Context to Sequence

Typical Frameworks and Applications

Piji Li

Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong

FDU-CUHK, 2017

Outline

1. Introduction
2. Frameworks
   - Overview
   - Teacher Forcing
   - Adversarial Reinforce
   - Tricks
3. Applications
4. Conclusions

Introduction

Introduction

- Typical ctx2seq frameworks have obtained significant improvements:
  - Neural machine translation.
  - Abstraction text summarization.
  - Dialog/Conversation system - Chatbot.
  - Caption generation for images and videos.
- Various strategies to train a better ctx2seq model:
  - Improving teacher forcing.
  - Adversarial training.
  - Reinforcement learning.
  - Tricks (copy, coverage, dual training, etc.).
- Interesting applications.

Frameworks

Outline

1 Introduction

2 Frameworks
- Overview
- Teacher Forcing
- Adversarial Reinforce
- Tricks

3 Applications

4 Conclusions

Overview

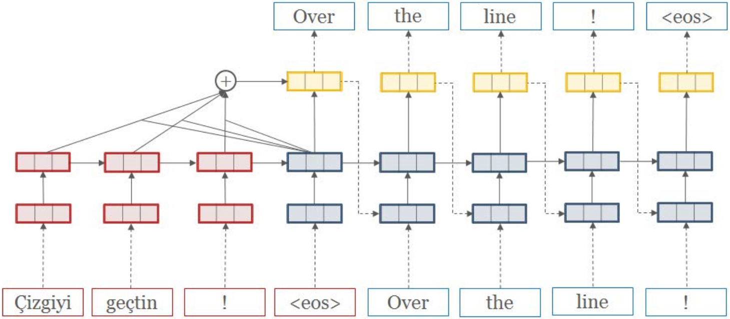


Figure 1: Seq2seq framework with attention mechanism and teacher forcing.[1]

[1] https://github.com/OpenNMT

Outline

1. Introduction
2. Frameworks
   - Overview
   - Teacher Forcing
   - Adversarial Reinforce
   - Tricks
3. Applications
4. Conclusions

Teacher Forcing

- Feed the **ground-truth** sample y_t back to the model to be conditioned on for the prediction of later outputs.
- **Advantages**:
  - Force the decoder to stay close to the ground-truth sequence.
  - Faster convergence speed.
- **Disadvantage**:
  - In prediction: sampling & greedy decoding; beam search.
  - Mismatch between training and testing.
  - Error accumulation during decoding phase.

- Bengio, Samy, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. "**Scheduled sampling for sequence prediction with recurrent neural networks**." NIPS, 2015. [Google Research]
- Lamb, Alex M., Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C. Courville, and Yoshua Bengio. "**Professor forcing: A new algorithm for training recurrent networks**." NIPS, 2016. [University of Montreal]
- Jang, Eric, Shixiang Gu, and Ben Poole. "**Categorical reparameterization with gumbel-softmax**." ICLR, 2017.
  Gu, Jiatao, Daniel Jiwoong Im, and Victor OK Li. "**Neural Machine Translation with Gumbel-Greedy Decoding**." arXiv (2017).

Teacher Forcing

- Bengio, Samy, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. "**Scheduled sampling for sequence prediction with recurrent neural networks**." NIPS, 2015. [Google Research]

- Overview of the scheduled sampling method:

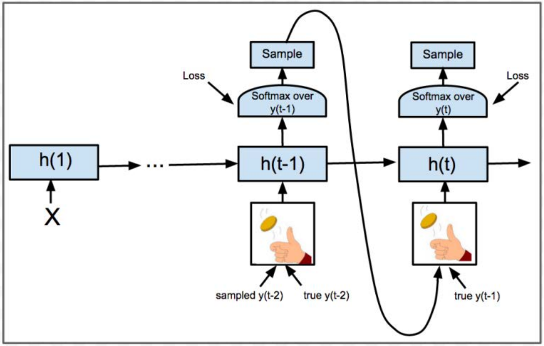


Figure 2: Illustration of the Scheduled Sampling approach, where one flips a coin at every time step to decide to use the true previous token or one sampled from the model itself.[1]

