TEXT GENERATION

Zhangming Chan

OUTLINE

- Word Selection
- Text Generation Assist
- Story Generation

WORD SELECTION

- Bag-of-Words as Target for Neural Machine Translation
- Faithful to the Original: Fact Aware Neural Abstractive Summarization

BAG-OF-WORDS AS TARGET FOR NEURAL MACHINE TRANSLATION ACL 2018

Model

$$x^i = \{x_1^i, x_2^i, ..., x_{L_i}^i\}$$

$$y^i = \{y_1^i, y_2^i, ..., y_{M_i}^i\}$$

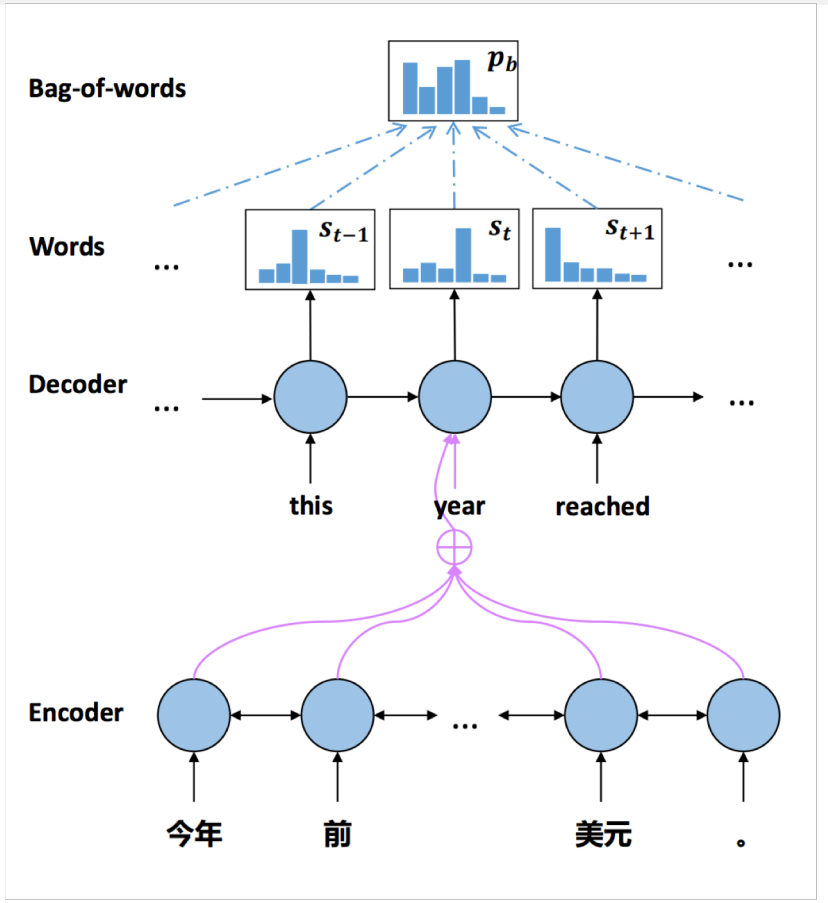
$$b^i = \{b_1^i, b_2^i, ..., b_{K_i}^i\}$$

$$l_1 = -\sum_{t=1}^{M} y_t \log p_{w_t}(y_t)$$

$$l_2 = -\sum_{i=1}^{K} b_i \log p_b(b_i)$$

$$l = l_1 + \lambda_i l_2$$

$$\lambda_i = min(\lambda, k + \alpha i)$$



Experiment

Model	MT-02	MT-03	MT-04	MT-05	MT-06	MT-08	All
Moses (Su et al., 2016)	33.19	32.43	34.14	31.47	30.81	23.85	31.04
RNNSearch (Su et al., 2016)	34.68	33.08	35.32	31.42	31.61	23.58	31.76
Lattice (Su et al., 2016)	35.94	34.32	36.50	32.40	32.77	24.84	32.95
CPR (Zhang et al., 2017)	33.84	31.18	33.26	30.67	29.63	22.38	29.72
POSTREG (Zhang et al., 2017)	34.37	31.42	34.18	30.99	29.90	22.87	30.20
PKI (Zhang et al., 2017)	36.10	33.64	36.48	33.08	32.90	24.63	32.51
Bi-Tree-LSTM (Chen et al., 2017)	36.57	35.64	36.63	34.35	30.57	-	-
Mixed RNN (Li et al., 2017)	37.70	34.90	38.60	35.50	35.60	-	-
Seq2Seq+Attn (our implementation)	34.71	33.15	35.26	32.36	32.45	23.96	31.96
+Bag-of-Words (this paper)	**39.77**	**38.91**	**40.02**	**36.82**	**35.93**	**27.61**	**36.51**

Table 2: Results of our model and the baselines (directly reported in the referred articles) on the Chinese-English translation. "-" means that the studies did not test the models on the corresponding datasets.

