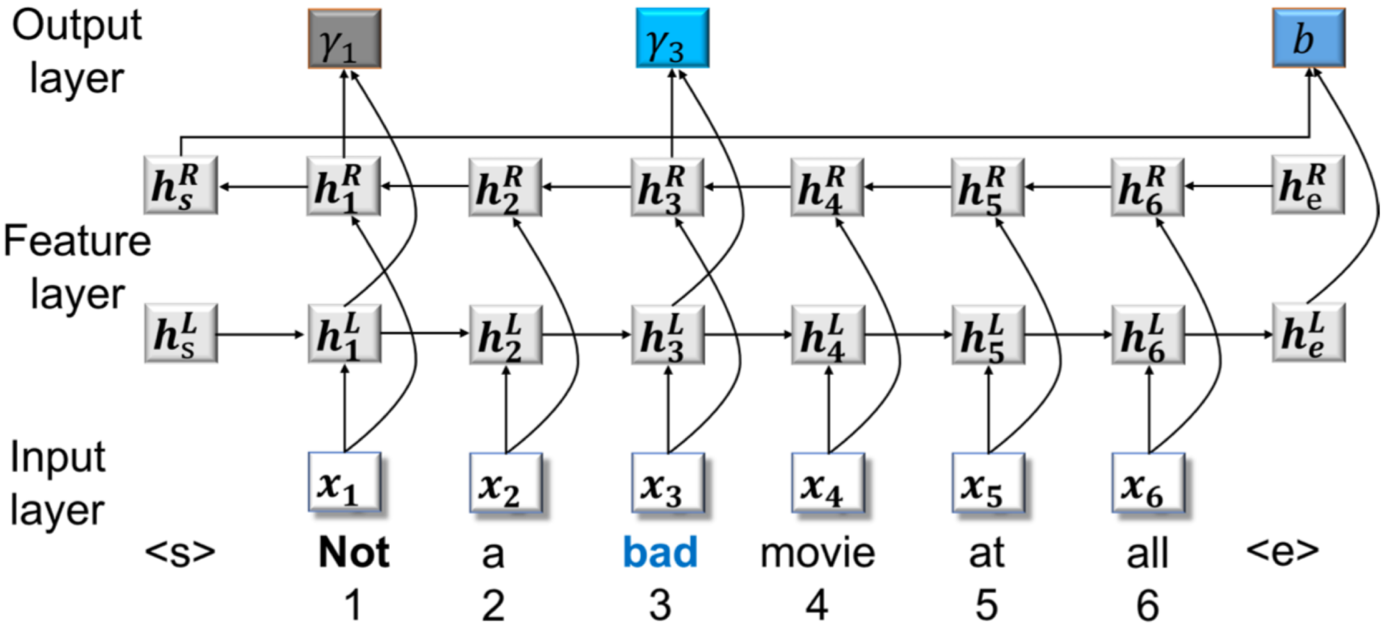


Negation effect depends on the negator, the modified text, and its sentiment

Zhu et al., 2014. An empirical study on the effect of negation words on sentiment. In *ACL*. pages 304–313.

Neural Weighing Schema

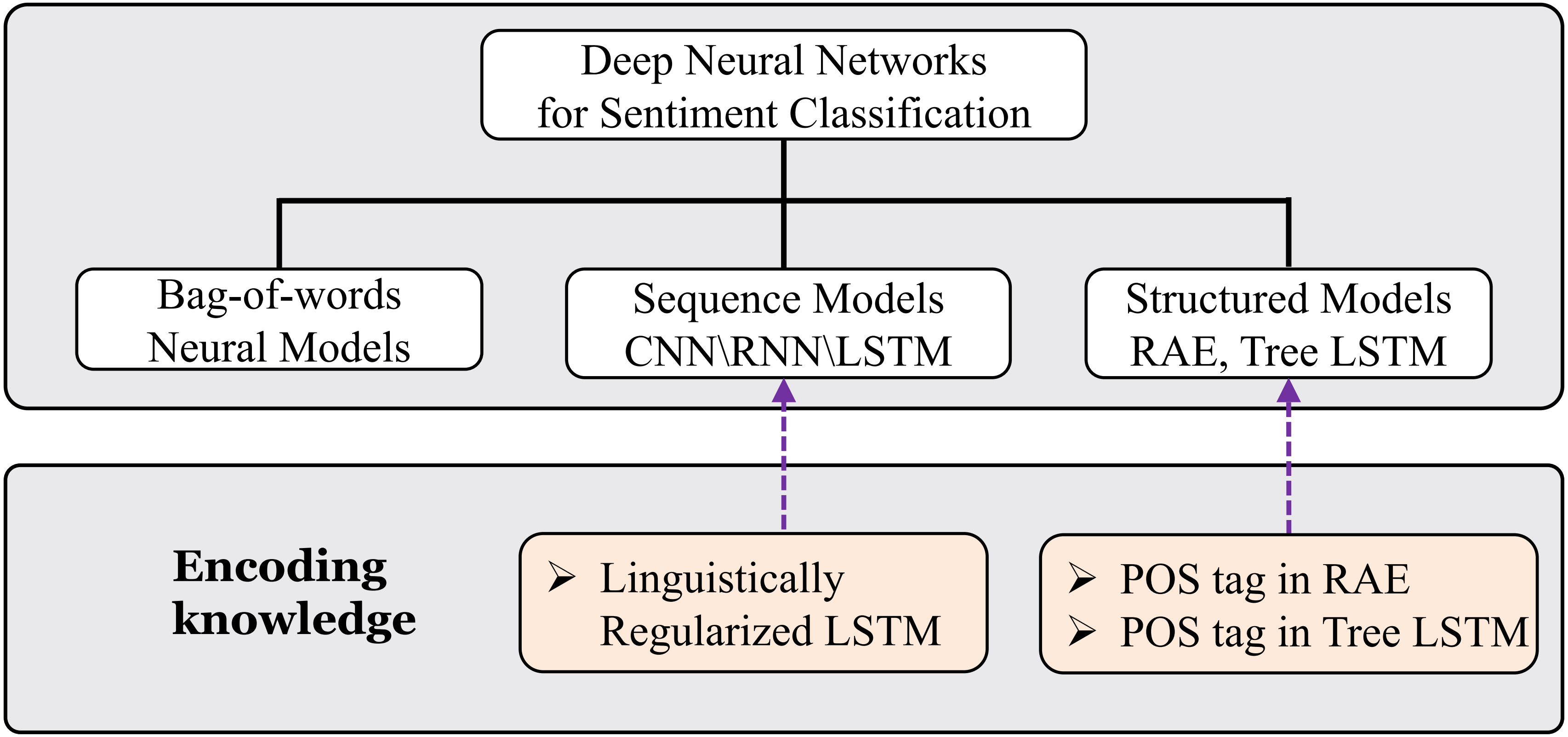
$$Score = \gamma_1 \times score(not) + \gamma_3 \times score(bad) + b$$



Sentence sentiment score= weighted sum of its sentiment words and negators.

Teng et al. EMNLP 2016. Context-sensitive lexicon features for neural sentiment analysis.