- Image Captioning, MSCOCO:

Approach vs Metric	BLEU-4	METEOR	CIDER
Baseline	28.8	24.2	89.5
Baseline with Dropout	28.1	23.9	87.0
Always Sampling	11.2	15.7	49.7
Scheduled Sampling	**30.6**	**24.3**	**92.1**
Uniform Scheduled Sampling	29.2	24.2	90.9
Baseline ensemble of 10	30.7	25.1	95.7
Scheduled Sampling ensemble of 5	**32.3**	**25.4**	**98.7**

- Constituency Parsing, WSJ 22:

Approach	F1
Baseline LSTM	86.54
Baseline LSTM with Dropout	87.0
Always Sampling	-
Scheduled Sampling	**88.08**
Scheduled Sampling with Dropout	**88.68**

Teacher Forcing

- Lamb, Alex M., Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C. Courville, and Yoshua Bengio. "**Professor forcing: A new algorithm for training recurrent networks**." NIPS, 2016. [University of Montreal]

- Architecture of the Professor Forcing:

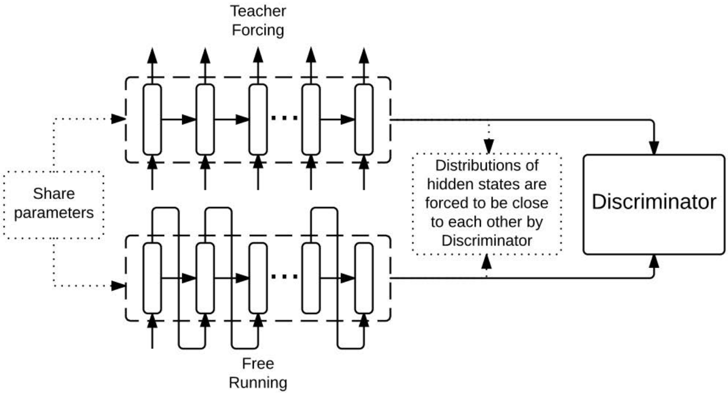


Figure 3: Match the dynamics of free running with teacher forcing. [3]

- Adversarial training paradigm: Discriminator is Bi-RNN + MLP.

D: $C_d(\boldsymbol{\theta}_d|\boldsymbol{\theta}_g) = E_{(\boldsymbol{x},\boldsymbol{y})\sim\text{data}}[-\log D(B(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}_g),\boldsymbol{\theta}_d) + E_{\boldsymbol{y}\sim P_{\boldsymbol{\theta}_g}(\boldsymbol{y}|\boldsymbol{x})}[-\log(1-D(B(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}_g),\boldsymbol{\theta}_d)]]$

G: $NLL(\boldsymbol{\theta}_g) = E_{(\boldsymbol{x},\boldsymbol{y})\sim\text{data}}[-\log P_{\boldsymbol{\theta}_g}(\boldsymbol{y}|\boldsymbol{x})]$

$C_f(\boldsymbol{\theta}_g|\boldsymbol{\theta}_d) = E_{\boldsymbol{x}\sim\text{data},\boldsymbol{y}\sim P_{\boldsymbol{\theta}_g}(\boldsymbol{y}|\boldsymbol{x})}[-\log D(B(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}_g),\boldsymbol{\theta}_d)]$

- Character-Level Language Modeling, Penn-Treebank:

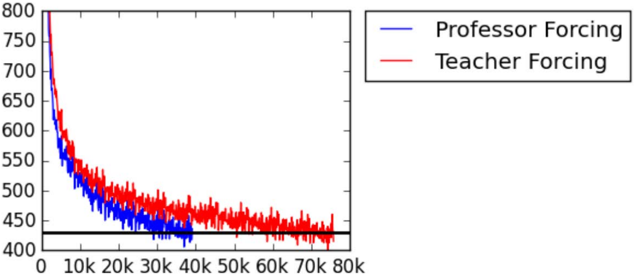


Figure 4: Training Negative Log-Likelihood.

