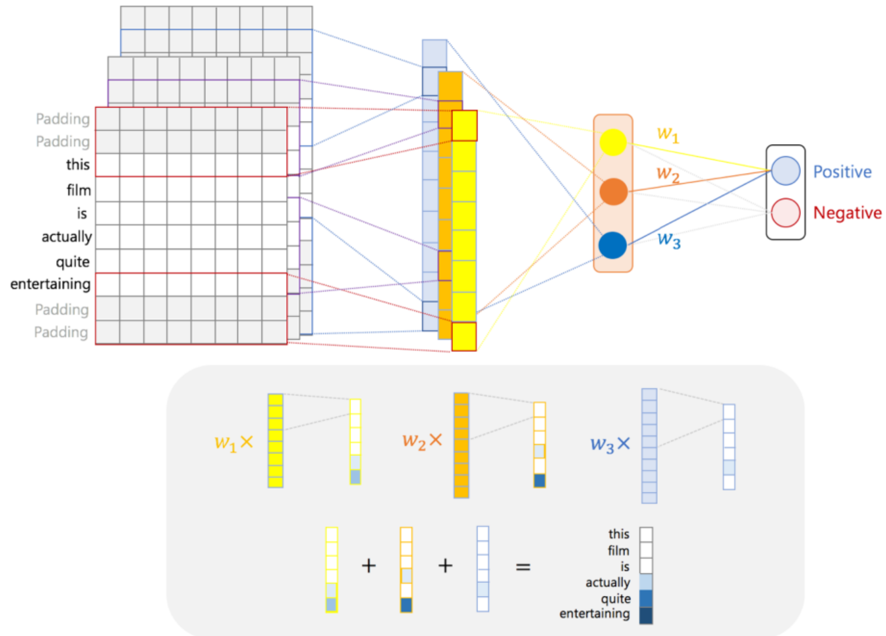
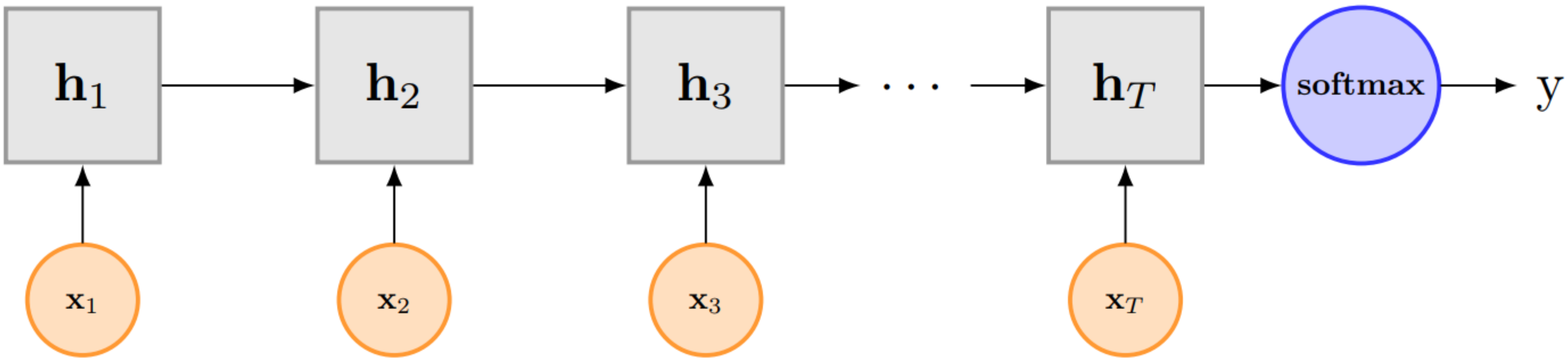