FAITHFUL TO THE ORIGINAL: FACT AWARE NEURAL ABSTRACTIVE SUMMARIZATION AAAI 2018

Motivation

Source	the repatriation of at least #,### bosnian moslems was postponed friday after the unhcr pulled out of the first joint scheme to return refugees to their homes in northwest bosnia .
Target	**repatriation** of bosnian moslems postponed
s2s	**bosnian moslems** postponed after unhcr pulled out of **bosnia**

Table 1: An example of fake summaries generated by the state-of-the-art s2s model. "#" stands for a digit masked during preprocessing.

faithfulness is also a vital prerequisite for a practical abstractive summarization system.

Model

Sentence	I saw a cat sitting on the desk
Triples	(I; saw; cat)
	(I; saw; cat sitting)
	(I; saw; cat sitting on desk)

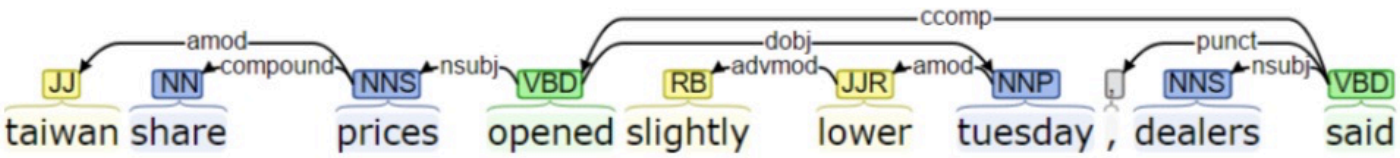
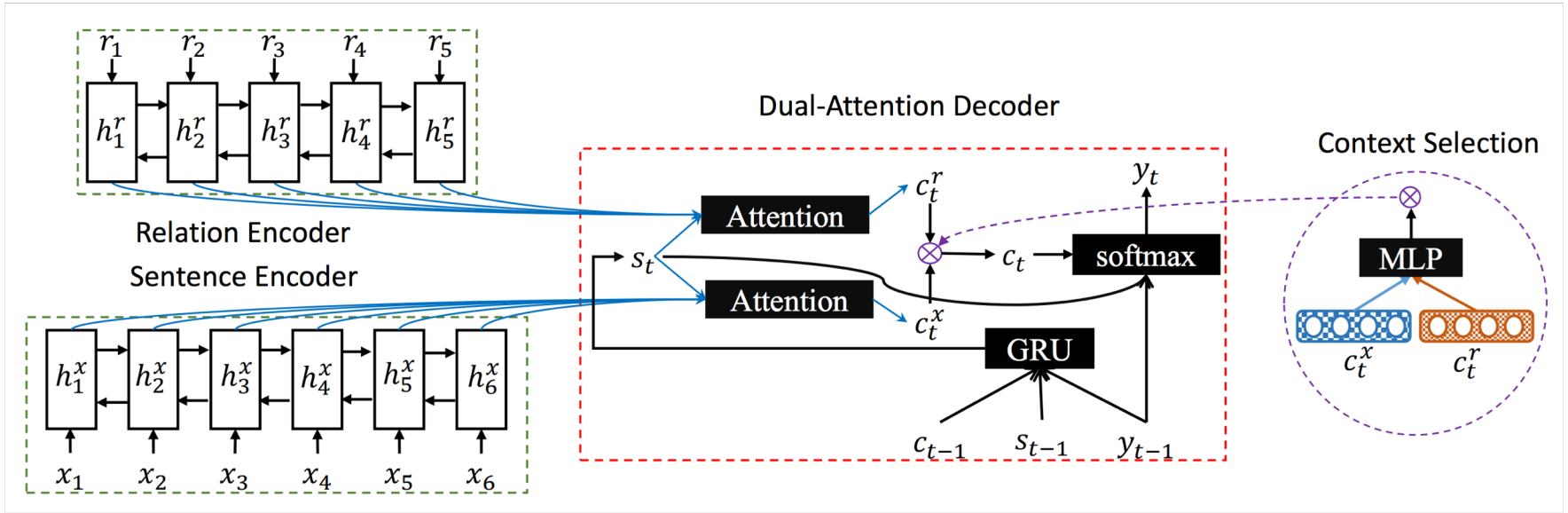