- Training cost decreases faster.
- Training time is 3 times more.

Teacher Forcing

- Jang, Eric, Shixiang Gu, and Ben Poole. "**Categorical reparameterization with gumbel-softmax**." ICLR, 2017.
  Gu, Jiatao, Daniel Jiwoong Im, and Victor OK Li. "**Neural Machine Translation with Gumbel-Greedy Decoding**." arXiv (2017).

- The Gumbel-Max trick (Gumbel, 1954) provides a simple and efficient way to draw samples z from a categorical distribution with class probabilities π:

$$z = \texttt{one\_hot}\left(\arg\max_i \left[g_i + \log \pi_i\right]\right)$$

$$\downarrow$$

$$y_i = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_{j=1}^{k} \exp((\log(\pi_j) + g_j)/\tau)} \qquad \text{for } i = 1, \ldots, k$$

  Gumbel(0, 1): $u \sim \text{Uniform}(0, 1)$ and $g = -log(-log(u))$.
- Gumbel-Softmax is differentiable. Between softmax and one_hot.
- Example: Char-RNN.

- Teacher forcing is good enough.
- Teacher forcing is indispensable.

Outline

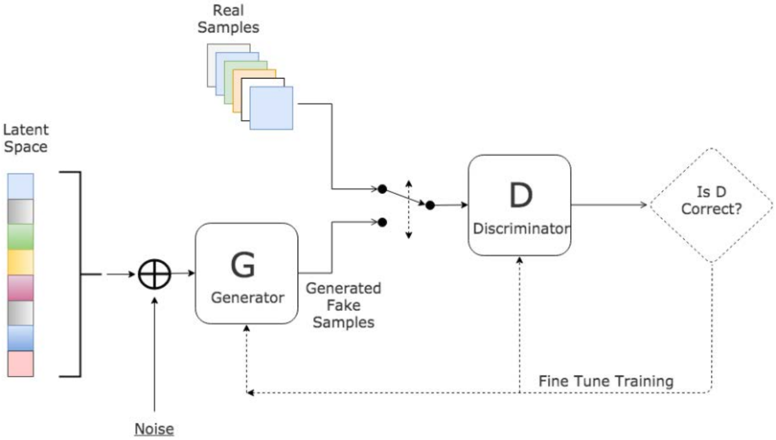
1 Introduction

2 Frameworks
- Overview
- Teacher Forcing
- Adversarial Reinforce
- Tricks

3 Applications

4 Conclusions

- Generative **Adversarial** Network (GAN) [2]:



[2] Source of figure: https://goo.gl/uPxWTs

Adversarial Training

- Bahdanau, Dzmitry, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. "**An actor-critic algorithm for sequence prediction**." arXiv 2016. (Basic work, connect AC with GAN)
- Yu, Lantao, Weinan Zhang, Jun Wang, and Yong Yu. "**SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient**." AAAI 2017.
- Li, Jiwei, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. "**Adversarial learning for neural dialogue generation**." EMNLP 2017.
- Wu, Lijun, Yingce Xia, Li Zhao, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. "**Adversarial Neural Machine Translation**." arXiv 2017.

- Yu, Lantao, Weinan Zhang, Jun Wang, and Yong Yu. "**SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient**." AAAI 2017.