Lecture 9-3: Document Classification RNN-based Classifier

Pilsung Kang
School of Industrial Management Engineering
Korea University

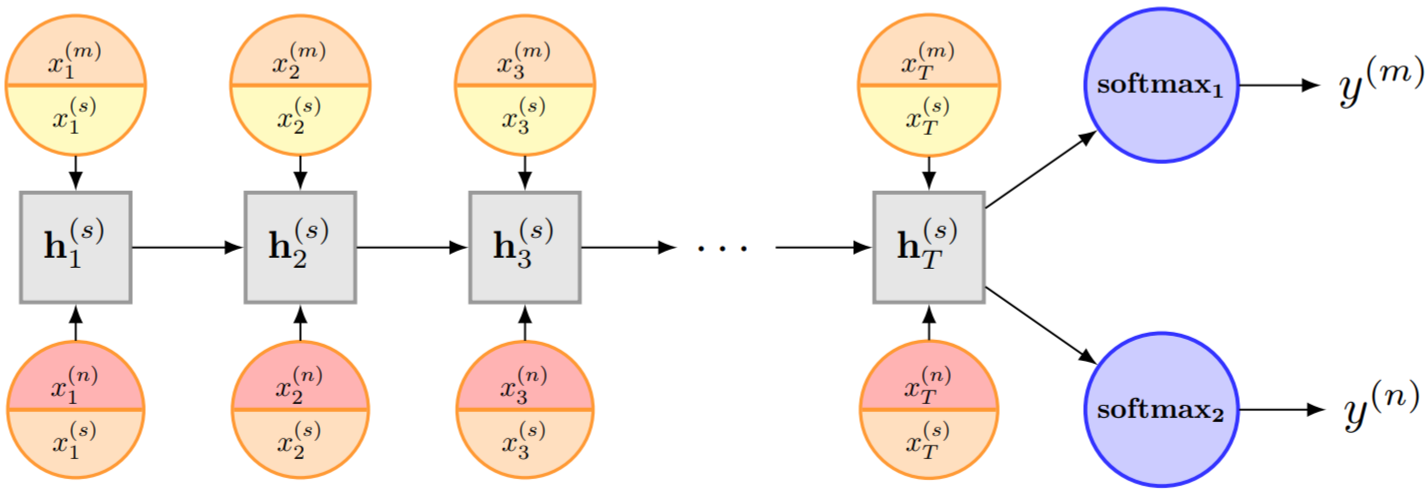
RNN for Text Classification

- RNN Basic Structure for Text Classification



RNN for Text Classification

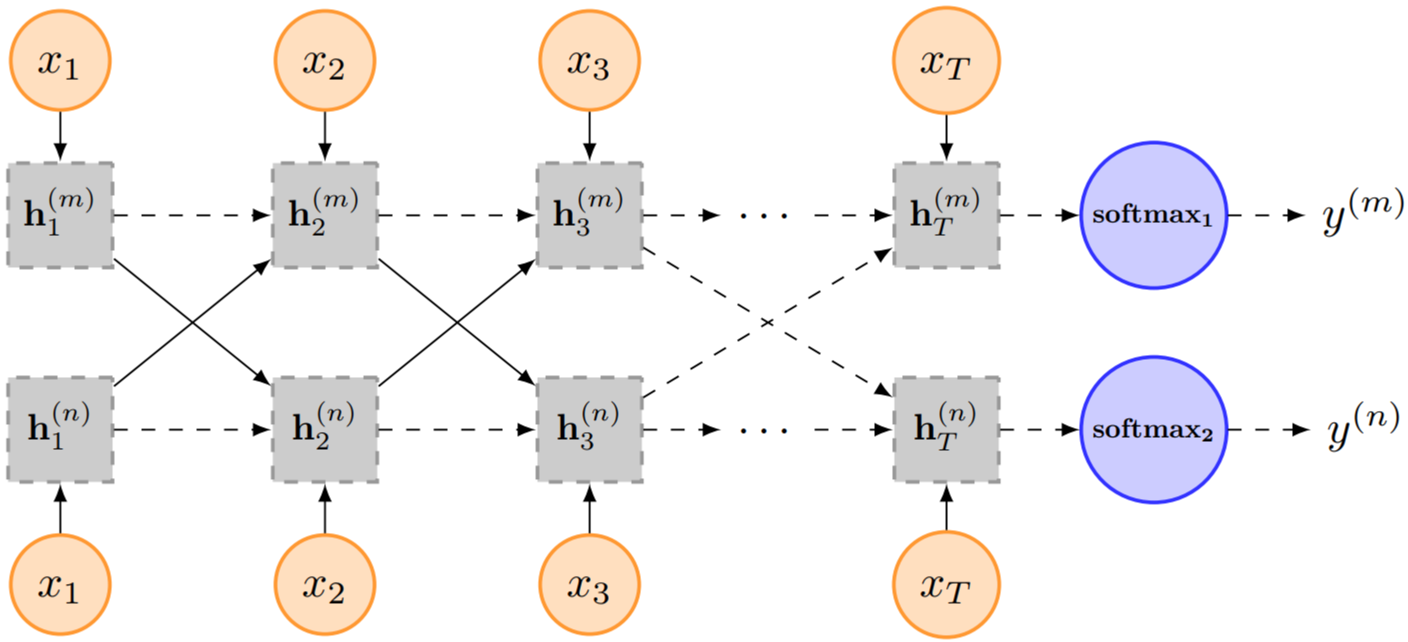
- RNN for Multi-Task Learning (Liu et al., 2016)
  - ✓ (Note) The task is identical but the datasets are different



(a) Model-I: Uniform-Layer Architecture

RNN for Text Classification

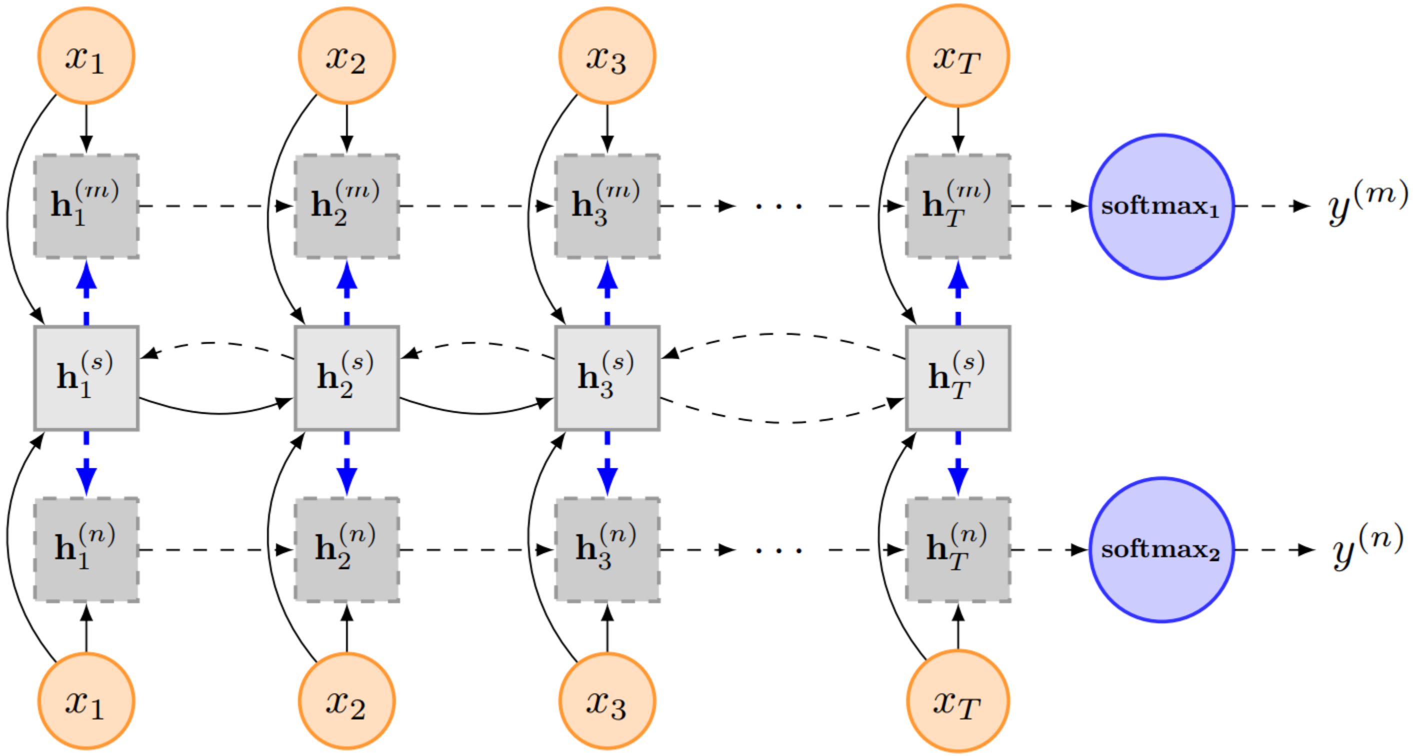
- RNN for Multi-Task Learning



(b) Model-II: Coupled-Layer Architecture

RNN for Text Classification

- RNN for Multi-Task Learning



(c) Model-III: Shared-Layer Architecture

RNN for Text Classification

- RNN for Multi-Task Learning

Model	SST-1	SST-2	SUBJ	IMDB	AvgΔ
Single Task	45.9	85.8	91.6	88.5	-
Joint Learning	46.5	86.7	92.0	89.9	+0.8
+ Fine Tuning	**48.5**	**87.1**	**93.4**	**90.8**	+2.0

Table 2: Results of the uniform-layer architecture.

Model	SST-1	SST-2	SUBJ	IMDB	AvgΔ
Single Task	45.9	85.8	91.6	88.5	-
SST1-SST2	**48.9**	**87.4**	-	-	+2.3
SST1-SUBJ	46.3	-	92.2	-	+0.5
SST1-IMDB	46.9	-	-	89.5	+1.0
SST2-SUBJ	-	86.5	92.5	-	+0.8
SST2-IMDB	-	86.8	-	**89.8**	+1.2
SUBJ-IMDB	-	-	**92.7**	89.3	+0.9