Figure 1: A dependency tree example. The meaning of the dependency labels can be referred to (De Marneffe and Manning 2008). We extract the following two fact descriptions: *taiwan share prices opened lower tuesday* ||| *dealers said*

Model


Dual-Attention Decoder
Context Selection
Relation Encoder
Sentence Encoder
Attention
Attention
softmax
GRU
MLP

Experiment

Model	Perplexity
ABS†	27.1
RAS-Elman†	18.9
s2s-att	24.5
FTSum$_c$	20.1
FTSum$_g$	**16.4**

Table 5: Final perplexity on the development set. † indicates the value is cited from the corresponding paper. ABS+, Feats2s and Luong-NMT do not provide this value.

Model	RG-1	RG-2	RG-L
ABS†	29.55*	11.32*	26.42*
ABS+†	29.78*	11.89*	26.97*
Feats2s†	32.67*	15.59*	30.64*
RAS-Elman†	33.78*	15.97*	31.15*
Luong-NMT†	33.10*	14.45*	30.71*
s2s+att	34.23*	15.52*	31.57*
FTSum$_c$	35.73*	16.02*	34.13
FTSum$_g$	**37.27**	**17.65**	**34.24**

Table 6: ROUGE F1 performance. "*" indicates statistical significance of the corresponding model with respect to the baseline model on the 95% confidence interval in the official ROUGE script. RG refers to ROUGE for short.

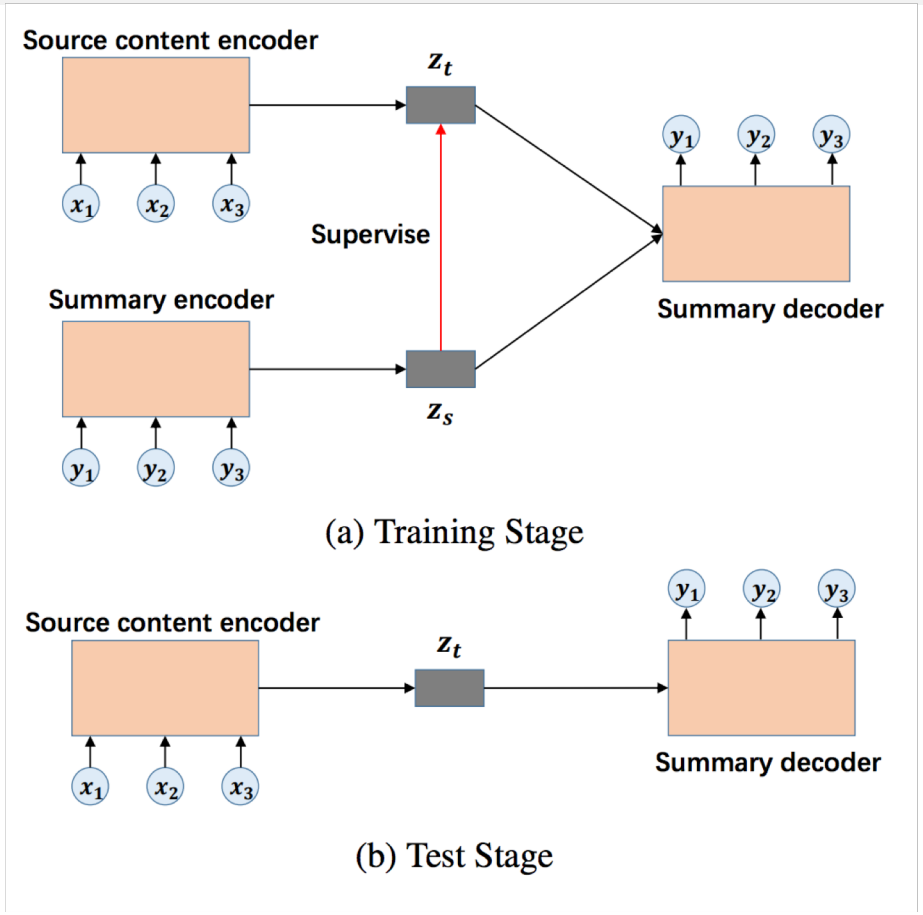
Q & A

TEXT GENERATION ASSIST

- Autoencoder as Assistant Supervisor-Improving Text Representation for Chinese Social Media Text Summarization
- Retrieve, Rerank and Rewrite-Soft Template Based Neural Summarization
- Learning to Write with Cooperative Discriminators