Our Proposal



Deep Neural Networks for Sentiment Classification

Bag-of-words Neural Models

Sequence Models CNN\RNN\LSTM

Structured Models RAE, Tree LSTM

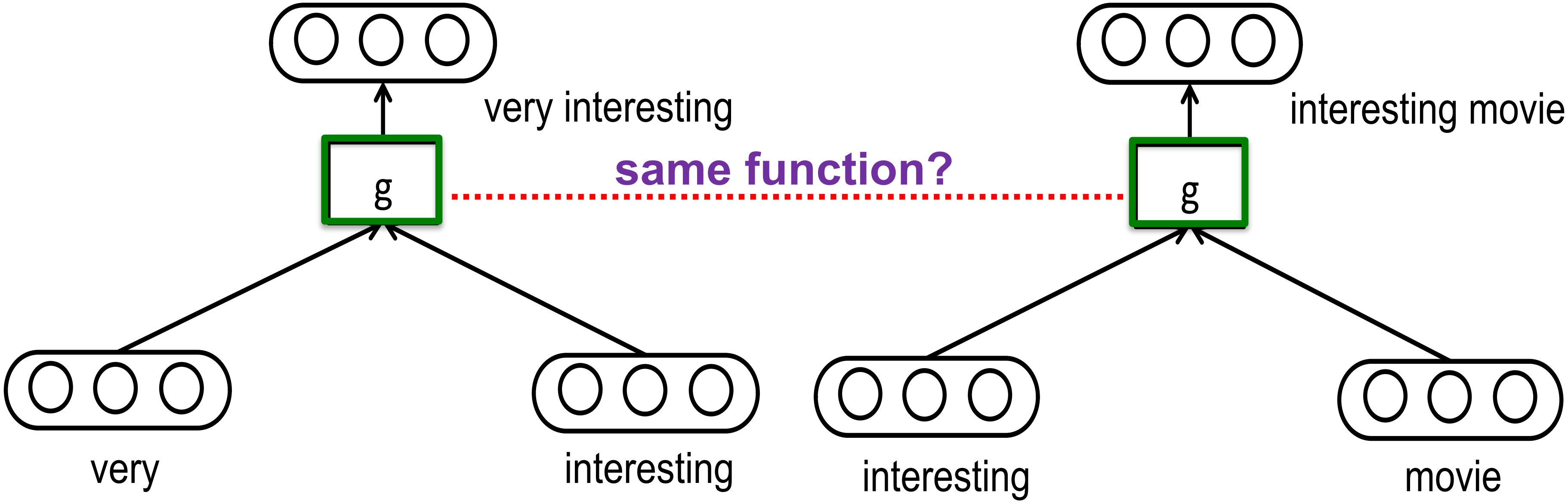
Encoding knowledge

- Linguistically Regularized LSTM

- POS tag in RAE
- POS tag in Tree LSTM

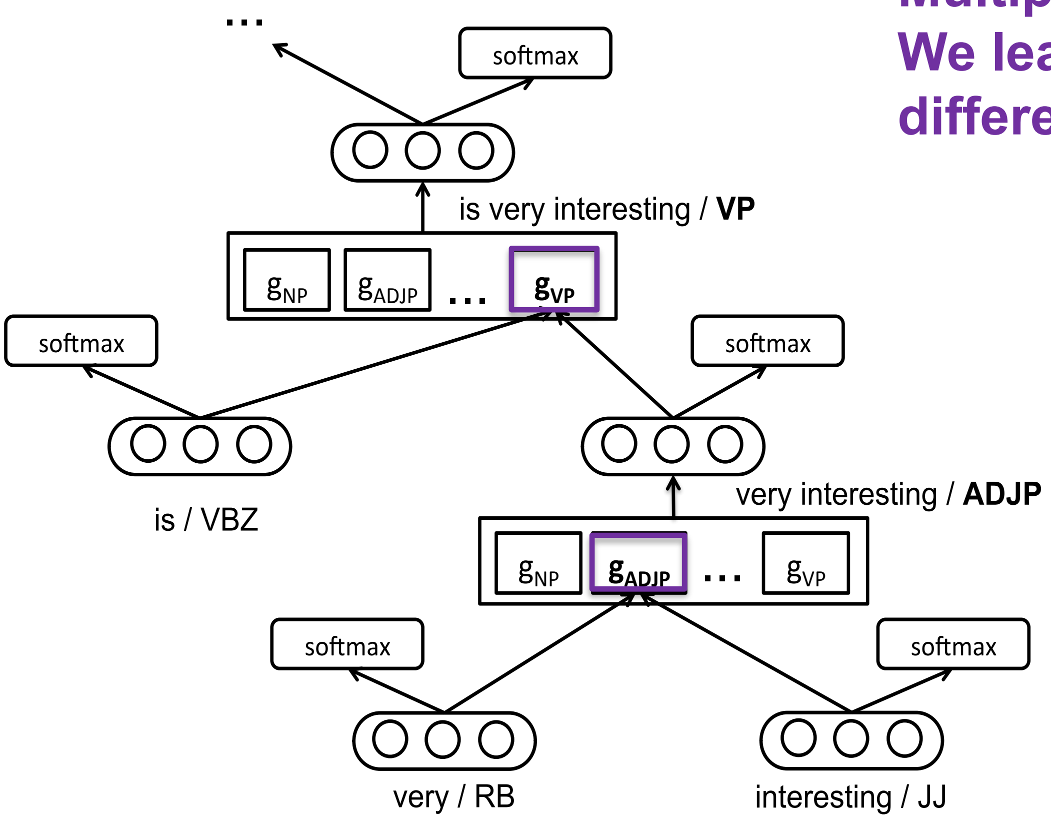
Using POS Tag in Recursive Models

$V_{very\ interesting}=\boldsymbol{g}(V_{very}, V_{interesting})$ $V_{interesting\ movie}=\boldsymbol{g}(V_{interesting}, V_{movie})$



Noun phrase vs. adjective phrase

Tag-Guided Recursive Model (TGRNN)

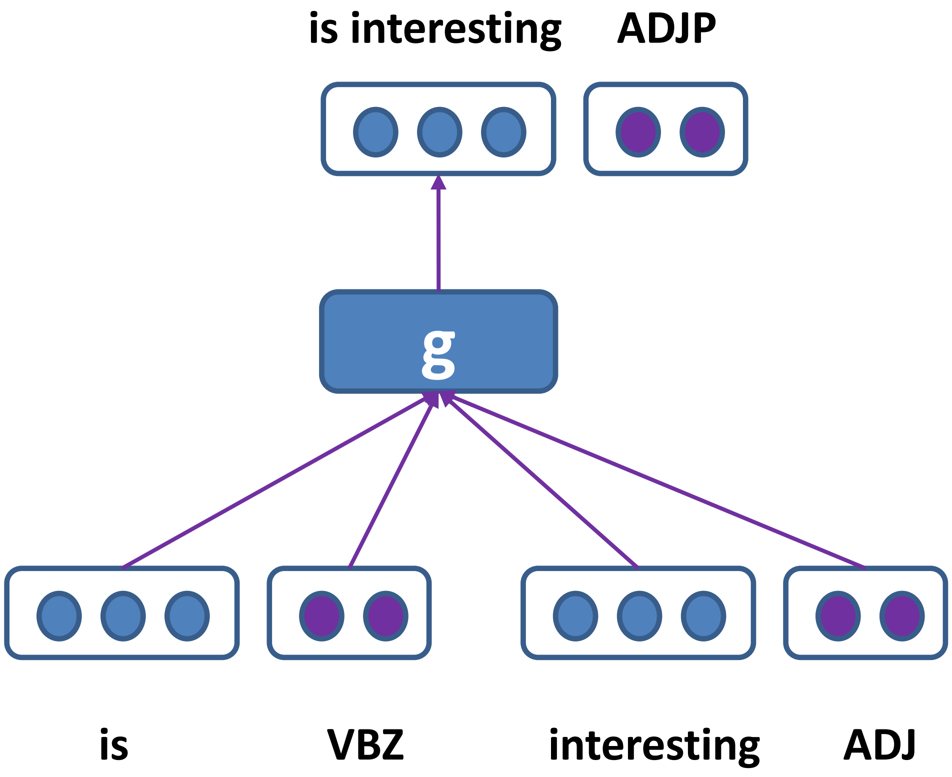


Multiple composition functions: We learn different functions for different POS tags.

$$g_{p_t}(h_t^l, h_t^r) = W_{p_t} \begin{bmatrix} h_t^l \\ h_t^r \end{bmatrix} + b_{p_t}$$

Limitation:
too many composition functions!

Tag-Embedded Recursive Model (TE-RNN)