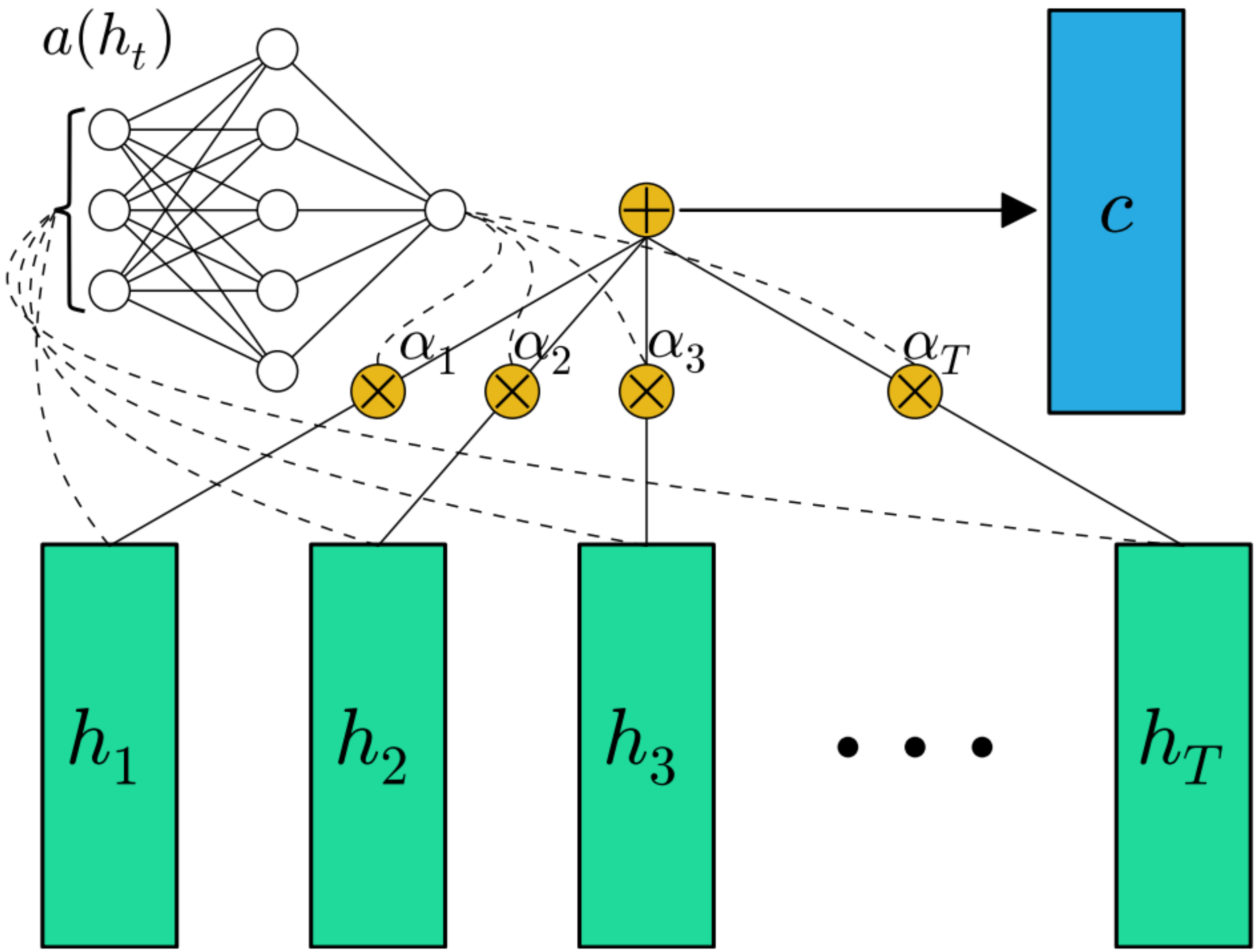
Table 3: Results of the coupled-layer architecture.

Model	SST-1	SST-2	SUBJ	IMDB	AvgΔ
Single Task	45.9	85.8	91.6	88.5	-
Joint Learning	47.1	87.0	92.5	90.7	+1.4
+ LM	47.9	86.8	93.6	91.0	+1.9
+ Fine Tuning	**49.6**	**87.9**	**94.1**	**91.3**	+2.8

Table 4: Results of the shared-layer architecture.

RNN: Attention

- Attention mechanism for finding significant words in document classification

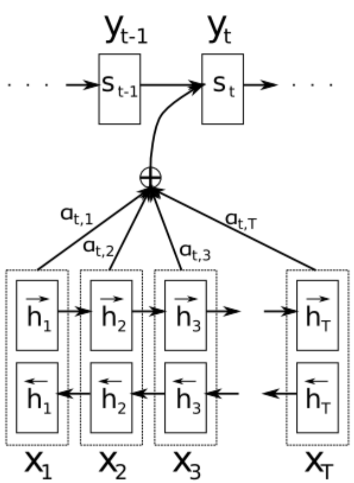


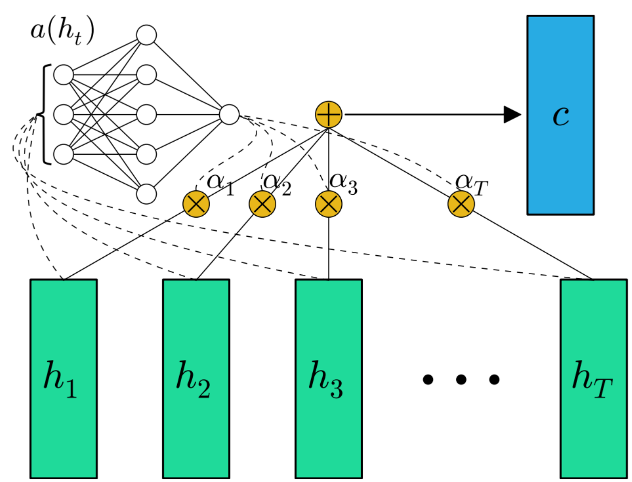
Source: (Raffel and Ellis, 2016)

RNN: Attention

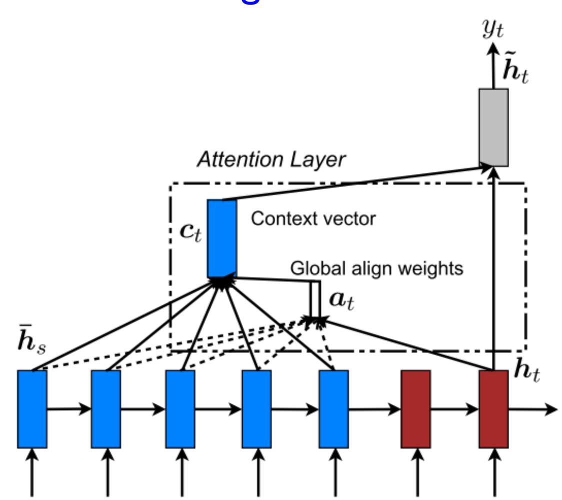
- Two main attention mechanisms
  - ✓ Bahadanau attention (Bahdanau et al., 2015)
    - Attention scores are separated trained, the current hidden state is a function of the context vector and the previous hidden state
  - ✓ Luong attention (Luong et al., 2015)
    - Attention scores are not trained, the new current hidden state is the simple tanh of the weigthed concatenation of the context vector and the current hidden state of the decoder

Bahdanau attention





Luong attention



RNN: Attention

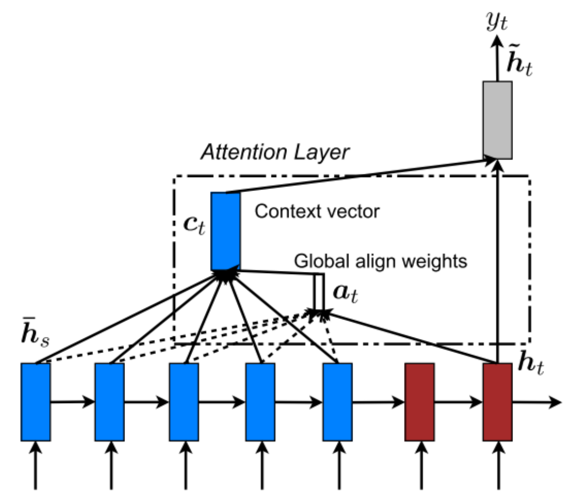
- Luong attention

  ✓ New hidden state of the decoder is the simple tanh of the weigthed concatenation of the context vector and the current hidden state of the decoder:

$$\tilde{\mathbf{h}}_t = tanh(\mathbf{W_c}[\mathbf{c}_t; \mathbf{h}_t])$$

  ✓ The attention vector is fed throught the softmax layer to produce the predictive distribution:

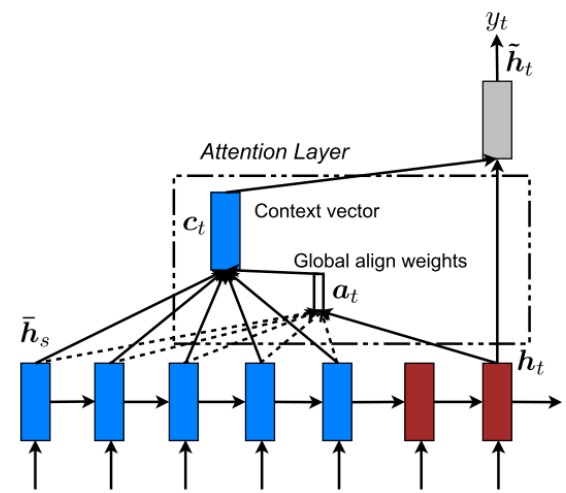
$$p(y_t | y_{y<t}, x) = \mathrm{softmax}(\mathbf{W_s}\tilde{\mathbf{h}}_t)$$



RNN: Attention

- Luong attention
  - ✓ A variable-length alignment vector:

$$\mathbf{a}_t(s) = \text{align}(\mathbf{h}_t, \overline{\mathbf{h}}_s) = \frac{exp\Big(\text{score}(\mathbf{h}_t, \overline{\mathbf{h}}_s)\Big)}{\sum_{s'} exp\Big(\text{score}(\mathbf{h}_t, \overline{\mathbf{h}}_s')\Big)}$$



  - ✓ **score** is referred as a context-based function:

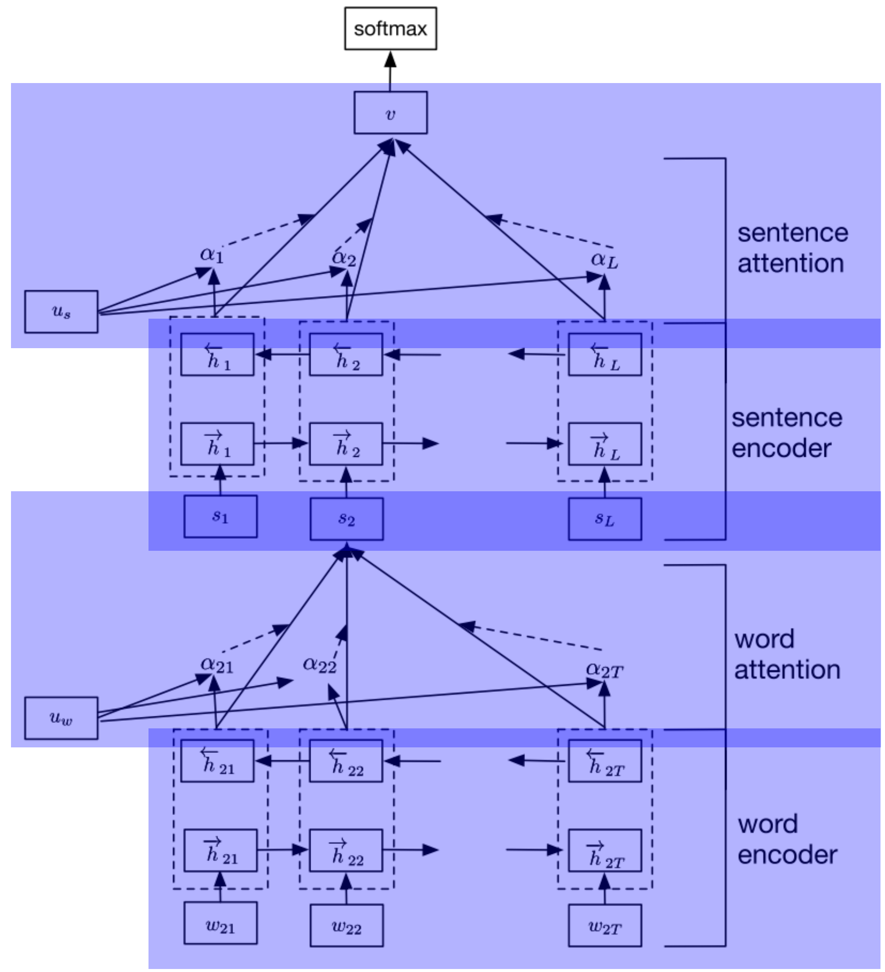
$$\text{score}(\mathbf{h}_t, \overline{\mathbf{h}}_s) = \begin{cases} {\mathbf{h}_t}^T \overline{\mathbf{h}}_s, & dot \\ {\mathbf{h}_t}^T \mathbf{W_a} \overline{\mathbf{h}}_s, & general \\ \mathbf{v}_a^T tanh(\mathbf{W_c}[\mathbf{c}_t; \mathbf{h}_t]), & concat \end{cases}$$

  - ✓ Context vector

$$\mathbf{c}_t = \overline{\mathbf{h}}_s \mathbf{a}_t$$

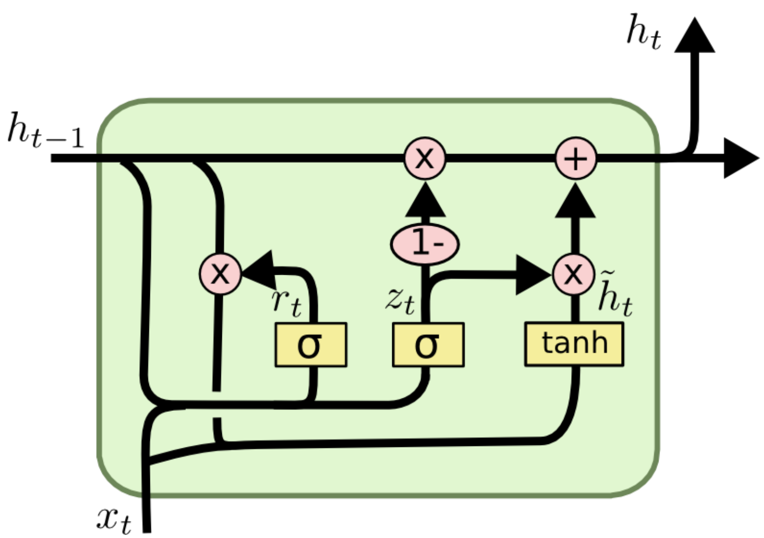
RNN for Document Classification and Attention

- Hierarchical Attention Network (Yang et al., 2016)
  - ✓ Level 1: Word sequence encoder
  - ✓ Level 2: Word-level attention layer
  - ✓ Level 3: Sentence encoder
  - ✓ Level 4: Sentence-level attention layer



RNN for Document Classification and Attention

- Hierarchical Attention Network: Sequence encoder
  - ✓ Bidirectional GRU is employed



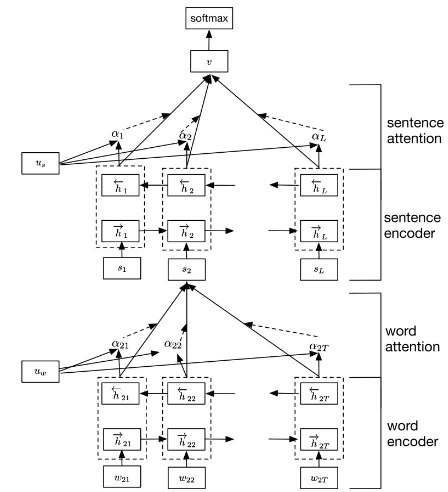
$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$ Update gate