AUTOENCODER AS ASSISTANT SUPERVISOR-IMPROVING TEXT REPRESENTATION FOR CHINESE SOCIAL MEDIA TEXT SUMMARIZATION
ACL 2018

Model



$$d(z_t, z_s) = \|z_t - z_s\|_2$$

$$L_D(\theta_D) = -\log P_{\theta_D}(y = 1|z_t) - \log P_{\theta_D}(y = 0|z_s)$$

Experiment

Models	R-1	R-2	R-L
RNN-W(Hu et al., 2015)	17.7	8.5	15.8
RNN(Hu et al., 2015)	21.5	8.9	18.6
RNN-cont-W(Hu et al., 2015)	26.8	16.1	24.1
RNN-cont(Hu et al., 2015)	29.9	17.4	27.2
SRB(Ma et al., 2017)	33.3	20.0	30.1
CopyNet-W(Gu et al., 2016)	35.0	22.3	32.0
CopyNet(Gu et al., 2016)	34.4	21.6	31.3
RNN-dist(Chen et al., 2016)	35.2	22.6	32.5
DRGD(Li et al., 2017)	37.0	24.2	34.2
Seq2Seq (our impl.)	32.1	19.9	29.2
+superAE (this paper)	**39.2**	**26.0**	**36.2**
w/o adversarial learning	37.7	25.3	35.2

Table 1: Comparison with state-of-the-art models on the LCSTS test set. R-1, R-2, and R-L denote ROUGE-1, ROUGE-2, and ROUGE-L, respectively. The models with a suffix of 'W' in the table are word-based, while the rest of models are character-based.

Models	2-class (%)	5-class (%)
Seq2seq	80.7	65.1
+superAE	**88.8 (+8.1)**	**71.7 (+6.6)**

Table 2: Accuracy of the sentiment classification on the Amazon dataset. We train a classifier which inputs internal representation provided by the sequence-to-sequence model, and outputs a predicted label. We compute the 2-class and 5-class accuracy of the predicted labels to evaluate the quality of the text representation.

RETRIEVE, RERANK AND REWRITE-SOFT TEMPLATE BASED NEURAL SUMMARIZATION
ACL 2018

Model

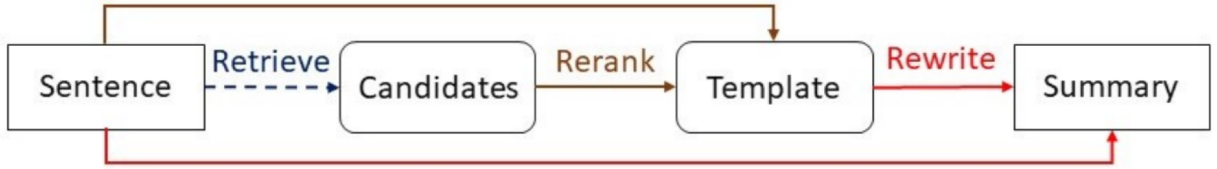


Figure 1: Flow chat of the proposed method. We use the dashed line for Retrieve since there is an IR system embedded.