- Overview of the framework:

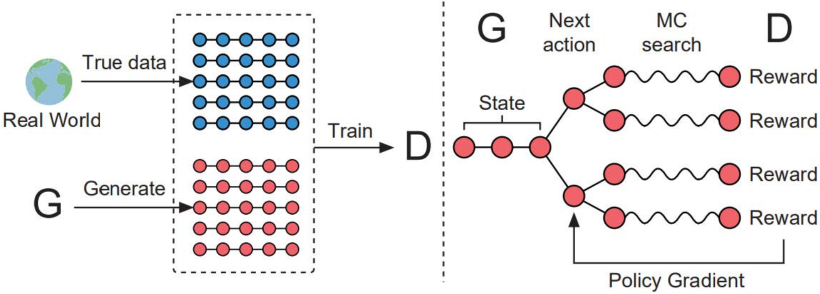


Figure 5: Left: D is trained over the real data and the generated data by G. Right: G is trained by policy gradient where the final reward signal is provided by D and is passed back to the intermediate action value via Monte Carlo search. [9]

- Discriminator: CNN (Highway)
- Policy Gradient:

$$\min_{\phi} -\mathbb{E}_{Y\sim p_{\text{data}}}[\log D_\phi(Y)] - \mathbb{E}_{Y\sim G_\theta}[\log(1 - D_\phi(Y))]$$

$$J(\theta) = \mathbb{E}[R_T|s_0,\theta] = \sum_{y_1\in\mathcal{Y}} G_\theta(y_1|s_0)\cdot Q_{D_\phi}^{G_\theta}(s_0,y_1)$$

$$Q_{D_\phi}^{G_\theta}(s=Y_{1:t-1},a=y_t) = \\ \begin{cases} \frac{1}{N}\sum_{n=1}^{N} D_\phi(Y_{1:T}^n),\ Y_{1:T}^n \in \mathrm{MC}^{G_\beta}(Y_{1:t};N) & \text{for} \quad t<T \\ D_\phi(Y_{1:t}) & \text{for} \quad t=T \end{cases}$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{Y_{1:t-1}\sim G_\theta}\Big[\sum_{y_t\in\mathcal{Y}} \nabla_\theta G_\theta(y_t|Y_{1:t-1})\cdot Q_{D_\phi}^{G_\theta}(Y_{1:t-1},y_t)\Big]$$

- (1) Pre-train the generator and discriminator. (2) Adversarial training

- Results on three tasks:

Table 2: Chinese poem generation performance comparison.

Algorithm	Human score	p-value	BLEU-2	p-value
MLE	0.4165	0.0034	0.6670	$< 10^{-6}$
SeqGAN	**0.5356**		**0.7389**	
Real data	0.6011		0.746	

Table 3: Obama political speech generation performance.

Algorithm	BLEU-3	p-value	BLEU-4	p-value
MLE	0.519	$< 10^{-6}$	0.416	0.00014
SeqGAN	**0.556**		**0.427**	

Table 4: Music generation performance comparison.

Algorithm	BLEU-4	p-value	MSE	p-value
MLE	0.9210	$< 10^{-6}$	22.38	0.00034
SeqGAN	**0.9406**		**20.62**	

- Policy Gradient: Wang, Jun, et. al. "**IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models**." SIGIR 2017.

- Li, Jiwei, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. "**Adversarial learning for neural dialogue generation**." EMNLP 2017.

- G: seq2seq.
- D: a hierarchical recurrent encoder.
- Training: policy gradient.
- Add teacher forcing back.

- Wu, Lijun, Yingce Xia, Li Zhao, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. "**Adversarial Neural Machine Translation**." arXiv 2017.

- G: seq2seq.
- D: CNN
- Training: policy gradient.

System	System Configurations	BLEU
	Representative end-to-end NMT systems	
Sutskever et al. (2014)	LSTM with 4 layers + $80K$ vocabs	30.59
Bahdanau et al. (2014)	RNNSearch	29.97[a]
Jean et al. (2015)	RNNSearch + UNK Replace	33.08
Jean et al. (2015)	RNNSearch + $500k$ vocabs + UNK Replace	34.11
Luong et al. (2015)	LSTM with 4 layers + $40K$ vocabs	29.50
Luong et al. (2015)	LSTM with 4 layers + $40K$ vocabs + PosUnk	31.80
Shen et al. (2016)	RNNSearch +Minimum Risk Training Objective	31.30
Sennrich et al. (2016)	RNNSearch +Monolingual Data	30.40 [b]
He et al. (2016)	RNNSearch+ Monolingual Data + Dual Objective	32.06
	Adversarial-NMT	
this work	RNNSearch + Adversarial Training Objective	31.91†
	RNNSearch + Adversarial Training Objective + UNK Replace	34.78

Figure 6: Different NMT systems' performances on En→Fr translation.