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$ Reset gate

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

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

RNN for Document Classification and Attention



- Hierarchical Attention Network: Hierarchical Attention

✓ Word encoder

$$\mathbf{x}_{it} = \mathbf{W}_e w_{it}, \ t \in [1, T],$$

$$\overrightarrow{\mathbf{h}}_{it} = \overrightarrow{GRU}(\mathbf{x}_{it}), \quad t \in [1, T],$$

$$\overleftarrow{\mathbf{h}}_{it} = \overleftarrow{GRU}(\mathbf{x}_{it}), \quad t \in [1, T],$$

$$\mathbf{h}_{it} = [\overrightarrow{\mathbf{h}}_{it}, \overleftarrow{\mathbf{h}}_{it}].$$

✓ Word Attention

$$\mathbf{u}_{it} = tanh(\mathbf{W}_w \mathbf{h}_{it} + \mathbf{b}_w)$$

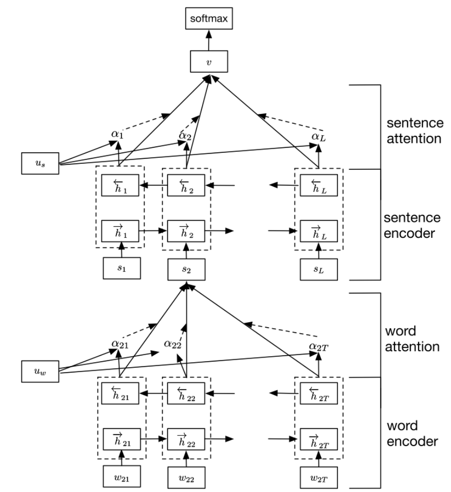
$$\alpha_{it} = \frac{\exp(\mathbf{u}_{it}^T \mathbf{u}_w)}{\sum_{t'} \exp(\mathbf{u}_{it'}^T \mathbf{u}_w)}$$

$$\mathbf{s}_i = \sum_t \alpha_{it} \mathbf{h}_{it}$$

Word context vector
- Can be seen as a high level representation of a fixed query "what is the informative word?"
- Randomly initialized and jointly learned during the training process

RNN for Document Classification and Attention

- Hierarchical Attention Network: Hierarchical Attention



✓ Sentence encoder

$$\overrightarrow{\mathbf{h}}_i = \overrightarrow{GRU}(\mathbf{s}_i), \quad i \in [1, L],$$

$$\overleftarrow{\mathbf{h}}_i = \overleftarrow{GRU}(\mathbf{s}_i), \quad i \in [1, L],$$

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i, \overleftarrow{\mathbf{h}}_i].$$

✓ Sentence attention

$$\mathbf{u}_i = tanh(\mathbf{W}_s \mathbf{h}_i + \mathbf{b}_s)$$

$$\alpha_i = \frac{\exp(\mathbf{u}_i^T \mathbf{u}_s)}{\sum_{i'} \exp(\mathbf{u}_{i'}^T \mathbf{u}_s)}$$

$$\mathbf{v} = \sum_i \alpha_i \mathbf{h}_i$$

Sentence context vector

- Can be seen as a high level representation of a fixed query "what is the informative sentence?"
- Randomly initialized and jointly learned during the training process

RNN for Document Classification and Attention

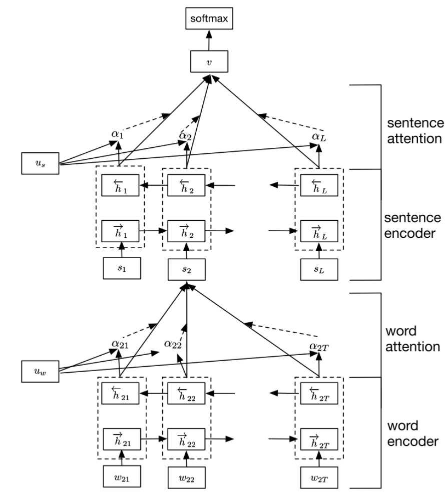
- Hierarchical Attention Network

  ✓ Document classification

$$p = \mathrm{softmax}(\mathbf{W}_c \mathbf{v} + b_c)$$

  ✓ Loss function: negative log likelihood of the correct labels

$$L = -\sum_d log p_{dj}$$



RNN for Document Classification and Attention

- Hierarchical Attention Network: Experiment
  - ✓ Data description

Data set	classes	documents	average #s	max #s	average #w	max #w	vocabulary
Yelp 2013	5	335,018	8.9	151	151.6	1184	211,245
Yelp 2014	5	1,125,457	9.2	151	156.9	1199	476,191
Yelp 2015	5	1,569,264	9.0	151	151.9	1199	612,636
IMDB review	10	348,415	14.0	148	325.6	2802	115,831
Yahoo Answer	10	1,450,000	6.4	515	108.4	4002	1,554,607
Amazon review	5	3,650,000	4.9	99	91.9	596	1,919,336

RNN for Document Classification and Attention

- Hierarchical Attention Network: Experiment
  - ✓ Classification performance

	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**

RNN for Document Classification and Attention

- Hierarchical Attention Network: Experiment
  - ✓ Attention distribution of two words: “good” and “bad”

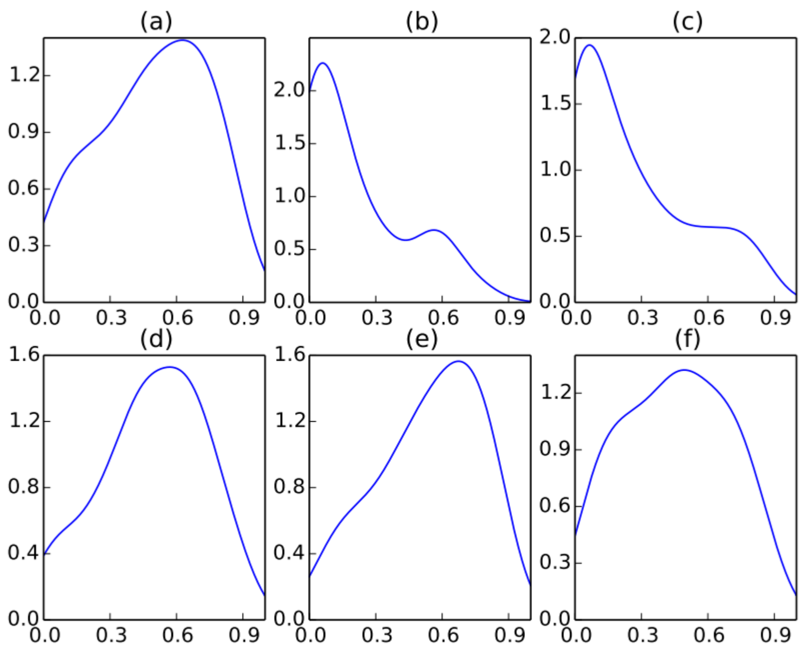


Figure 3: Attention weight distribution of `good`. (a) — aggregate distribution on the test split; (b)-(f) stratified for reviews with ratings 1-5 respectively. We can see that the weight distribution shifts to *higher* end as the rating goes higher.

