

Hierarchical Attention Networks for Document Classification

Zichao Yang¹ Diyi Yang¹ Chris Dyer¹
Xiaodong He² Alex Smola¹ Eduard Hovy¹

¹Carnegie Mellon University

²Microsoft Research, Redmond

San Diego, California, June 12-17, 2016

Outline

Gated Recurrent Unit

LSTM

S-LSTM Over Tree Structures

Hierarchical Attention Network

Experiments & Results

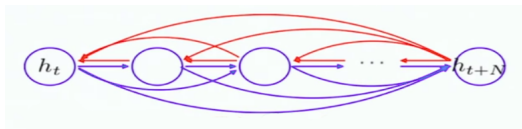
Context dependent attention weights

Visualization of attention

Natural language sentence matching

References

Gated Recurrent Unit (GRU)



- ▶ New hidden state at time t

$$h_t = f(h_{t-1}, x_t) = u_t \odot \tilde{h}_t + (1 - u_t) \odot h_{t-1}$$

- ▶ Candidate Update $\tilde{h}_t = \tanh(W[x_t] + U(r_t \odot h_{t-1}) + b)$

- ▶ Reset gate $r_t = \sigma(W_r[x_t] + U_r h_{t-1} + b_r)$

- ▶ Update Gate $u_t = \sigma(W_u[x_t] + U_u h_{t-1} + b_u)$

Long Short-Term Memory (LSTM)

Gated Recurrent Unit

[Cho et al., EMNLP2014;
Chung, Gulcehre, Cho, Bengio, DLUFL2014]

$$h_t = u_t \odot \tilde{h}_t + (1 - u_t) \odot h_{t-1}$$

$$\tilde{h} = \tanh(W[x_t] + U(r_t \odot h_{t-1}) + b)$$

$$u_t = \sigma(W_u[x_t] + U_u h_{t-1} + b_u)$$

$$r_t = \sigma(W_r[x_t] + U_r h_{t-1} + b_r)$$

Long Short-Term Memory

[Hochreiter & Schmidhuber, NC1999;
Gers, Thesis2001]

$$h_t = o_t \odot \tanh(c_t)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

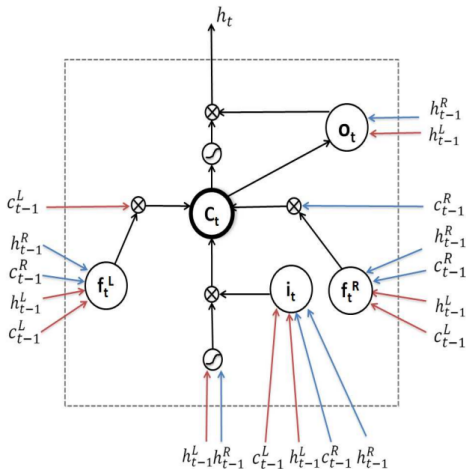
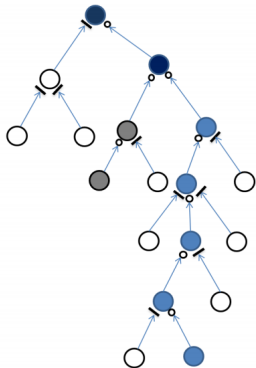
$$\tilde{c}_t = \tanh(W_c[x_t] + U_c h_{t-1} + b_c)$$

$$o_t = \sigma(W_o[x_t] + U_o h_{t-1} + b_o)$$

$$i_t = \sigma(W_i[x_t] + U_i h_{t-1} + b_i)$$

$$f_t = \sigma(W_f[x_t] + U_f h_{t-1} + b_f)$$

Long Short-Term Memory Over Tree Structures (S-LSTM)

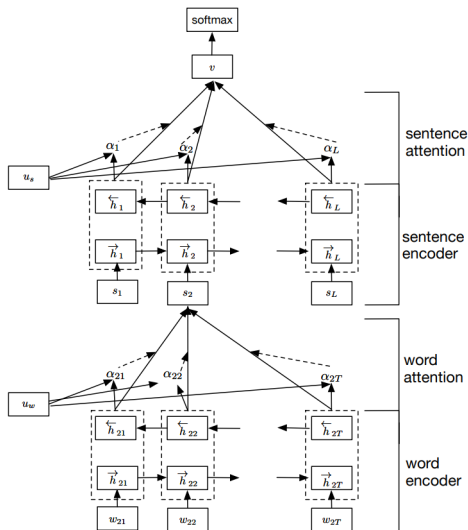


Long Short-Term Memory Over Tree Structures (S-LSTM)