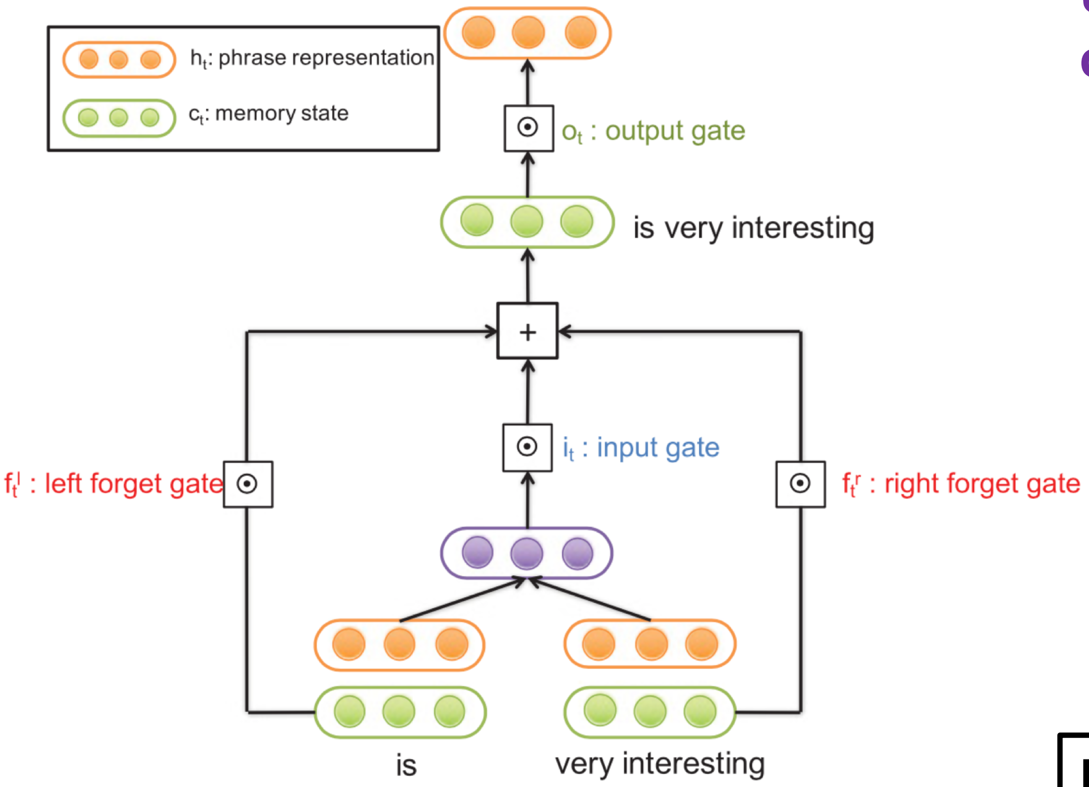


tag vector: syntax knowledge

phrase vector: semantic info.

$$g(v_i^l, e_{t_i^l}, v_i^r, e_{t_i^r}) = W \begin{bmatrix} v_i^l \\ e_{t_i^l} \\ v_i^r \\ e_{t_i^r} \end{bmatrix} + b$$

Tag Weighted LSTM (TW-LSTM)



Use the pos-tag to directly control the gates in LSTM

$$i_j = \sigma(W_i[t_j]),$$

$$f_j^l = \sigma(W_f[t_j^l]),$$

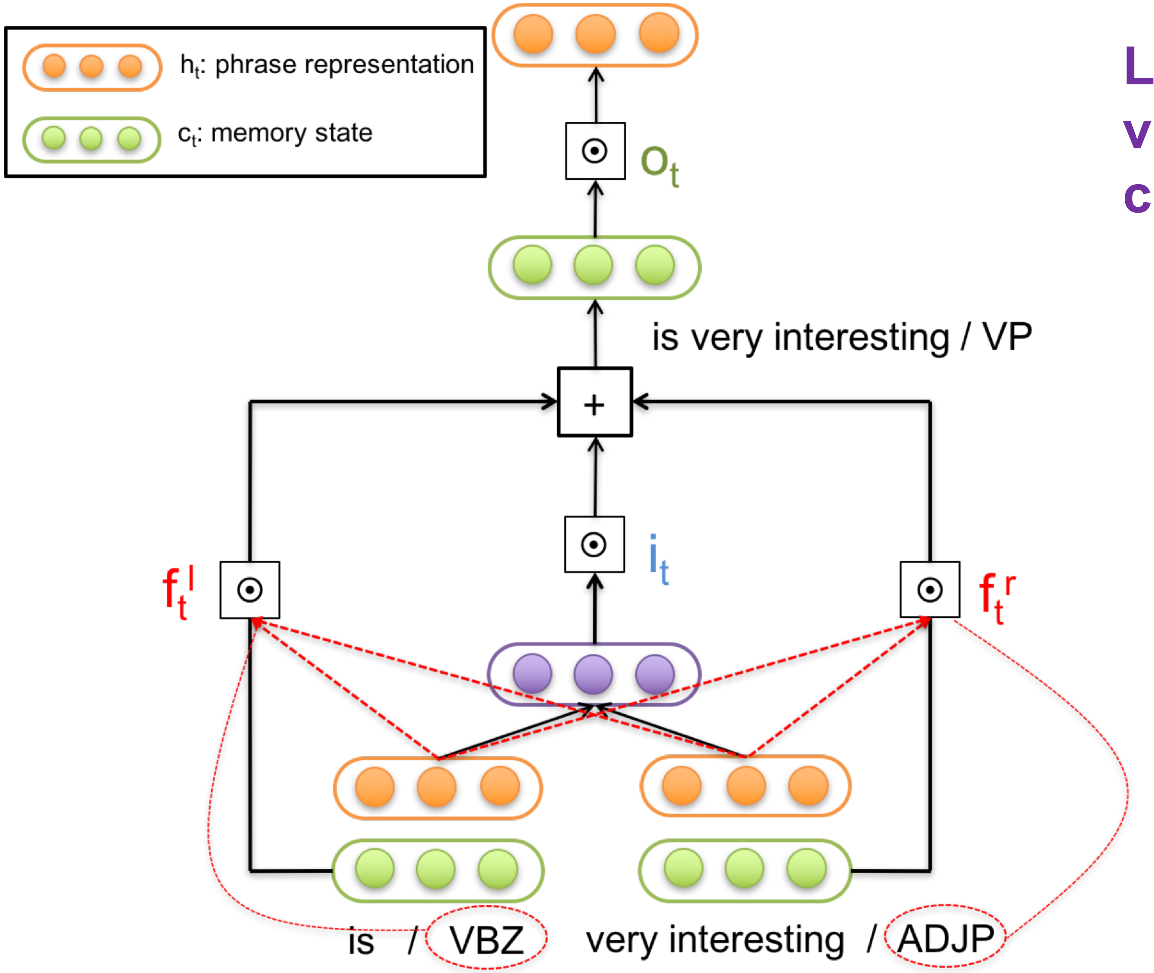
$$f_j^r = \sigma(W_f[t_j^r]),$$

$$o_j = \sigma(W_o[t_j]),$$

W_x: weight-tag matrix

Limitation:
The word information is totally ignored.

Tag Embedded LSTM (TE-LSTM)



Let tag embeddings and word vectors both participate in the control of LSTM gates

$$i_j = \sigma\left(\alpha \cdot U_i E[t_j] + (1-\alpha) \cdot S_i \begin{bmatrix} h_j^l \\ h_j^r \end{bmatrix}\right),$$

$$f_j^l = \sigma\left(\alpha \cdot U_f \begin{bmatrix} E[t_j] \\ E[t_j^l] \end{bmatrix} + (1-\alpha) \cdot S_f^l \begin{bmatrix} h_j^l \\ h_j^r \end{bmatrix}\right),$$

$$f_j^r = \sigma\left(\alpha \cdot U_f \begin{bmatrix} E[t_j] \\ E[t_j^r] \end{bmatrix} + (1-\alpha) \cdot S_f^r \begin{bmatrix} h_j^l \\ h_j^r \end{bmatrix}\right),$$

$$o_j = \sigma\left(\alpha \cdot U_o E[t_j] + (1-\alpha) \cdot S_o \begin{bmatrix} h_j^l \\ h_j^r \end{bmatrix}\right),$$