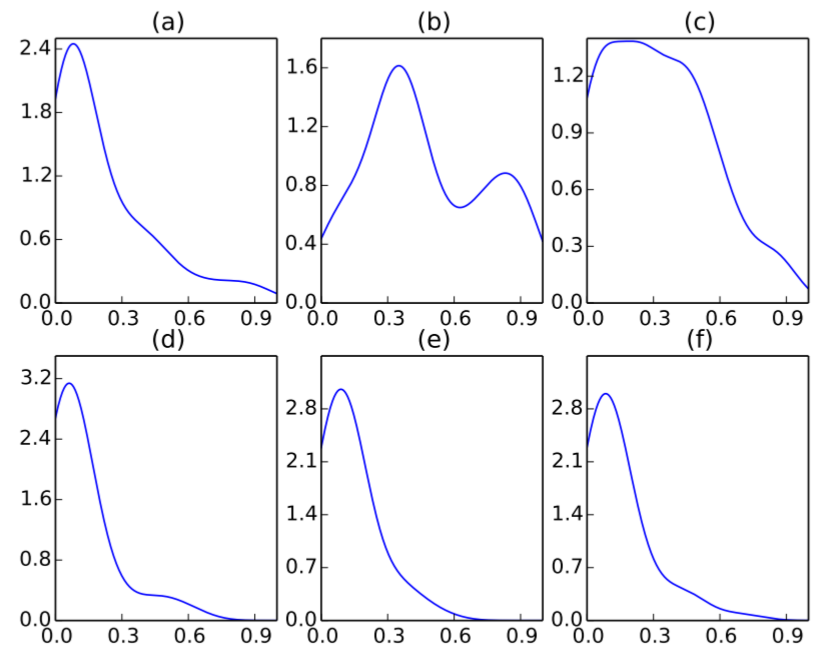


Figure 4: Attention weight distribution of the word `bad`. The setup is as above: (a) contains the aggregate distribution, while (b)-(f) contain stratifications to reviews with ratings 1-5 respectively. Contrary to before, the word `bad` is considered important for poor ratings and less so for good ones.

RNN for Document Classification and Attention

- Hierarchical Attention Network: Experiment
  - ✓ Visualization of attention scores

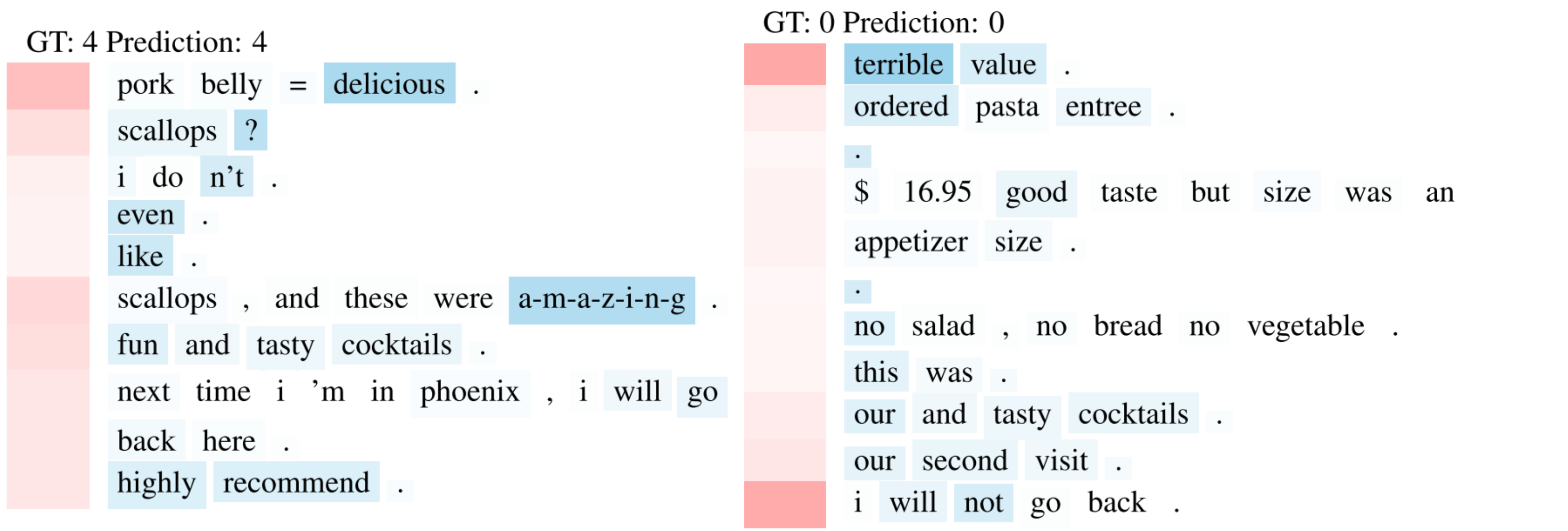
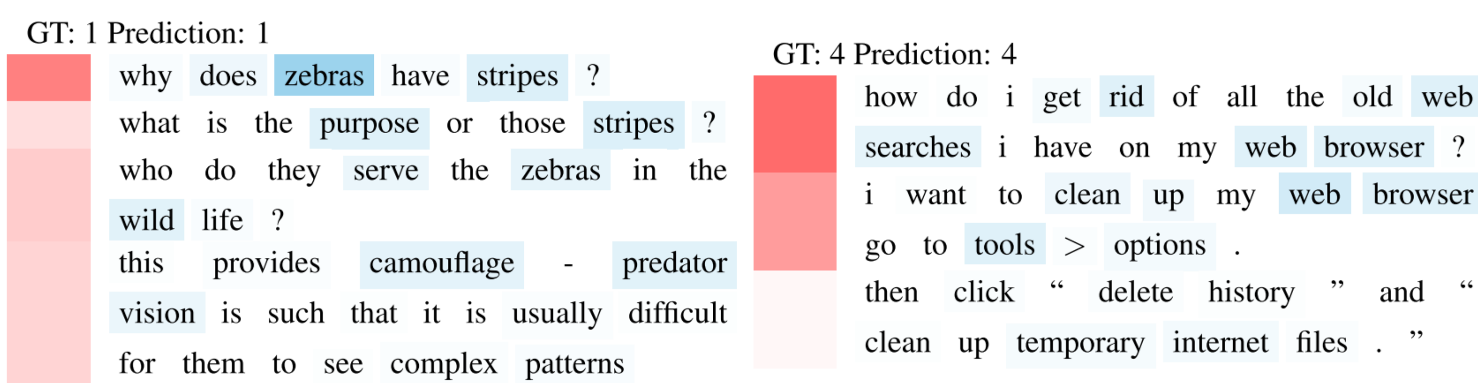


Figure 5: Documents from Yelp 2013. Label 4 means star 5, label 0 means star 1.