Table 1. Performances (accuracies) of different models on the test set of Stanford Sentiment Tree Bank, at the sentence level (roots) and the phrase level. † shows the performance are statistically significantly better ($p < 0.05$) than the corresponding models.

MODELS	ROOTS	PHRASES
NB	41.0	67.2
SVM	40.7	64.3
RvNN	43.2	79.0
RNTN	45.7	80.7
S-LSTM	48.0†	81.9†

Hierarchical Attention Network



Word Encoder:

- ▶ $x_{it} = W_e w_{it}, t \in [1, T]$
- ▶ $\vec{h}_{it} = \overrightarrow{GRU}(x_{it}), t \in [1, T]$
- ▶ $\overleftarrow{h}_{it} = \overleftarrow{GRU}(x_{it}), t \in [T, 1]$

Word Attention:

- ▶ $u_{it} = \tanh(W_w h_{it} + b_w)$
- ▶ $\alpha_{it} = \frac{\exp(u_{it}^T u_w)}{\sum_t \exp(u_{it}^T u_w)}$
- ▶ $s_i = \sum_t \alpha_{it} h_{it}$

Experiments & Results

Figure: Document Classification, in percentage

	Methods	Yelp'13	Yelp'14	Yelp'15	IMDB	Yahoo Answer	Amazon
Zhang et al., 2015	BoW	-	-	58.0	-	68.9	54.4
	BoW TFIDF	-	-	59.9	-	71.0	55.3
	ngrams	-	-	56.3	-	68.5	54.3
	ngrams TFIDF	-	-	54.8	-	68.5	52.4
	Bag-of-means	-	-	52.5	-	60.5	44.1
Tang et al., 2015	Majority	35.6	36.1	36.9	17.9	-	-
	SVM + Unigrams	58.9	60.0	61.1	39.9	-	-
	SVM + Bigrams	57.6	61.6	62.4	40.9	-	-
	SVM + TextFeatures	59.8	61.8	62.4	40.5	-	-
	SVM + AverageSG	54.3	55.7	56.8	31.9	-	-
	SVM + SSWE	53.5	54.3	55.4	26.2	-	-
Zhang et al., 2015	LSTM	-	-	58.2	-	70.8	59.4
	CNN-char	-	-	62.0	-	71.2	59.6
	CNN-word	-	-	60.5	-	71.2	57.6
Tang et al., 2015	Paragraph Vector	57.7	59.2	60.5	34.1	-	-
	CNN-word	59.7	61.0	61.5	37.6	-	-
	Conv-GRNN	63.7	65.5	66.0	42.5	-	-
	LSTM-GRNN	65.1	67.1	67.6	45.3	-	-
This paper	HN-AVE	67.0	69.3	69.9	47.8	75.2	62.9
	HN-MAX	66.9	69.3	70.1	48.2	75.2	62.9
	HN-ATT	68.2	70.5	71.0	49.4	75.8	63.6

Context dependent attention weights

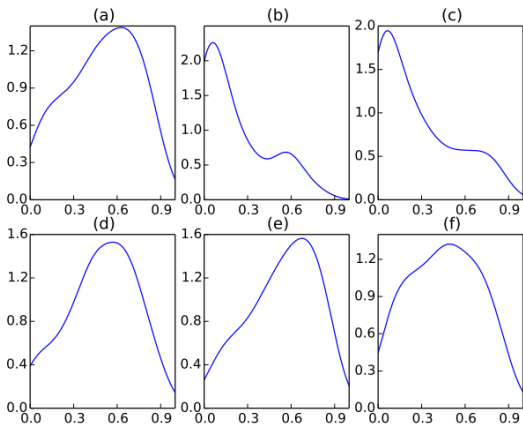


Figure 3: Attention weight distribution of `good`. (a) — aggregate distribution on the test split; (b)-(f) stratified for reviews with ratings 1-5 respectively. We can see that the weight distribution shifts to *higher* end as the rating goes higher.