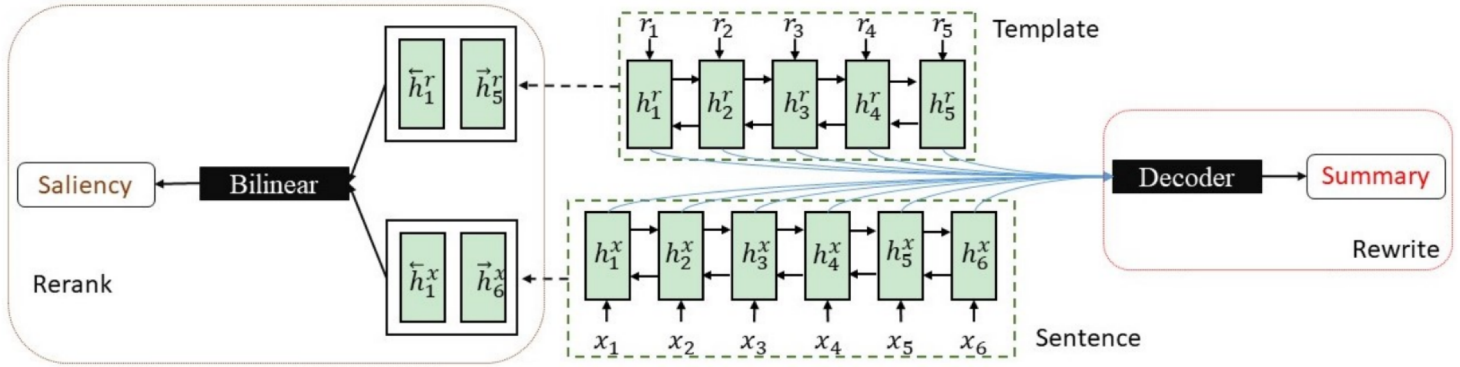


Figure 2: Jointly Rerank and Rewrite

Experiment

Model	RG-1	RG-2	RG-L
ABS†	29.55*	11.32*	26.42*
ABS+†	29.78*	11.89*	26.97*
Featseq2seq†	32.67*	15.59*	30.64*
RAS-Elman†	33.78*	15.97*	31.15*
Luong-NMT†	33.10*	14.45*	30.71*
FTSum†	**37.27**	17.65*	34.24
OpenNMT$_O^\dagger$	33.13*	16.09*	31.00*
OpenNMT$_I$	35.01*	16.55*	32.42*
PIPELINE	36.49	17.48*	33.90
Re^{3}Sum	37.04	**19.03**	**34.46**

Table 3: ROUGE F1 (%) performance. "RG" represents "ROUGE" for short. "*" indicates statistical significance of the corresponding model with respect to the baseline model on the 95% confidence interval in the official ROUGE script.

Type	RG-1	RG-2	RG-L
Random	2.81	0.00	2.72
First	24.44	9.63	22.05
Max	38.90	19.22	35.54
Optimal	52.91	31.92	48.63
Rerank	28.77	12.49	26.40

Table 4: ROUGE F1 (%) performance of different types of soft templates.

Type	RG-1	RG-2	RG-L
+Random	32.60	14.31	30.19
+First	36.01	17.06	33.21
+Max	41.50	21.97	38.80
+Optimal	46.21	26.71	43.19
+Rerank(Re^{3}Sum)	37.04	19.03	34.46

Table 6: ROUGE F1 (%) performance of Re^{3}Sum generated with different soft templates.

LEARNING TO WRITE WITH COOPERATIVE DISCRIMINATORS
ACL 2018

Motivations

When RNN-LM is used for long-form text generation, RNNs often lead to degenerate text that is :

- Repetitive
- Self-contradictory
- Overly generic

Context: The two guards thudded into the door on the other side and began pounding on it, shouting furiously. Scious raised the large bunch of keys then placed them in a large pocket in his coat. "Come, we have to go," he whispered and moved up the corridor, the guards still hammering on the door. Jared looked around. The table next to the room they had been in was bare.

LM: There was a small table set into the center of the table, and a table with two chairs and a table. A chair sat next to the couch. A large man stood in the doorway of the room, his arms crossed over his chest. "What 's up?" He asked.

The Learning Framework

- Base Language Model
- Cooperative Communication Models
  - Repetition Model
  - Entailment Model
  - Relevance Model
  - Lexical Style Model
- Mixture Weight Learning
- Beam Search

$$f_\lambda(\mathbf{x}, \mathbf{y}) = \log(P_{\text{lm}}(\mathbf{y}|\mathbf{x})) + \sum_k \lambda_k s_k(\mathbf{x}, \mathbf{y}), \quad (1)$$

Repetition Model

$$d_i = \max_{j=i-k \ldots i-1} (\text{CosSim}(e(y_j), e(y_i))),$$

$$s_{\text{rep}}(\mathbf{y}) = \sigma(\mathbf{w}_r^\top \text{RNN}_{\text{rep}}(\mathbf{d})),$$

$$L_{\text{rep}} = \sum_{\substack{(\mathbf{x},\mathbf{y}_g) \in D, \\ \mathbf{y}_s \sim \text{LM}(\mathbf{x})}} \log \sigma(s_{\text{rep}}(\mathbf{y}_g) - s_{\text{rep}}(\mathbf{y}_s)),$$

Entailment Model

$$s_{\text{entail}}(\mathbf{x}, \mathbf{y}) = min_{\mathbf{a} \in S(\mathbf{x}) \cup S_{\text{init}}(\mathbf{y})} t(\mathbf{a}, S_{\text{last}}(\mathbf{y})).$$

If the current sentence entails a previous one it may simply be adding more specific information, for instance: “He hated broccoli. Every time he ate broccoli he was reminded that it was the thing he hated most.”