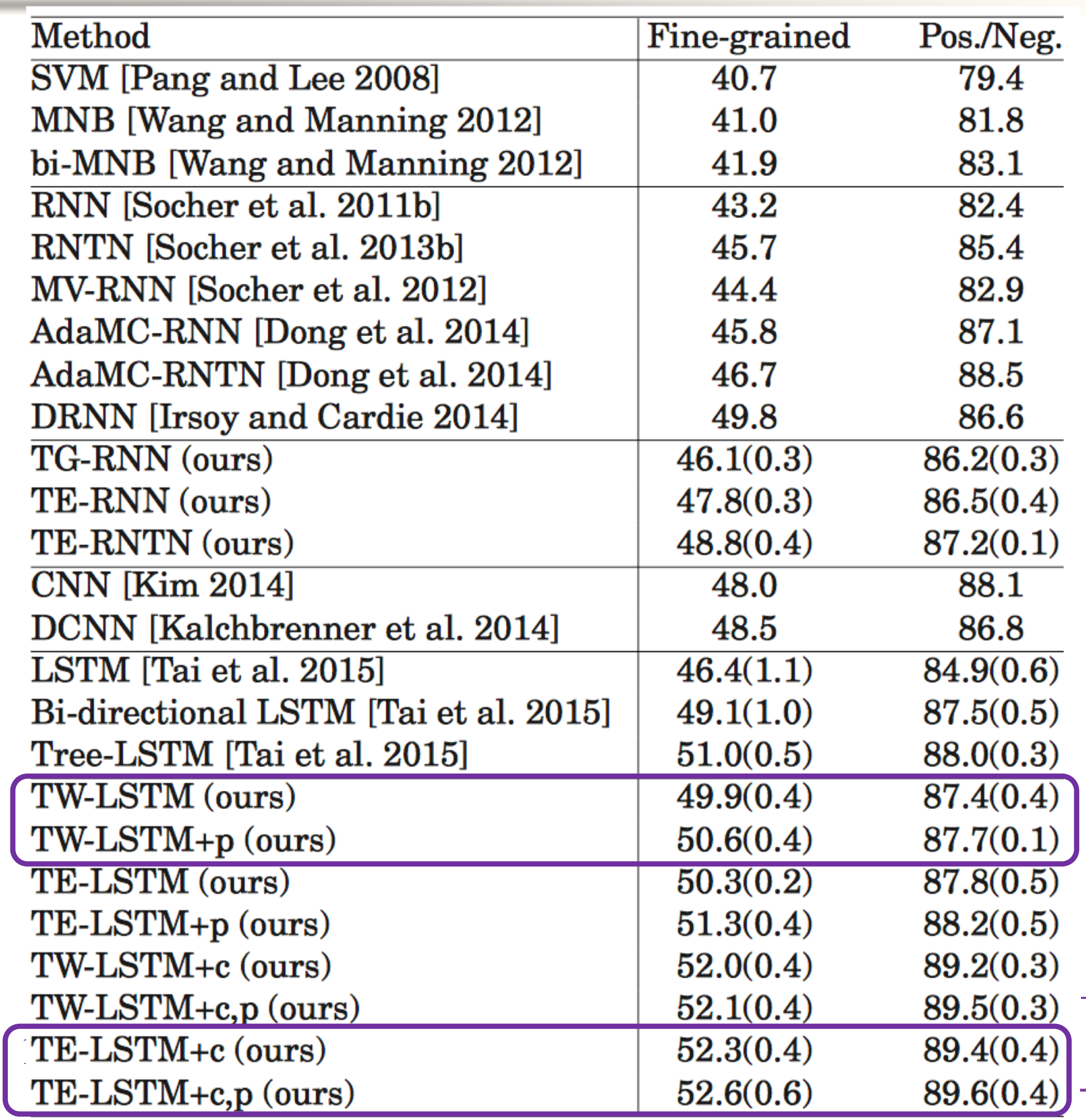
Experiment & Evaluation

- Datasets

Dataset	Movie Review (MR)	Stanford Sentiment Treebank (SST)
Task	pos. / neg.	fine-grained
Sentences	10,662	11,885
Label	sentence-level	sentence-level & phrase-level
Evaluation	10-cross-validation	train:valid:test=7:1:2

- Baselines:
  - Recursive models: RNN\RNTN\DRNN\MC-RNN
  - LSTM: Bi-LSTM\Tree-LSTM
  - CNN:CNN\DCNN

Accuracy on SST



Method	Fine-grained	Pos./Neg.
SVM [Pang and Lee 2008]	40.7	79.4
MNB [Wang and Manning 2012]	41.0	81.8
bi-MNB [Wang and Manning 2012]	41.9	83.1
RNN [Socher et al. 2011b]	43.2	82.4
RNTN [Socher et al. 2013b]	45.7	85.4
MV-RNN [Socher et al. 2012]	44.4	82.9
AdaMC-RNN [Dong et al. 2014]	45.8	87.1
AdaMC-RNTN [Dong et al. 2014]	46.7	88.5
DRNN [Irsoy and Cardie 2014]	49.8	86.6
TG-RNN (ours)	46.1(0.3)	86.2(0.3)
TE-RNN (ours)	47.8(0.3)	86.5(0.4)
TE-RNTN (ours)	48.8(0.4)	87.2(0.1)
CNN [Kim 2014]	48.0	88.1
DCNN [Kalchbrenner et al. 2014]	48.5	86.8
LSTM [Tai et al. 2015]	46.4(1.1)	84.9(0.6)
Bi-directional LSTM [Tai et al. 2015]	49.1(1.0)	87.5(0.5)
Tree-LSTM [Tai et al. 2015]	51.0(0.5)	88.0(0.3)
TW-LSTM (ours)	49.9(0.4)	87.4(0.4)
TW-LSTM+p (ours)	50.6(0.4)	87.7(0.1)
TE-LSTM (ours)	50.3(0.2)	87.8(0.5)
TE-LSTM+p (ours)	51.3(0.4)	88.2(0.5)
TW-LSTM+c (ours)	52.0(0.4)	89.2(0.3)
TW-LSTM+c,p (ours)	52.1(0.4)	89.5(0.3)
TE-LSTM+c (ours)	52.3(0.4)	89.4(0.4)
TE-LSTM+c,p (ours)	52.6(0.6)	89.6(0.4)

c: combining tag embeddings and word vectors

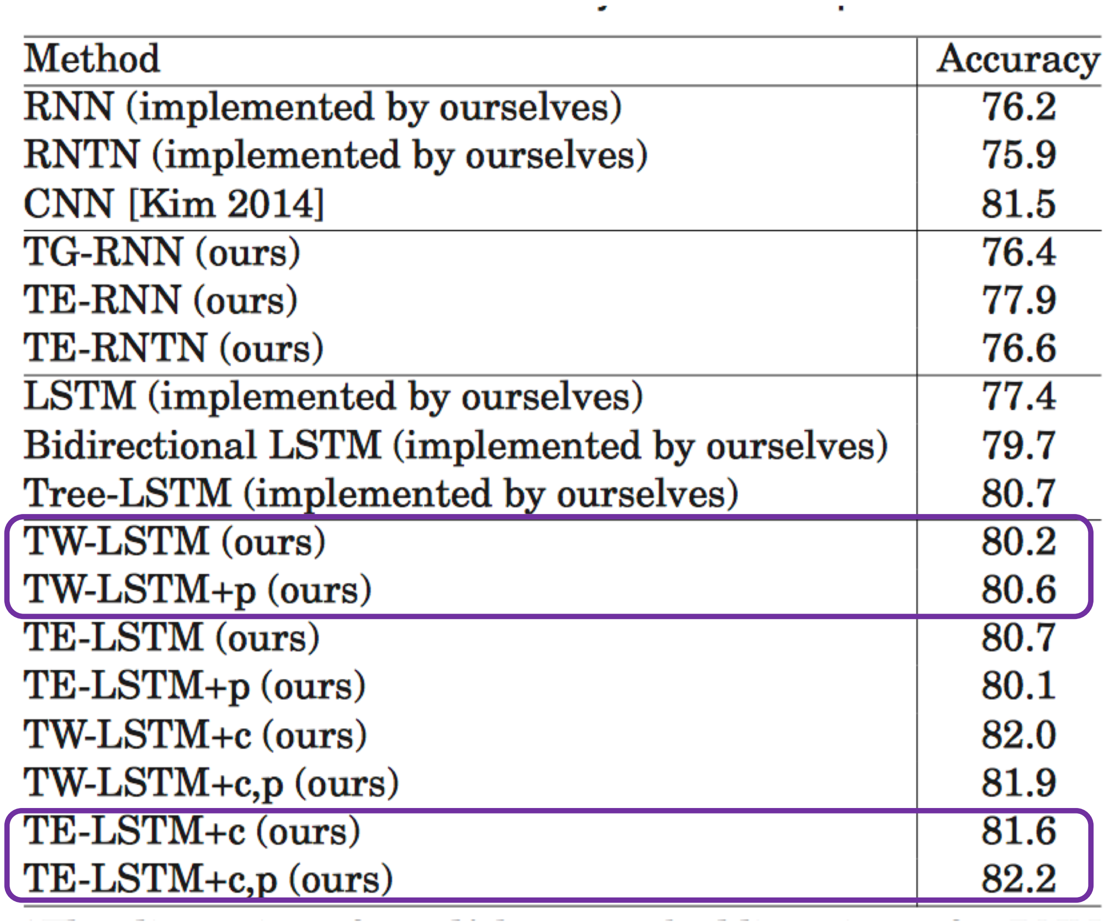
p: considering child-parent association

Low-dimension word vectors with d=25

Only using POS Tag to control LSTM gates can still produce competitive results

High-dimension word vectors with d=300

Accuracy on MR (Movie Review)