Context dependent attention weights

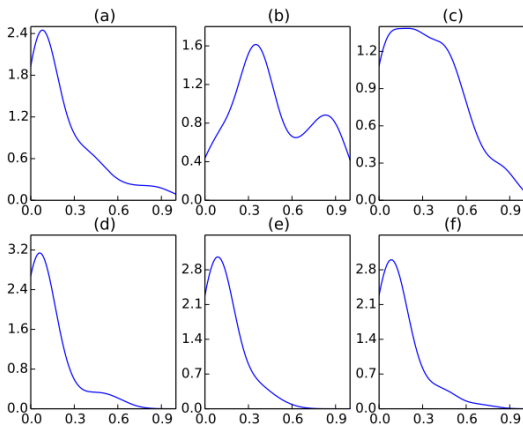
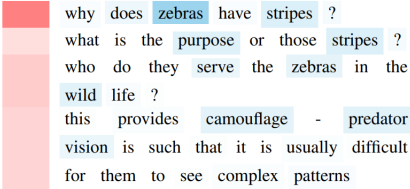


Figure 4: Attention weight distribution of the word `bad`. The setup is as above: (a) contains the aggregate distribution, while (b)-(f) contain stratifications to reviews with ratings 1-5 respectively. Contrary to before, the word `bad` is considered important for poor ratings and less so for good ones.

Visualization of attention

Figure: Documents from Yahoo Answers.
Left label - Science and Mathematics
Right label - Computers and Internet
Red - sentence weight, blue - word weight

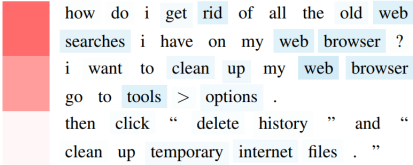
GT: 1 Prediction: 1



why does zebras have stripes ?
what is the purpose or those stripes ?
who do they serve the zebras in the
wild life ?
this provides camouflage - predator
vision is such that it is usually difficult
for them to see complex patterns

Detailed description: This visualization shows attention weights for a document in the 'Science and Mathematics' category. A vertical red bar on the left indicates sentence weights, with the highest weight on the first sentence. Blue highlights on individual words indicate word weights. The words 'zebras', 'stripes', 'purpose', 'stripes', 'zebras', 'camouflage', 'predator', 'vision', and 'difficult' are highlighted in blue.

GT: 4 Prediction: 4



how do i get rid of all the old web
searches i have on my web browser ?
i want to clean up my web browser
go to tools > options .
then click “ delete history ” and “
clean up temporary internet files . ”

Detailed description: This visualization shows attention weights for a document in the 'Computers and Internet' category. A vertical red bar on the left indicates sentence weights, with the highest weight on the second sentence. Blue highlights on individual words indicate word weights. The words 'get', 'rid', 'old', 'web', 'searches', 'browser', 'clean', 'up', 'web', 'browser', 'tools', 'options', 'delete', 'history', 'internet', and 'files' are highlighted in blue.

Natural language sentence matching (NLSM)

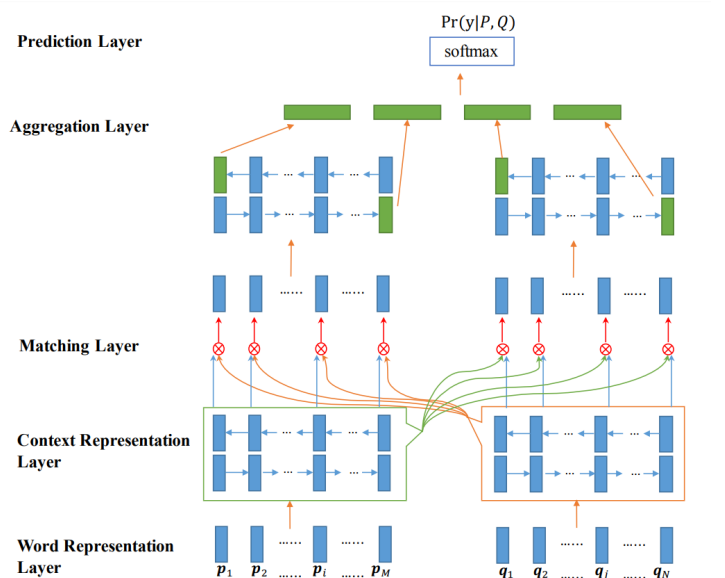
Task:

- ▶ (P, Q, y) , where $P = (p_1, \dots, p_j, \dots, p_M)$,
 $Q = (q_1, \dots, q_i, \dots, q_N)$, y - label (relationship between P and Q)
- ▶ $y^* = \operatorname{argmax}_{y \in Y} \Pr(y|P, Q)$

Example tasks:

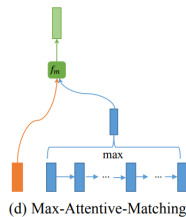
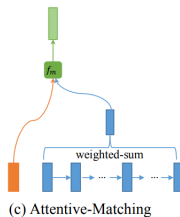
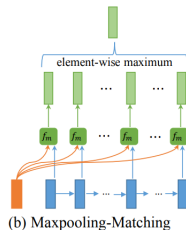
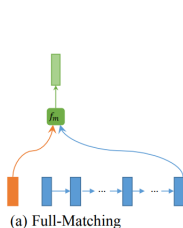
- ▶ Paraphrase identification task
- ▶ Natural language inference task
- ▶ Answer sentence selection task

Natural language sentence matching (NLSM)

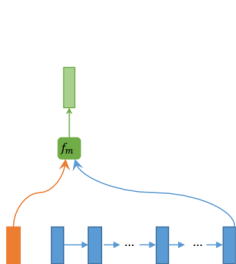


Multi-perspective Matching Operation

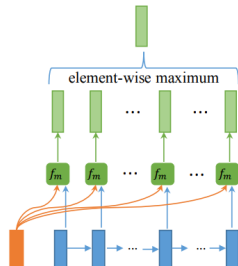
- ▶ $m = f_m(v_1, v_2; W)$, where $v_1, v_2 \in R^d$, $W \in R^{l \times d}$
 l - number of perspectives
- ▶ $m_k = \text{cosine}(W_k \odot v_1, W_k \odot v_2)$



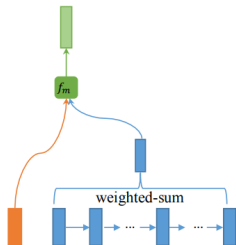
Multi-perspective Matching Operation



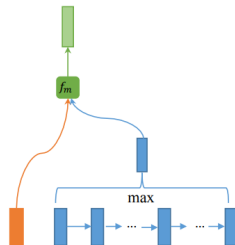
(a) Full-Matching



(b) Maxpooling-Matching



(c) Attentive-Matching



(d) Max-Attentive-Matching

Multi-perspective Matching Operation

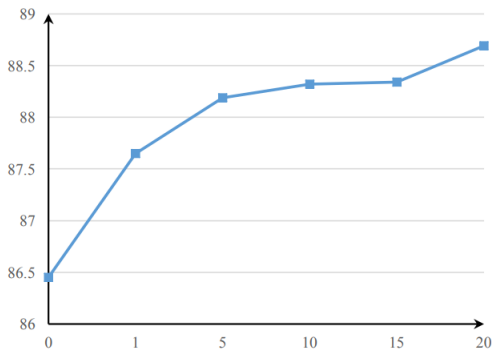


Figure 3: Influence of the multi-perspective cosine matching function in Eq.(3) .

References



Zichao Yang, Diyi Yang, Chris Dyer
Hierarchical Attention Networks for Document Classification
<http://www.aclweb.org/anthology/N16-1174>



Zhiguo Wang, Wael Hamza, Radu Florian
Bilateral Multi-Perspective Matching for Natural Language
Sentences
<https://arxiv.org/pdf/1702.03814.pdf>



Xiaodan Zhu, Parinaz Sobhani, Hongyu Guo
Long Short-Term Memory Over Tree Structures
<https://arxiv.org/pdf/1503.04881.pdf>