Lexical Style Model

$$s_{\text{bow}}(\mathbf{y}) = \mathbf{w}_s^T \text{maxpool}(e(\mathbf{y})).$$

Mixture Weight Learning

$$L_{\text{mix}} = \sum_{(\mathbf{x},\mathbf{y})\in D} (f_\lambda(\mathbf{x},\mathbf{y}) - f_\lambda(\mathbf{x},\mathcal{A}(\mathbf{x}))^2,$$

Beam Search

At each step, the discriminator scores are recomputed for all candidates, with the exception of the entailment score, which is only recomputed for hypotheses which end with a sentence terminating symbol.

Data: context $\mathbf{x}$, beam size k, sampling temperature t
Result: best continuation
best = None
beam = [$\mathbf{x}$]
for step = 0; step $<$ max_steps; step = step $+1$ **do**
  next_beam = []
  for candidate in beam **do**
    next_beam.extend(next_k(candidate))
    if termination_score(candidate) $>$ best.score **then**
      best = candidate.append(term)
    end
  end
  for candidate in next_beam **do**
    ▷ score with models
    candidate.score += f_λ(candidate)
  end
  ▷ sample k candidates by score
  beam = sample(next_beam, k, t)
end
if learning **then**
  update λ with gradient descent by comparing best against the gold.
end
return best

Experiments

	BookCorpus					TripAdvisor				
Model	BLEU	Meteor	Length	Vocab	Trigrams	BLEU	Meteor	Length	Vocab %	Trigrams
L2W	**0.52**	**6.8**	**43.6**	**73.8**	98.9	1.7	11.0	83.8	**64.1**	**96.2**
ADAPTIVELM	**0.52**	6.3	43.5	59.0	92.7	**1.94**	**11.2**	**94.1**	52.6	92.5
CACHELM	0.33	4.6	37.9	31.0	44.9	1.36	7.2	52.1	39.2	57.0
SEQ2SEQ	0.32	4.0	36.7	23.0	33.7	1.84	8.0	59.2	33.9	57.0
SEQGAN	0.18	5.0	28.4	73.4	**99.3**	0.73	6.7	47.0	57.6	93.4
REFERENCE	100.0	100.0	65.9	73.3	99.7	100.0	100.0	92.8	69.4	99.4

Table 1: Results for automatic evaluation metrics for all systems and domains, using the original continuation as the reference. The metrics are: Length - Average total length per example; Trigrams - % unique trigrams per example; Vocab - % unique words per example.

Trip Advisor Ablation

Ablation vs. LM	Repetition	Contradiction	Relevance	Clarity	Better	Neither	Worse
REPETITION ONLY	+0.63	+0.30	+0.37	+0.42	**50%**	23%	27%
ENTAILMENT ONLY	+0.01	+0.02	+0.05	-0.10	39%	20%	**41%**
RELEVANCE ONLY	-0.19	+0.09	+0.10	+0.060	36%	22%	**42%**
LEXICAL STYLE ONLY	+0.11	+0.16	+0.20	+0.16	38%	25%	**38%**
ALL	+0.23	-0.02	+0.19	-0.03	**47%**	19%	34%

Table 4: Crowd-sourced ablation evaluation of generations on TripAdvisor. Each ablation uses only one discriminative communication model, and is compared to ADAPTIVELM.

Experiments

BookCorpus	**Specific Criteria**				**Overall Quality**		
L2W vs.	Repetition	Contradiction	Relevance	Clarity	Better	Equal	Worse
AdaptiveLM	+0.48	+0.18	+0.12	+0.11	**47%**	20%	32%
CacheLM	+1.61	+0.37	+1.23	+1.21	**86%**	6%	8%
Seq2Seq	+1.01	+0.54	+0.83	+0.83	**72%**	7%	21%
SeqGAN	+0.20	+0.32	+0.61	+0.62	**63%**	20%	17%
LM vs. Reference	-0.10	-0.07	-0.18	-0.10	41%	7 %	**52%**
L2W vs. Reference	+0.49	+0.37	+0.46	+0.55	**53%**	18%	29%