- Fine tuning.
- More robust.
- Difficult to train.

Outline

1 Introduction

2 Frameworks
- Overview
- Teacher Forcing
- Adversarial Reinforce
- Tricks

3 Applications

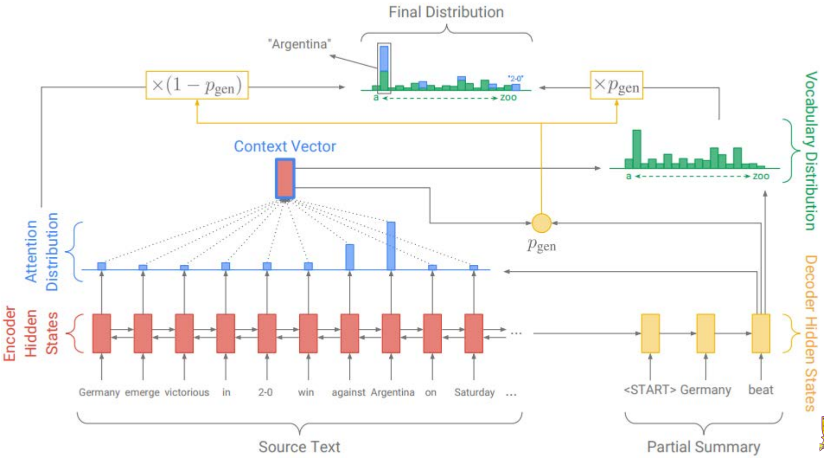
4 Conclusions

Tricks

- Copy mechanism.
- Coverage or diversity.
- Dual or reconstruction.
- CNN based seq2seq

- Gulcehre, Caglar, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. "**Pointing the unknown words**." arXiv 2016.
- Gu, Jiatao, Zhengdong Lu, Hang Li, and Victor OK Li. "**Incorporating copying mechanism in sequence-to-sequence learning**." ACL 2016.

- See, Abigail, et al. "Get To The Point: Summarization with Pointer-Generator Networks." ACL 2017. [7]



- Summarization results on DNN/DailyMail:

	ROUGE			METEOR	
	1	2	L	exact match	+ stem/syn/para
abstractive model (Nallapati et al., 2016)*	35.46	13.30	32.65	-	-
seq-to-seq + attn baseline (150k vocab)	30.49	11.17	28.08	11.65	12.86
seq-to-seq + attn baseline (50k vocab)	31.33	11.81	28.83	12.03	13.20
pointer-generator	36.44	15.66	33.42	15.35	16.65
pointer-generator + coverage	**39.53**	**17.28**	**36.38**	17.32	18.72
lead-3 baseline (ours)	40.34	17.70	36.57	20.48	22.21
lead-3 baseline (Nallapati et al., 2017)*	39.2	15.7	35.5	-	-
extractive model (Nallapati et al., 2017)*	39.6	16.2	35.3	-	-

- Significant improvement.

- Tu, Zhaopeng, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. "**Modeling coverage for neural machine translation**." ACL 2016.
- **Application**:
  - See, Abigail, Peter J. Liu, and Christopher D. Manning. "**Get To The Point: Summarization with Pointer-Generator Networks**." ACL 2017.
  - Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei and Hui Jiang. "**Distraction-Based Neural Networks for Document Summarization**." IJCAI 2016.