RNN for Document Classification and Attention

- Comparison of RNN attention and CNN localization

CAM^2-4channel	Seeing as the vote average was pretty low and the fact that the clerk in the video store thought was just OK I didn t t have much expectations when renting this film But contrary to the above I **enjoyed it** a lot This is acharming movie It didn t need to grow on me **I enjoyed it** from the beginning Mel Brooks gives a great performance as thelead character I think somewhat different from his usual persona in his movies There s not a lot of knockout jokes or somethinglike that but there are some rather hilarious scenes and overall this is a **very enjoyable and very easy** to watch film **Very recommended** **Positive**
HAN	Seeing as the vote average was **pretty** low and the fact that the clerk in the video store thought it was just OK I didnt have much expectations when renting this film But contrary to the above I **enjoyed** it a lot This is a **charming** movie It didnt need to grow on me I **enjoyed** it from the beginning Mel Brooks gives a great **performance** as the lead character I think somewhat different from his usual persona in his movies There not a lot of knockoutjokes or something like that but there are some rather hilarious scenes and **overall** this is a very enjoyable and veryeasy to watch film **Very** recommended **Positive**

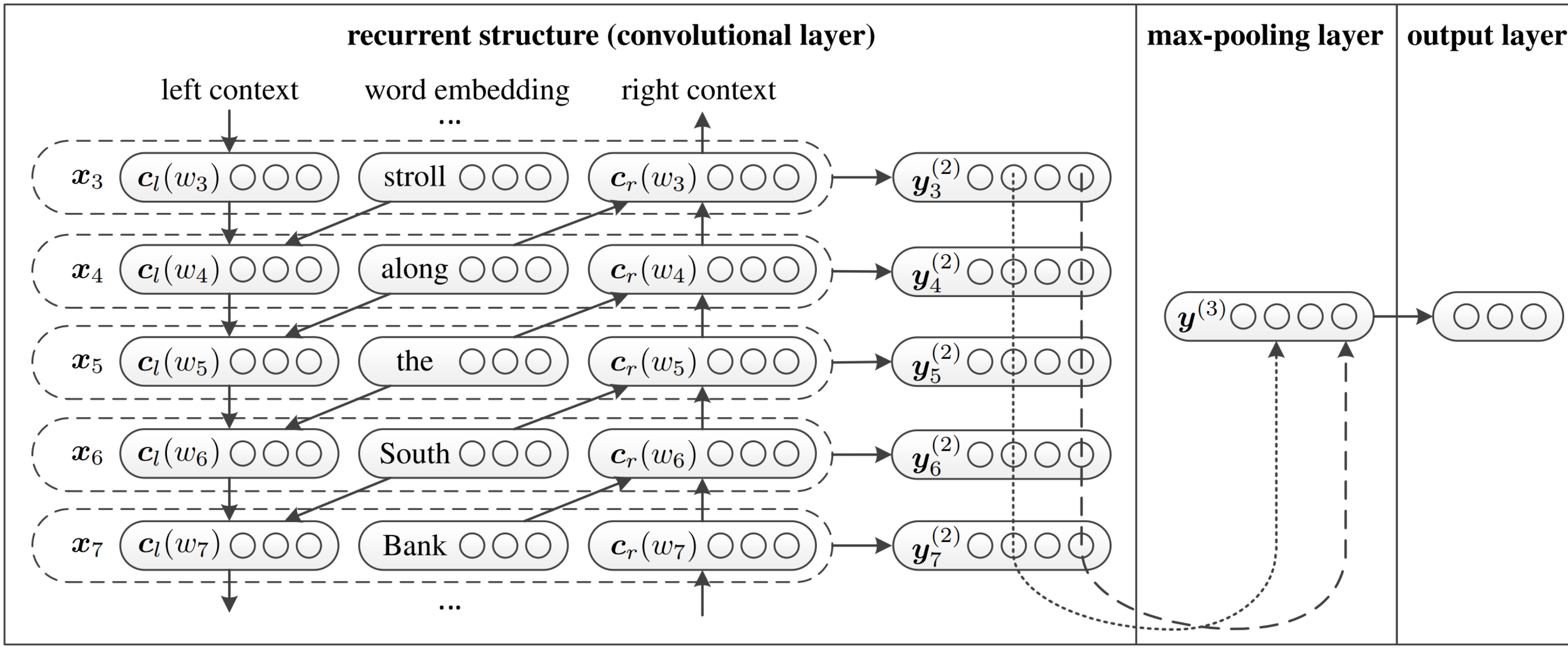
RNN for Document Classification and Attention

- Comparison of RNN attention and CNN localization

CAM²-4channel	I hate this movie It is a horrid movie Sean Young s character is completely unsympathetic He r performance is wooden at best The storyline is completely **predictable and completely uni nteresting** I would never recommend this film to anyone It is one of the **worst** movies I have ever had the misfortune to see **Negative**
HAN	I **hate** this movie It is a **horrid** movie Sean Youngs character **is** completely unsympathetic He rperformance **is** wooden at best The storyline is completely predictable and completely **unin teresting** I would never recommend **this** film to anyone It is one of the **worst** movies I have e ver had the misfortune to see **Negative**

RNN + CNN

- Recurrent Convolutional Neural Network for Text Classification (Lai et al., 2015)



RNN + CNN

- Recurrent Convolutional Neural Network for Text Classification
  - ✓ Word representation learning
    - Consider the left and right context words

$$\boldsymbol{c}_l(w_i) = f(W^{(l)}\boldsymbol{c}_l(w_{i-1}) + W^{(sl)}\boldsymbol{e}(w_{i-1}))$$
$$\boldsymbol{c}_r(w_i) = f(W^{(r)}\boldsymbol{c}_r(w_{i+1}) + W^{(sr)}\boldsymbol{e}(w_{i+1}))$$

    - Concatenate the target and left/right context word vectors

$$\boldsymbol{x}_i = [\boldsymbol{c}_l(w_i); \boldsymbol{e}(w_i); \boldsymbol{c}_r(w_i)]$$

    - Weighted averaging and non-linear transformation

$$\boldsymbol{y}_i^{(2)} = \tanh\left(W^{(2)}\boldsymbol{x}_i + \boldsymbol{b}^{(2)}\right)$$

RNN + CNN

- Recurrent Convolutional Neural Network for Text Classification
  - ✓ Text classification
    - Max pooling of each word representation

$$\boldsymbol{y}^{(3)} = \max_{i=1}^{n} \boldsymbol{y}_i^{(2)}$$

    - Linear transform and softmax

$$\boldsymbol{y}^{(4)} = W^{(4)}\boldsymbol{y}^{(3)} + \boldsymbol{b}^{(4)}$$

$$p_i = \frac{\exp\left(\boldsymbol{y}_i^{(4)}\right)}{\sum_{k=1}^{n} \exp\left(\boldsymbol{y}_k^{(4)}\right)}$$

Which Structure is Better?

- CNN vs. RNN (Yin et al., 2017)

			performance	lr	hidden	batch	sentLen	filter_size	margin
TextC	SentiC (acc)	CNN	82.38	0.2	20	5	60	3	–
		GRU	**86.32**	0.1	30	50	60	–	–
		LSTM	84.51	0.2	20	40	60	–	–
	RC (F1)	CNN	68.02	0.12	70	10	20	3	–
		GRU	**68.56**	0.12	80	100	20	–	–
		LSTM	66.45	0.1	80	20	20	–	–
SemMatch	TE (acc)	CNN	77.13	0.1	70	50	50	3	–
		GRU	**78.78**	0.1	50	80	65	–	–
		LSTM	77.85	0.1	80	50	50	–	–
	AS (MAP & MRR)	CNN	(**63.69**,**65.01**)	0.01	30	60	40	3	0.3
		GRU	(62.58,63.59)	0.1	80	150	40	–	0.3
		LSTM	(62.00,63.26)	0.1	60	150	45	–	0.1
	QRM (acc)	CNN	**71.50**	0.125	400	50	17	5	0.01
		GRU	69.80	1.0	400	50	17	-	0.01
		LSTM	71.44	1.0	200	50	17	-	0.01
SeqOrder	PQA (hit@10)	CNN	54.42	0.01	250	50	5	3	0.4
		GRU	**55.67**	0.1	250	50	5	–	0.3
		LSTM	55.39	0.1	300	50	5	–	0.3
ContextDep	POS tagging (acc)	CNN	94.18	0.1	100	10	60	5	–
		GRU	93.15	0.1	50	50	60	–	–
		LSTM	93.18	0.1	200	70	60	–	–
		Bi-GRU	94.26	0.1	50	50	60	–	–
		Bi-LSTM	**94.35**	0.1	150	5	60	–	–

Which Structure is Better?