TripAdvisor	**Specific Criteria**				**Overall Quality**		
L2W vs.	Repetition	Contradiction	Relevance	Clarity	Better	Equal	Worse
AdaptiveLM	+0.23	-0.02	+0.19	-0.03	**47%**	19%	34%
CacheLM	+1.25	+0.12	+0.94	+0.69	**77%**	9%	14%
Seq2Seq	+0.64	+0.04	+0.50	+0.41	**58%**	12%	30%
SeqGAN	+0.53	+0.01	+0.49	+0.06	**55%**	22%	22%
LM vs. Reference	-0.10	-0.04	-0.15	-0.06	38%	10%	**52%**
L2W vs. Reference	-0.49	-0.36	-0.47	-0.50	25%	18%	**57%**

Table 2: Results of crowd-sourced evaluation on different aspects of the generation quality as well as overall quality judgments. For each sub-criteria we report the average of comparative scores on a scale from -2 to 2. For the overall quality evaluation decisions are aggregated over 3 annotators per example.

Q & A

STORY GENERATION

- Neural Text Generation in Stories Using Entity Representations as Context
- A Skeleton-Based Model for Promoting Coherence Among Sentences in Narrative Story Generation
- Dixit: Interactive Visual Storytelling via Term Manipulation

NEURAL TEXT GENERATION IN STORIES USING ENTITY REPRESENTATIONS AS CONTEXT NAACL 2018

Motivation

Context	All of a sudden, [*Emily*]$_1$ walked towards [*the dragon*]$_2$.
Current Sentence	[*Seth*]$_3$ yelled at [*her*]$_1$ to get back but _____

Figure 1: An example of entity-labeled story data. The brackets indicate which words are part of entity mentions. Mentions marked with the same number refer to the same entity. The goal is to continue the story in a coherent way. The actual story reads, "*Seth yelled at her to get back but she ignored him.*"

Model

$$\exp(\boldsymbol{h}_{t-1}^{\top}\mathbf{W}_{entity}\boldsymbol{e}_{i,t-1} + \boldsymbol{w}_{dist}^{\top}\boldsymbol{f}(i)),$$

$$\boldsymbol{c}_t[k] = \max(\boldsymbol{h}_{t-1}[k], \boldsymbol{p}_t[k], \boldsymbol{e}_{current}[k]).$$

Experiments

cluster and mention	cluster only	mention only
[*Emily*]$_1$	*EMILY	Emily
[*the dragon*]$_2$	THE DRAGON	the dragon
[*Seth*]$_3$	SETH	Seth
[*her*]$_1$		her
*[*she*]$_1$		*she

Figure 2: Candidate lists for each of the mention generation tasks for completing the blank in Figure 1. The asterisk (*) indicates the correct choice.

Context	All of a sudden, [*Emily*]$_1$ walked towards [*the dragon*]$_2$.
1.	[*Seth*]$_3$ yelled at [*her*]$_1$ to get back but [*she*]$_1$ ignored [*him*]$_3$.
2.	[*She*]$_1$ patted [*its head*]$_4$ and [*it*]$_2$ curled up outside [*the cave*]$_5$.
3.	"[*Emily*]$_1$, how did [*you*]$_1$ keep [*that dragon*]$_2$ from attacking [*us*]$_6$?"

Figure 3: A passage's last sentence of context, and 3 sentences from various points in the next passage.