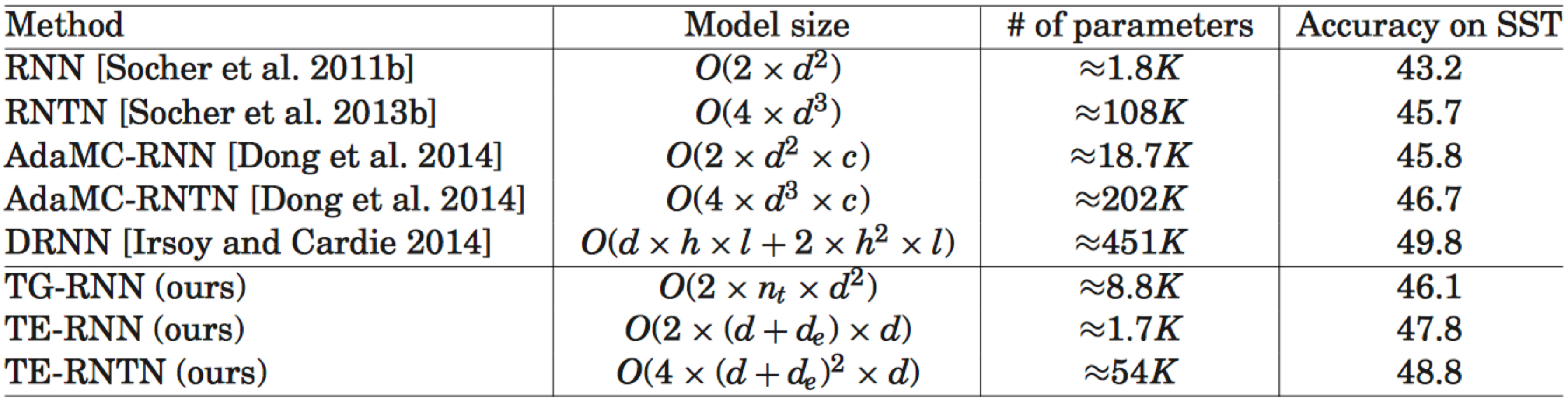


Method	Accuracy
RNN (implemented by ourselves)	76.2
RNTN (implemented by ourselves)	75.9
CNN [Kim 2014]	81.5
TG-RNN (ours)	76.4
TE-RNN (ours)	77.9
TE-RNTN (ours)	76.6
LSTM (implemented by ourselves)	77.4
Bidirectional LSTM (implemented by ourselves)	79.7
Tree-LSTM (implemented by ourselves)	80.7
TW-LSTM (ours)	80.2
TW-LSTM+p (ours)	80.6
TE-LSTM (ours)	80.7
TE-LSTM+p (ours)	80.1
TW-LSTM+c (ours)	82.0
TW-LSTM+c,p (ours)	81.9
TE-LSTM+c (ours)	81.6
TE-LSTM+c,p (ours)	82.2

Only using POS Tag to control LSTM gates can still produce competitive results

Model Complexity I

- Complexity Analysis for RNN models

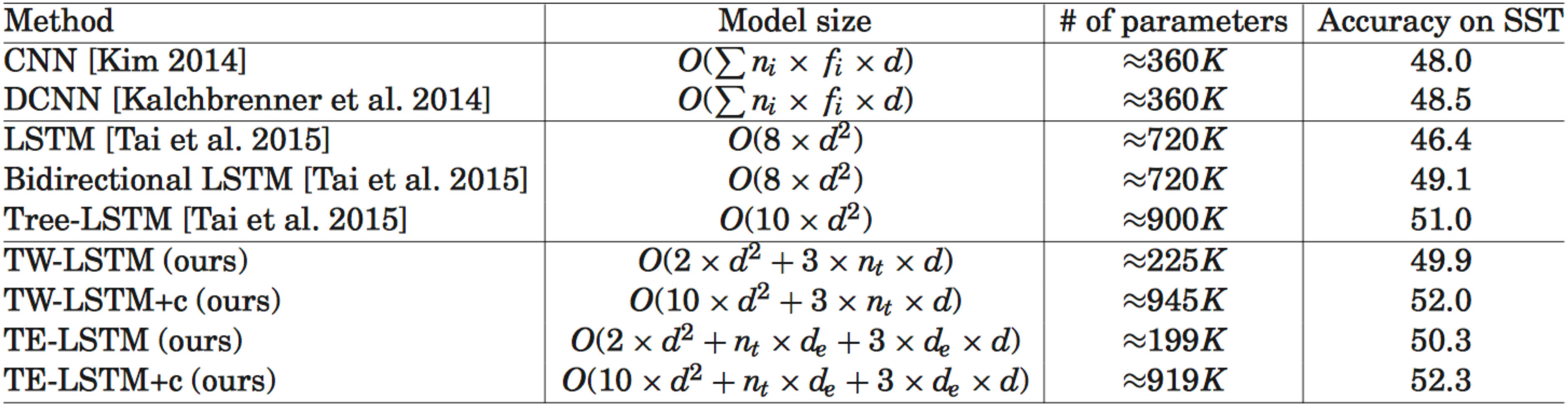


Method	Model size	# of parameters	Accuracy on SST
RNN [Socher et al. 2011b]	$O(2 \times d^2)$	$\approx 1.8K$	43.2
RNTN [Socher et al. 2013b]	$O(4 \times d^3)$	$\approx 108K$	45.7
AdaMC-RNN [Dong et al. 2014]	$O(2 \times d^2 \times c)$	$\approx 18.7K$	45.8
AdaMC-RNTN [Dong et al. 2014]	$O(4 \times d^3 \times c)$	$\approx 202K$	46.7
DRNN [Irsoy and Cardie 2014]	$O(d \times h \times l + 2 \times h^2 \times l)$	$\approx 451K$	49.8
TG-RNN (ours)	$O(2 \times n_t \times d^2)$	$\approx 8.8K$	46.1
TE-RNN (ours)	$O(2 \times (d + d_e) \times d)$	$\approx 1.7K$	47.8
TE-RNTN (ours)	$O(4 \times (d + d_e)^2 \times d)$	$\approx 54K$	48.8

- **d**: the dimension for word vectors;
- **d_e**: the dimension for tag embedding;
- **c**: the number of composition function;
- **n_t**: the number of frequency tags.

Model Complexity II

- Complexity Analysis for LSTM models



Method	Model size	# of parameters	Accuracy on SST
CNN [Kim 2014]	$O(\sum n_i \times f_i \times d)$	$\approx 360K$	48.0
DCNN [Kalchbrenner et al. 2014]	$O(\sum n_i \times f_i \times d)$	$\approx 360K$	48.5
LSTM [Tai et al. 2015]	$O(8 \times d^2)$	$\approx 720K$	46.4
Bidirectional LSTM [Tai et al. 2015]	$O(8 \times d^2)$	$\approx 720K$	49.1
Tree-LSTM [Tai et al. 2015]	$O(10 \times d^2)$	$\approx 900K$	51.0
TW-LSTM (ours)	$O(2 \times d^2 + 3 \times n_t \times d)$	$\approx 225K$	49.9
TW-LSTM+c (ours)	$O(10 \times d^2 + 3 \times n_t \times d)$	$\approx 945K$	52.0
TE-LSTM (ours)	$O(2 \times d^2 + n_t \times d_e + 3 \times d_e \times d)$	$\approx 199K$	50.3
TE-LSTM+c (ours)	$O(10 \times d^2 + n_t \times d_e + 3 \times d_e \times d)$	$\approx 919K$	52.3

- **d**: the dimension for word vectors;
- **$\mathbf{d_e}$**: the dimension for tag embedding;
- **$\mathbf{n_t}$**: the number of frequency tags.