- CNN vs. RNN (Yin et al., 2017)

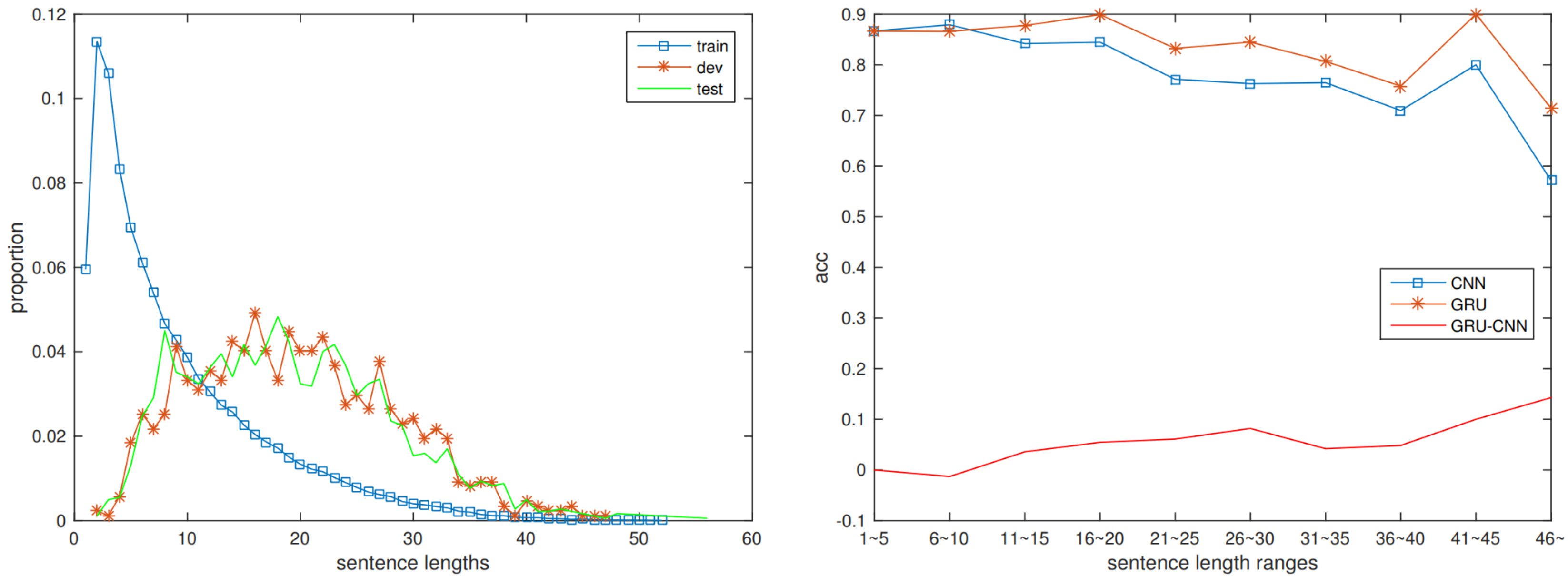


Figure 2: Distributions of sentence lengths (left) and accuracies of different length ranges (right).

Which Structure is Better?

- CNN vs. RNN (Yin et al., 2017)

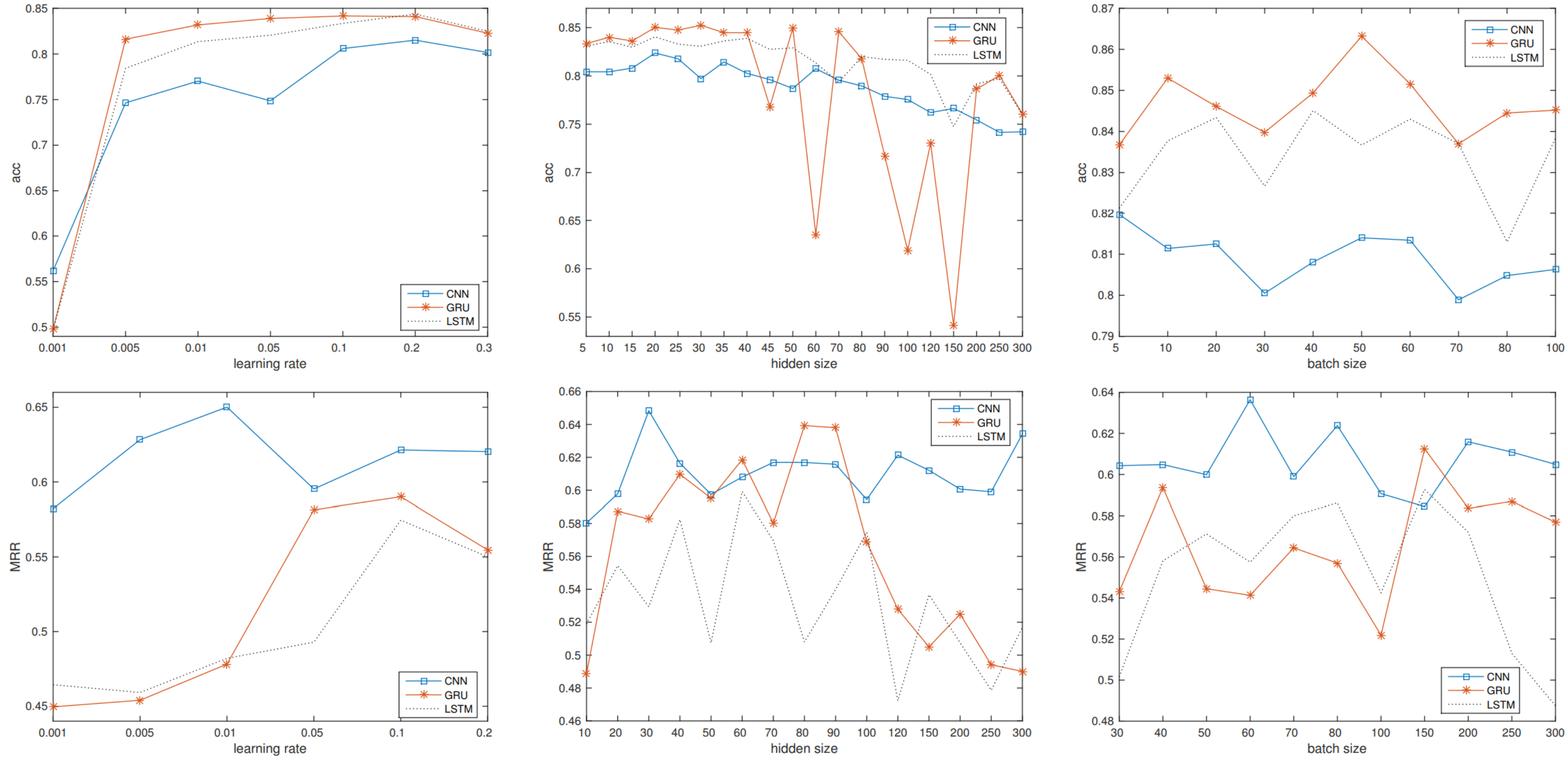


Figure 3: Accuracy for sentiment classification (top) and MRR for WikiQA (bottom) as a function of three hyperparameters: learning rate (left), hidden size (center) and batch size (right).


ANY
questions?