- Accumulation of the history attentions:

$$c^t = \sum_{t'=0}^{t-1} a^{t'}$$

$$e_i^t = v^T \tanh(W_h h_i + W_s s_t + w_c c_i^t + b_{\text{attn}})$$

- Summarization results on DNN/DailyMail:

	ROUGE			METEOR	
	1	2	L	exact match	+ stem/syn/para
abstractive model (Nallapati et al., 2016)*	35.46	13.30	32.65	-	-
seq-to-seq + attn baseline (150k vocab)	30.49	11.17	28.08	11.65	12.86
seq-to-seq + attn baseline (50k vocab)	31.33	11.81	28.83	12.03	13.20
pointer-generator	36.44	15.66	33.42	15.35	16.65
pointer-generator + coverage	**39.53**	**17.28**	**36.38**	17.32	18.72
lead-3 baseline (ours)	40.34	17.70	36.57	20.48	22.21
lead-3 baseline (Nallapati et al., 2017)*	39.2	15.7	35.5	-	-
extractive model (Nallapati et al., 2017)*	39.6	16.2	35.3	-	-

- Significant improvement.

- A $\rightarrow$ B $\rightarrow$ A
- Works:
  - Tu, Zhaopeng, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. "**Neural Machine Translation with Reconstruction**." AAAI 2017.
  - He, Di, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tieyan Liu, and Wei-Ying Ma. "**Dual learning for machine translation**." NIPS 2016.
  - Xia, Yingce, Tao Qin, Wei Chen, Jiang Bian, Nenghai Yu, and Tie-Yan Liu. "**Dual Supervised Learning**." ICML 2017.
- Paraphrase generation; Image $\rightarrow$ caption $\rightarrow$ image, etc.

- Gehring, Jonas, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. "**Convolutional Sequence to Sequence Learning**." arXiv 2017.
- CNN n-gram
- Attention mechanism.
- Language model in decoder.
- Teacher forcing.

- Tricks $\rightarrow$ Performance.

Applications

- See, Abigail, Peter J. Liu, and Christopher D. Manning. "**Get To The Point: Summarization with Pointer-Generator Networks**." ACL 2017.
- Du, Xinya, Junru Shao, and Claire Cardie. "**Learning to Ask: Neural Question Generation for Reading Comprehension**." ACL 2017.
- Meng, Rui, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. "**Deep Keyphrase Generation**." ACL 2017.

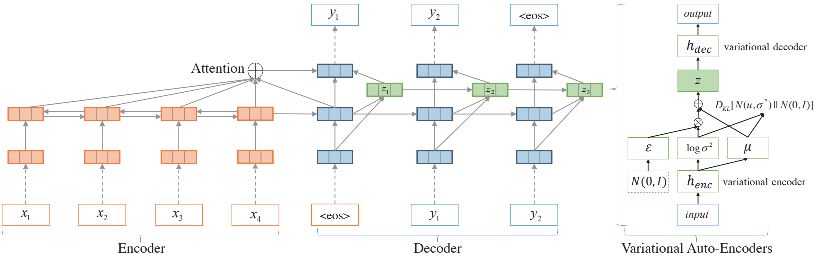
- Sequence to sequence with attention modeling.
- Input:

  X: 扬帆远东做与中国合作的先行。<eos>

  Y: 扬帆<eow>远东<eow>做<eow>与<eow>中国<eow>合作<eow>的<eow>先行<eow>。<eow><eos>

- icwb2: sighan bakeoff2005.
- MSR: Recall = 0.956, Precision = 0.956, F1-Measure = 0.956
  PKU: Recall = 0.911, Precision = 0.920, F1-Measure = 0.915
- https://github.com/lipiji/cws-seq2seq

- Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. **Deep Recurrent Generative Decoder for Abstractive Text Summarization**. EMNLP 2017. [5]



- Evaluation results on Gigawords:

Table 1: ROUGE-F1 on Gigawords

System	**R-1**	**R-2**	**R-L**
ABS	29.55	11.32	26.42
ABS+	29.78	11.89	26.97
RAS-LSTM	32.55	14.70	30.03
RAS-Elman	33.78	15.97	31.15
ASC + FSC_1	34.17	15.94	31.92
lvt2k-1sent	32.67	15.59	30.64
lvt5k-1sent	35.30	16.64	32.62
DRGD	**36.27**	**17.57**	**33.62**

- Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. **Neural Rating Regression with Abstractive Tips Generation for Recommendation**. SIGIR 2017. [6]

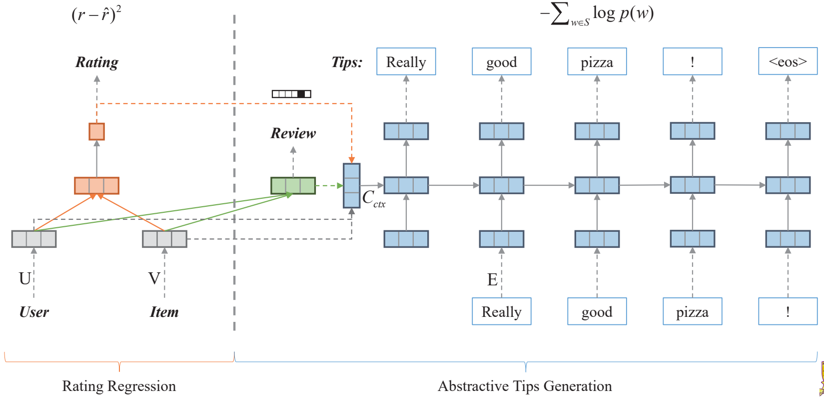


Table 2: **MAE** and **RMSE** values for rating prediction.

	Books		Electronics		Movies		Yelp-2016	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
LRMF	1.939	2.153	2.005	2.203	1.977	2.189	1.809	2.038
PMF	0.882	1.219	1.220	1.612	0.927	1.290	1.320	1.752
NMF	0.731	1.035	0.904	1.297	0.794	1.135	1.062	1.454
SVD++	0.686	0.967	0.847	1.194	0.745	1.049	1.020	1.349
URP	0.704	0.945	0.860	1.126	0.764	1.006	1.030	1.286
CTR	0.736	0.961	0.903	1.154	0.854	1.069	1.174	1.392
RMR	0.681	0.933	0.822	1.123	0.741	1.005	0.994	1.286
NRT	**0.667***	**0.927***	**0.806***	**1.107***	**0.702***	**0.985***	**0.985***	**1.277***

Table 3: ROUGE evaluation on dataset Books.

Methods	ROUGE-1			ROUGE-2			ROUGE-L			ROUGE-SU4		
	R	P	F1	R	P	F1	R	P	F1	R	P	F1
LexRank	12.94	12.02	12.18	2.26	2.29	2.23	11.72	10.89	11.02	4.13	4.15	4.02
RMR_t	13.80	11.69	12.43	1.79	1.57	1.64	12.54	10.55	11.25	4.49	3.54	3.80
CTR_t	14.06	11.85	12.62	2.03	1.80	1.87	12.68	10.64	11.35	4.71	3.71	3.99
NRT	10.30	**19.28**	**12.67**	1.91	**3.76**	**2.36**	9.71	**17.92**	**11.88**	3.24	**8.03**	**4.13**

Table 4: ROUGE evaluation on dataset Electronics.

Methods	ROUGE-1			ROUGE-2			ROUGE-L			ROUGE-SU4		
	R	P	F1	R	P	F1	R	P	F1	R	P	F1
LexRank	13.42	13.48	12.08	1.90	2.04	1.83	11.72	11.48	10.44	4.57	4.51	3.88
RMR_t	15.68	11.32	12.30	2.52	2.04	2.15	13.37	9.61	10.45	5.41	3.72	3.97
CTR_t	15.81	11.37	12.38	2.49	1.92	2.05	13.45	9.62	10.50	5.39	3.63	3.89
NRT	13.08	**17.72**	**13.95**	**2.59**	**3.36**	**2.72**	11.93	**16.01**	**12.67**	4.51	**6.69**	**4.68**

Table 5: ROUGE evaluation on dataset Movies&TV.