Tag Embedding Analysis I

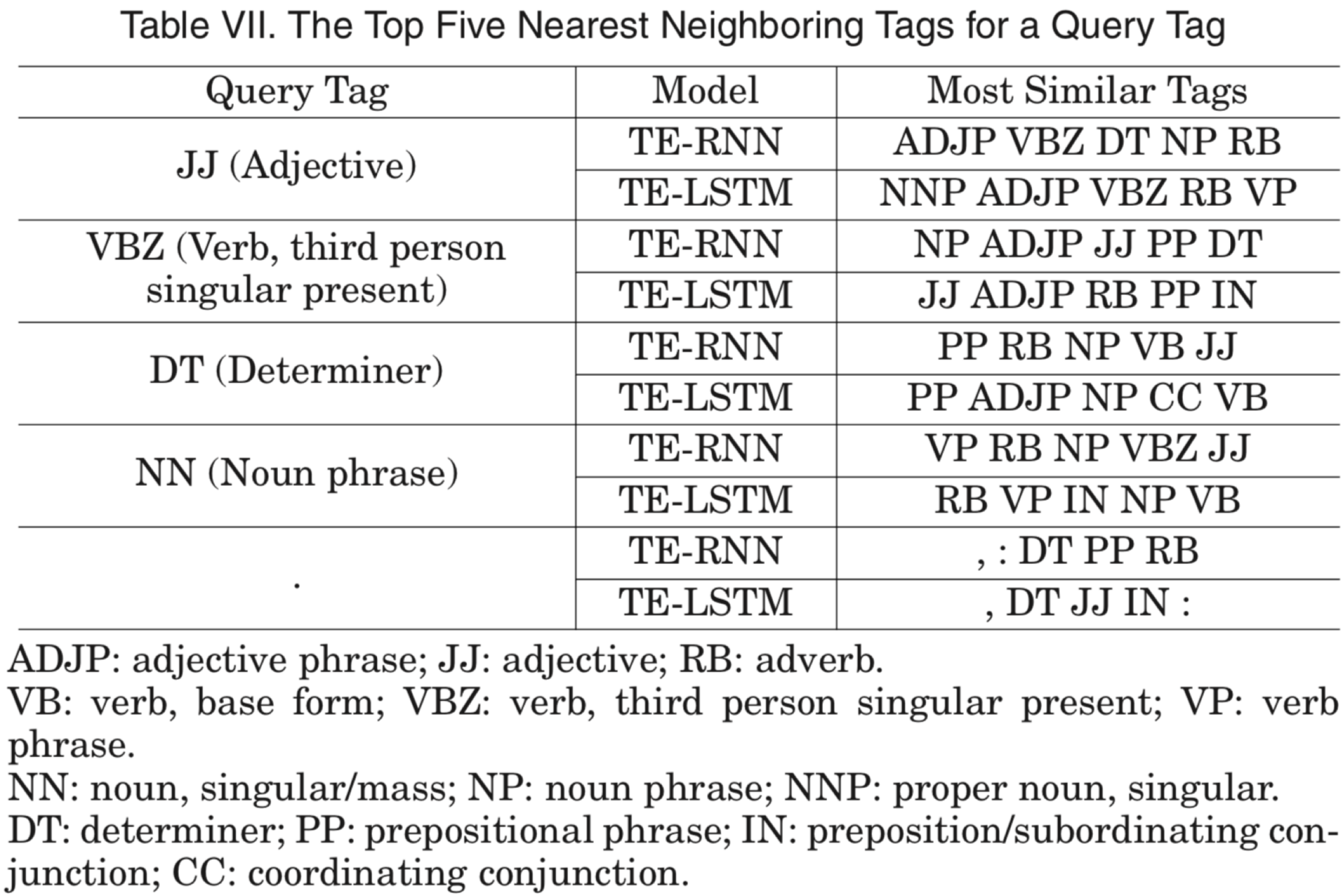


Table VII. The Top Five Nearest Neighboring Tags for a Query Tag

Query Tag	Model	Most Similar Tags
JJ (Adjective)	TE-RNN	ADJP VBZ DT NP RB
	TE-LSTM	NNP ADJP VBZ RB VP
VBZ (Verb, third person singular present)	TE-RNN	NP ADJP JJ PP DT
	TE-LSTM	JJ ADJP RB PP IN
DT (Determiner)	TE-RNN	PP RB NP VB JJ
	TE-LSTM	PP ADJP NP CC VB
NN (Noun phrase)	TE-RNN	VP RB NP VBZ JJ
	TE-LSTM	RB VP IN NP VB
.	TE-RNN	, : DT PP RB
	TE-LSTM	, DT JJ IN :

ADJP: adjective phrase; JJ: adjective; RB: adverb.
VB: verb, base form; VBZ: verb, third person singular present; VP: verb phrase.
NN: noun, singular/mass; NP: noun phrase; NNP: proper noun, singular.
DT: determiner; PP: prepositional phrase; IN: preposition/subordinating conjunction; CC: coordinating conjunction.

Similar tags are close in the learned embedding space.

Tag Embedding Analysis II

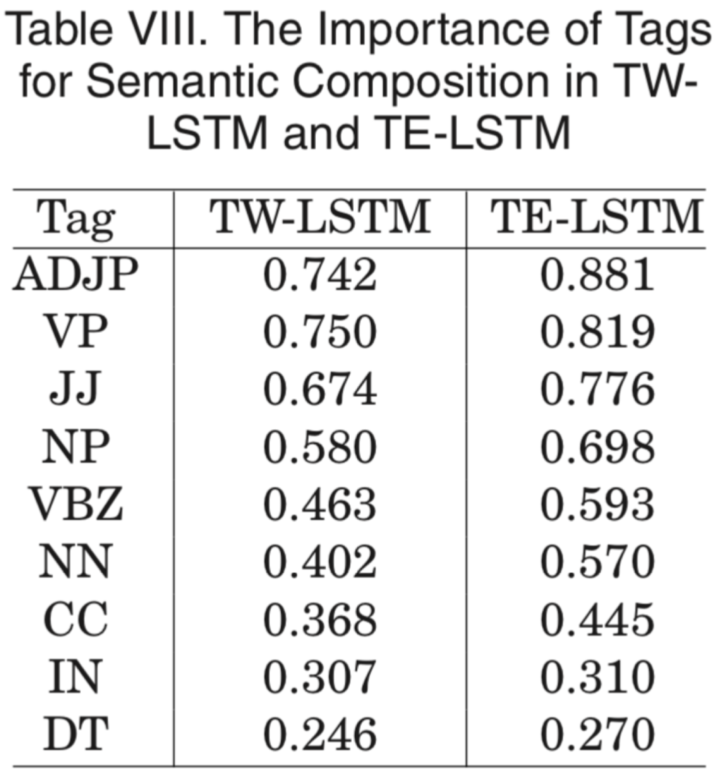


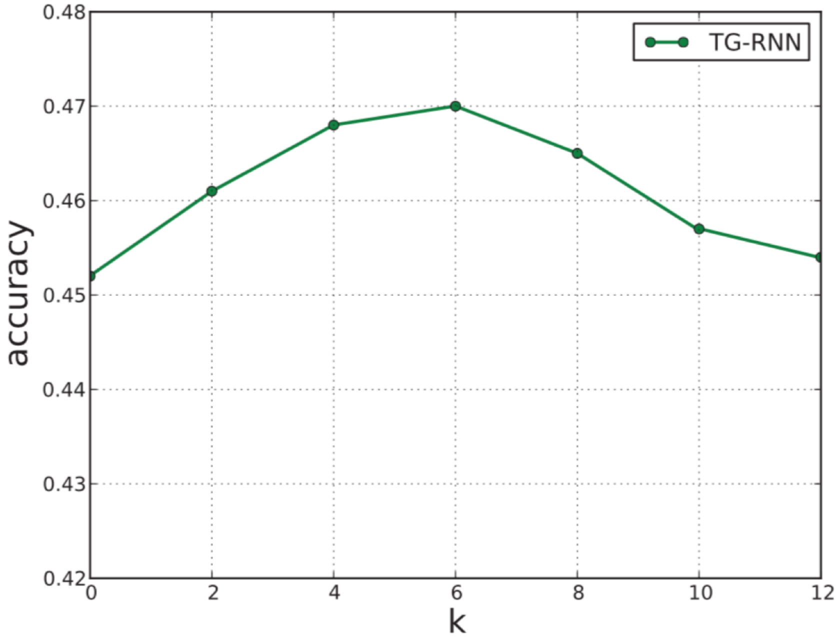
Table VIII. The Importance of Tags for Semantic Composition in TW-LSTM and TE-LSTM

Tag	TW-LSTM	TE-LSTM
ADJP	0.742	0.881
VP	0.750	0.819
JJ	0.674	0.776
NP	0.580	0.698
VBZ	0.463	0.593
NN	0.402	0.570
CC	0.368	0.445
IN	0.307	0.310
DT	0.246	0.270