Methods	ROUGE-1			ROUGE-2			ROUGE-L			ROUGE-SU4		
	R	P	F1	R	P	F1	R	P	F1	R	P	F1
LexRank	13.62	14.11	12.37	1.92	2.09	1.81	11.69	11.74	10.47	4.47	4.53	3.75
RMR_t	14.64	10.26	11.33	1.78	1.36	1.46	12.62	8.72	9.67	4.63	3.00	3.28
CTR_t	15.13	10.37	11.57	1.90	1.42	1.54	13.02	8.77	9.85	4.88	3.03	3.36
NRT	**15.17**	**20.22**	**16.20**	**4.25**	**5.72**	**4.56**	**13.82**	**18.36**	**14.73**	**6.04**	**8.76**	**6.33**

Table 6: ROUGE evaluation on dataset Yelp-2016.

Methods	ROUGE-1			ROUGE-2			ROUGE-L			ROUGE-SU4		
	R	P	F1	R	P	F1	R	P	F1	R	P	F1
LexRank	11.32	11.16	11.04	1.32	1.34	1.31	10.33	10.16	10.06	3.41	3.38	3.26
RMR_t	11.17	10.25	10.54	2.25	2.16	2.19	10.22	9.39	9.65	3.88	3.66	3.72
CTR_t	10.74	9.95	10.19	2.21	2.14	2.15	9.91	9.19	9.41	3.96	3.64	3.70
NRT	9.39	**17.75**	**11.64**	1.83	**3.39**	**2.22**	8.70	**16.27**	**10.74**	3.01	**7.06**	**3.78**

Applications

Rating Prediction and Tips Generation - Case Analysis

Table 7: Examples of the predicted ratings and the generated tips.

Rating	Tips
4.64	**This is a great product for a great price.**
5	Great product at a great price.
4.87	**I purchased this as a replacement and it is a perfect fit and the sound is excellent.**
5	Amazing sound.
4.87	**One of my favorite movies.**
5	This is a movie that is not to be missed.
4.07	**Why do people hate this film.**
4	Universal why didnt your company release this edition in 1999.
2.25	**Not as good as i expected.**
5	Jack of all trades master of none.
1.46	**What a waste of time and money.**
1	The coen brothers are two sick bastards.
4.34	**Not bad for the price.**
3	Ended up altering it to get rid of ripples.

Conclusions

Conclusions

- Teacher forcing.
- Adversarial reinfoce.
- Tricks (copy, coverage, dual training, etc.).
- Applications.

References I

[1] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179, 2015.

[2] J. Gu, D. J. Im, and V. O. Li. Neural machine translation with gumbel-greedy decoding. *arXiv preprint arXiv:1706.07518*, 2017.

[3] A. M. Lamb, A. G. A. P. GOYAL, Y. Zhang, S. Zhang, A. C. Courville, and Y. Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, pages 4601–4609, 2016.

[4] J. Li, W. Monroe, T. Shi, A. Ritter, and D. Jurafsky. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*, 2017.

[5] P. Li, W. Lam, L. Bing, and Z. Wang. Deep recurrent generative decoder for abstractive text summarization. *EMNLP*, 2017.

References II

[6] P. Li, Z. Wang, Z. Ren, L. Bing, and W. Lam. Neural rating regression with abstractive tips generation for recommendation. *SIGIR*, 2017.

[7] A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. *ACL*, 2017.

[8] L. Wu, Y. Xia, L. Zhao, F. Tian, T. Qin, J. Lai, and T.-Y. Liu. Adversarial neural machine translation. *arXiv preprint arXiv:1704.06933*, 2017.

[9] L. Yu, W. Zhang, J. Wang, and Y. Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858, 2017.

Thanks a lot!
Q & A