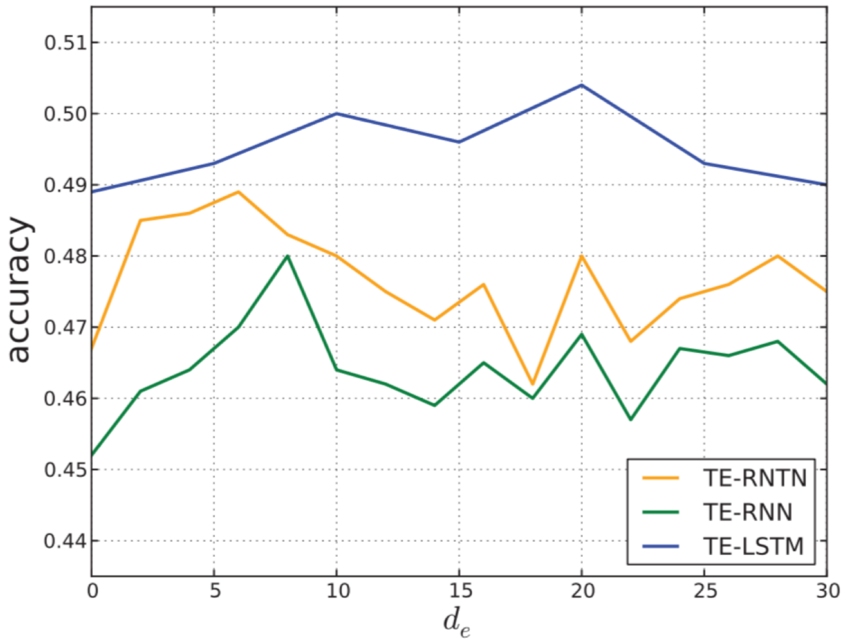
Score: the average of all dimensions of the output of the forget gates

More important tags for sentiment Classification have higher scores

Parameter Tuning



Accuracy curve over the number of composition functions (k)



Accuracy curve over the dimension of tag embeddings

Linguistically Regularized LSTM

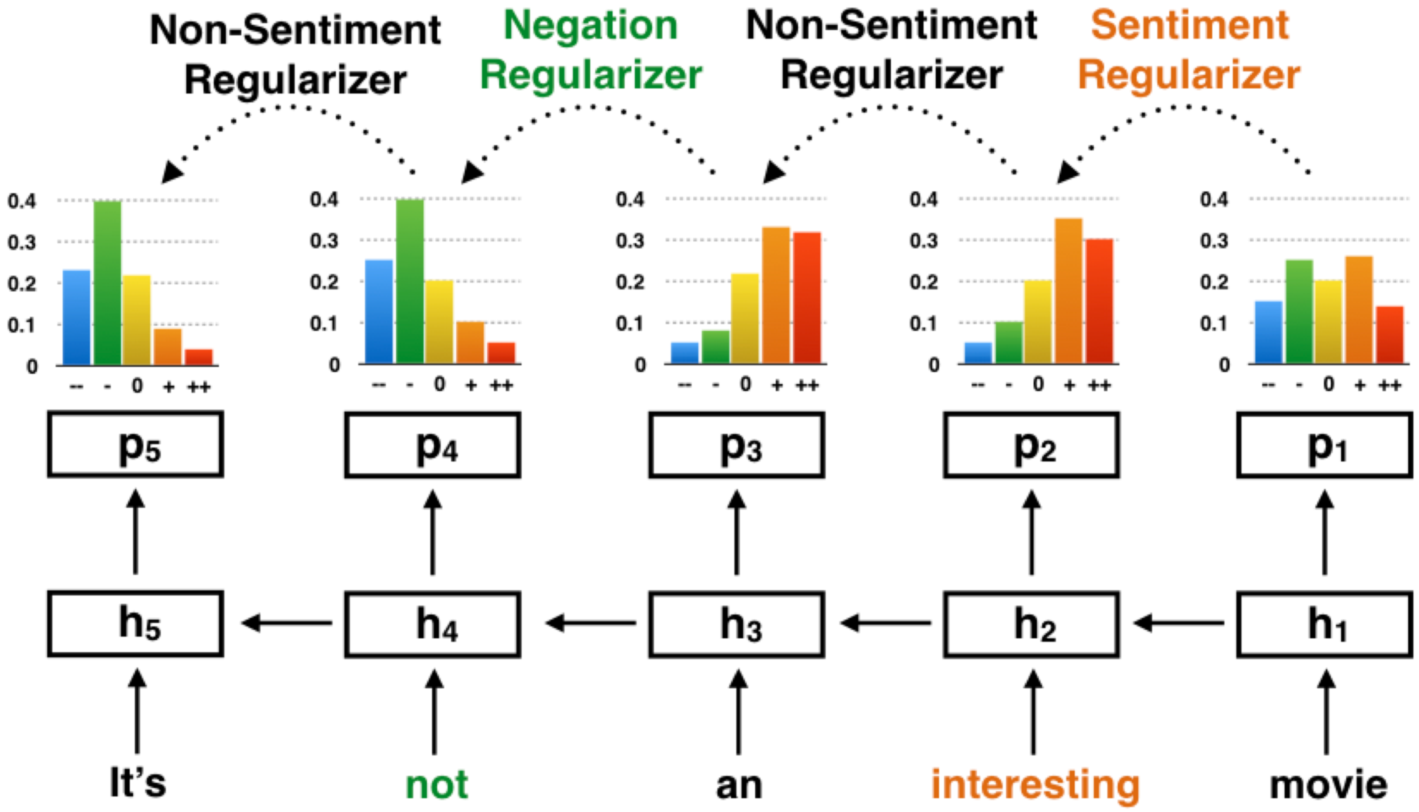
- Linguistic resources for sentiment classification
  - Negator: **not, never, cannot**
  - Intensifier: **very, absolutely**
  - Sentiment resources: sentiment words like **interesting, wonderful, etc**

This is **not** a **very** **interesting** movie.

How to leverage linguistic resources in neural networks?

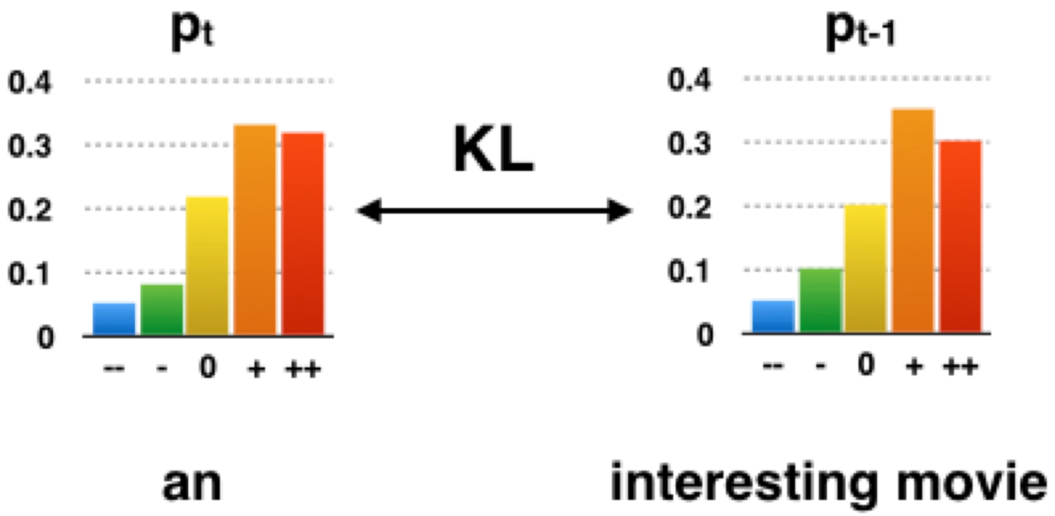
Overview

- Linguistically Regularized LSTM



Non-Sentiment Regularizer

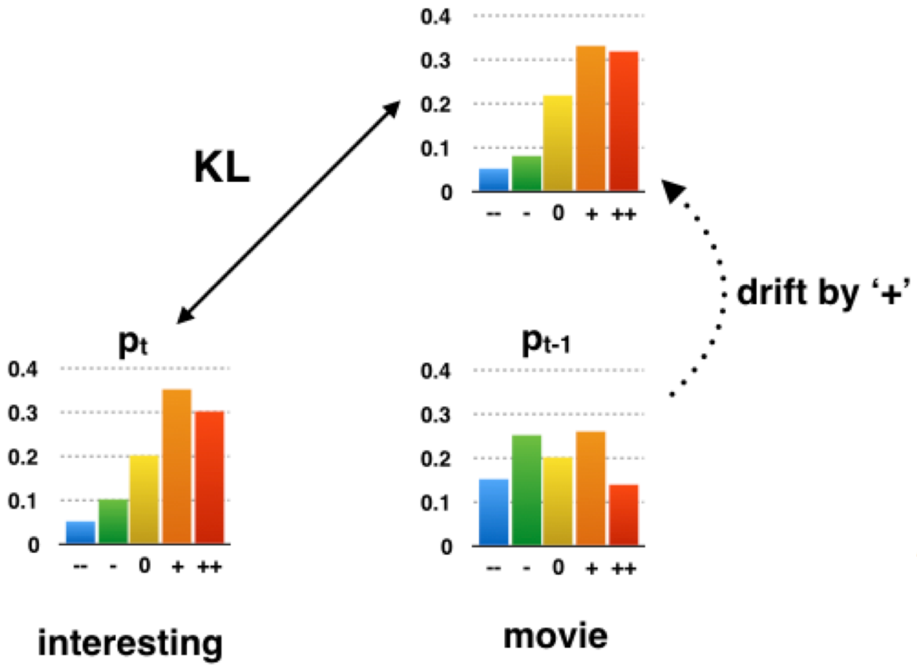
- The sentiment distributions of adjacent positions should be close to each other.



$$L_t^{(NSR)} = max(0, D_{KL}(p_t||p_{t-1}) - M)$$

Sentiment Regularizer

- The sentiment distributions of adjacent positions should drift accordingly.



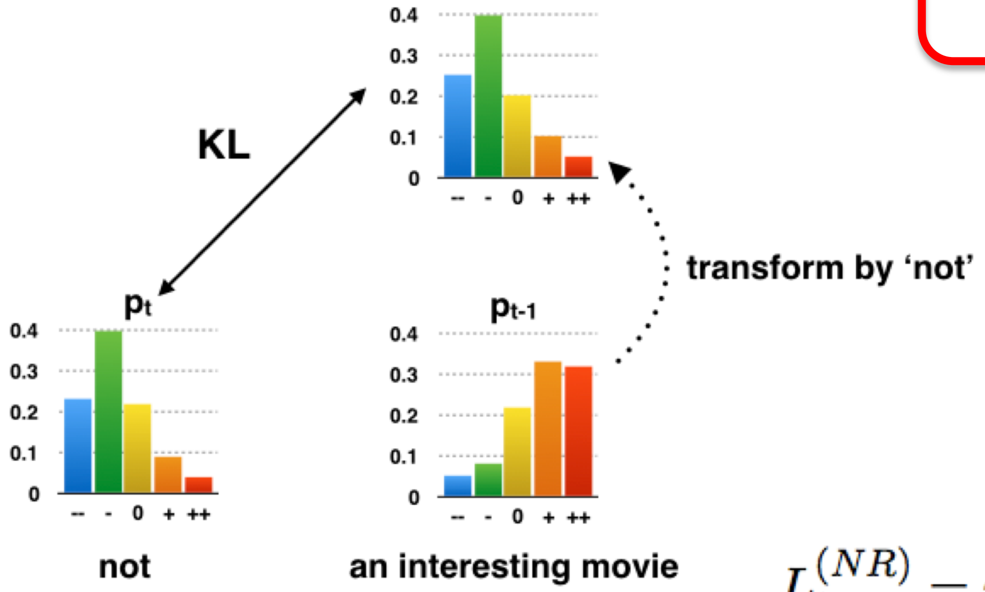
Each sentiment class has a shifting distribution

$$p_{t-1}^{(SR)} = p_{t-1} + s_{c(x_t)}$$

$$L_t^{(SR)} = max(0, D_{KL}(p_t || p_{t-1}^{(SR)}) - M)$$

Negation Regularizer

- The sentiment distribution should be shifted or reversed accordingly.



Each negator has a transition matrix

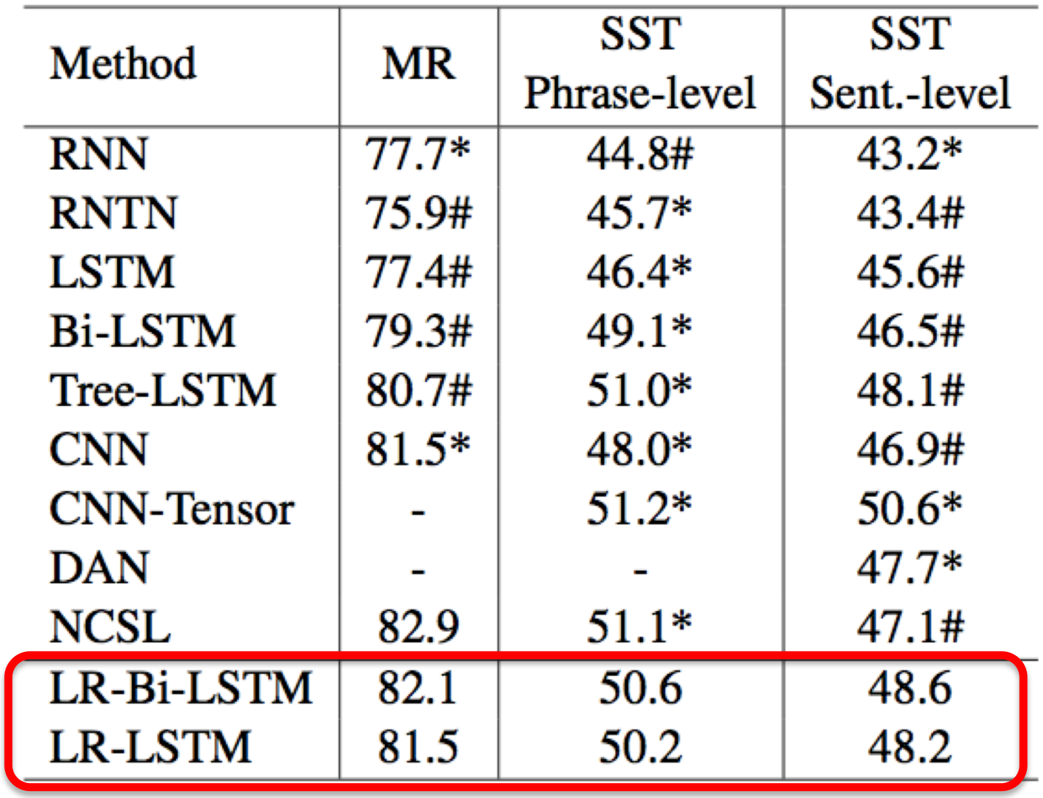
$$p_{t-1}^{(NR)} = softmax(T_{x_j} \times p_{t-1})$$

$$p_{t+1}^{(NR)} = softmax(T_{x_j} \times p_{t+1})$$

$$L_t^{(NR)} = min \begin{cases} max(0, D_{KL}(p_t || p_{t-1}^{(NR)}) - M) \\ max(0, D_{KL}(p_t || p_{t+1}^{(NR)}) - M) \end{cases}$$

Results

- *Phrase-level* means the models use phrase level annotation for training.
- *Sent.-level* means the models only use sentence level annotation.



Method	MR	SST Phrase-level	SST Sent.-level
RNN	77.7*	44.8#	43.2*
RNTN	75.9#	45.7*	43.4#
LSTM	77.4#	46.4*	45.6#
Bi-LSTM	79.3#	49.1*	46.5#
Tree-LSTM	80.7#	51.0*	48.1#
CNN	81.5*	48.0*	46.9#
CNN-Tensor	-	51.2*	50.6*
DAN	-	-	47.7*
NCSL	82.9	51.1*	47.1#
LR-Bi-LSTM	82.1	50.6	48.6
LR-LSTM	81.5	50.2	48.2

Summary

- **How abstractive linguistic knowledge (e.g., POS tags) can help representation learning?**

- **Our discoveries:**
  - **Syntactic knowledge** can help representation learning for sentiment classification
  - **Even only using POS tags**, structured models perform quite well
    - POS tag encodes much abstractive information
  - **Compact models** (fewer model parameters but still strong performance)

Our Related Papers

- Minlie Huang, Qiao Qian, Xiaoyan Zhu. Encoding Syntactic Knowledge in Neural Networks for Sentiment Classification. **ACM Trans. Inf. Syst**. 35, 3, Article 26 (June 2017), 27 pages.
- Qiao Qian, Minlie Huang, Xiaoyan Zhu. Linguistically Regularized LSTM for Sentiment Analysis. **ACL** 2017.
- Qiao Qian, Bo Tian, Minlie Huang, Yang Liu, Xuan Zhu, Xiaoyan Zhu. Learning Tag Embeddings and Tag-specific Composition Functions in Recursive Neural Network. **ACL** 2015, Beijing, China.

Thanks for attention!

- Dr. Minlie Huang, Tsinghua University
- Email: aihuang@tsinghua.edu.cn
- Homepage: http://coai.cs.tsinghua.edu.cn/hml