A SKELETON-BASED MODEL FOR PROMOTING COHERENCE AMONG SENTENCES IN NARRATIVE STORY GENERATION
EMNLP 2018

Motivation

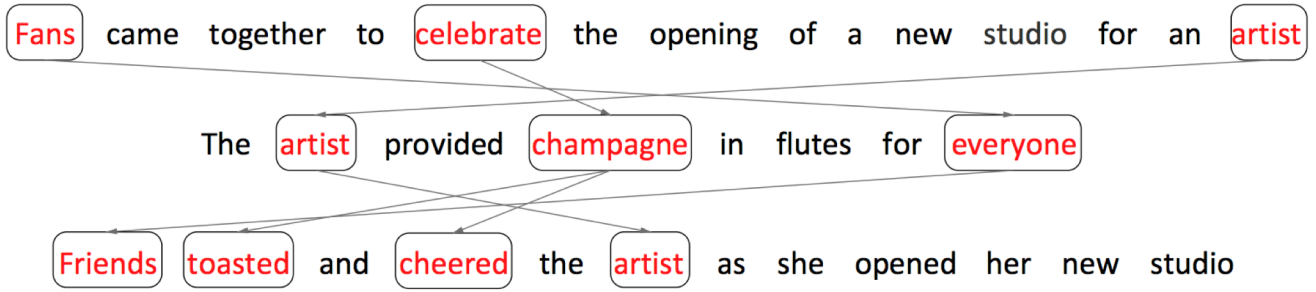
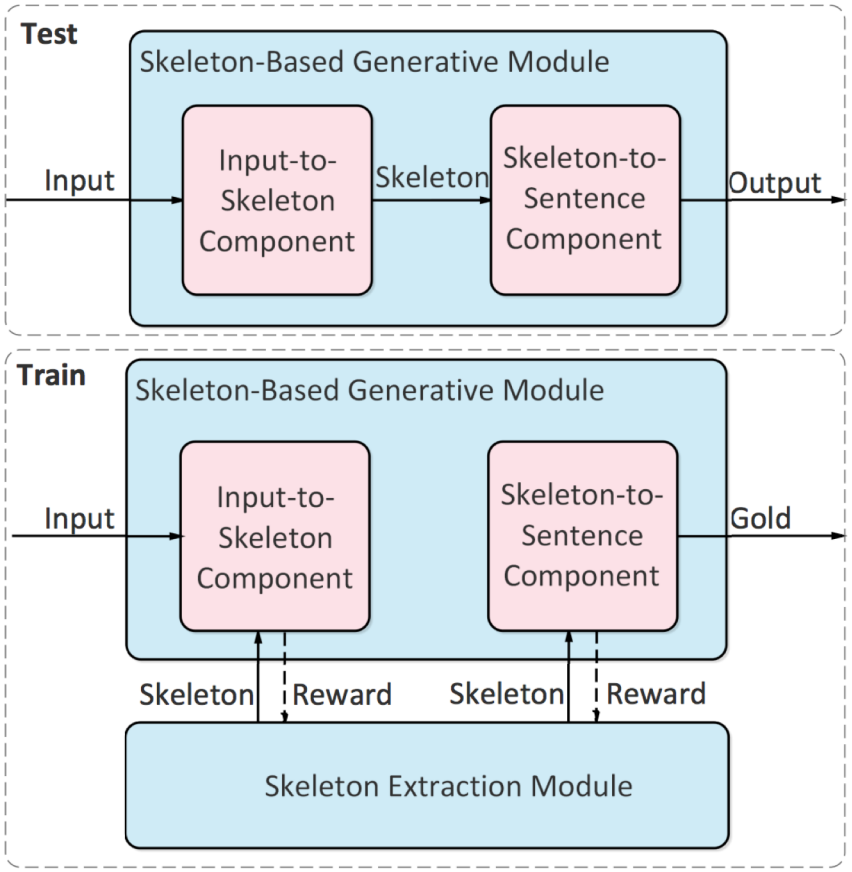


Figure 1: The semantic dependency among sentences in a narrative story. It can be seen that the connection among sentences is mainly reflected through key phrases (shown in red). In this work, we regard such key phrases as a skeleton.

Model



$$L_\alpha = -\sum_{i=1}^{T} P_Q(s_i|\boldsymbol{c}, \alpha)$$

$$L_\theta = -\sum_{i=1}^{M} P_D(y_i|\boldsymbol{s}, \theta)$$

$$\nabla J(\gamma) = \mathbb{E}[R_c \cdot \nabla \log(P_E(\boldsymbol{s}|\boldsymbol{x}), \gamma)]$$

$$R_c = [K - (R_1 \times R_2)^{\frac{1}{2}}]$$

Experiment

Models	BLEU
EE-Seq2Seq	0.0029
DE-Seq2Seq	0.0027
GE-Seq2Seq	0.0022
Proposed Model	**0.0042 (+44.8%)**

Table 2: Automatic evaluations of the proposed model and the state-of-the-art models.

Models	Fluency	Coherence	G-Score
EE-Seq2Seq	6.28	5.14	5.68
DE-Seq2Seq	8.48	3.54	5.48
GE-Seq2Seq	9.48	3.58	5.82
Proposed Model	8.69	5.62	**6.99 (+20.1%)**

Table 3: Human evaluations of the proposed model and the state-of-the-art models.

Models	BLEU
Seq2Seq	0.0028
+Skeleton Extraction Module	0.0029
+Reinforcement Learning	**0.0042**

Table 5: Automatic evaluations of key components.

Models	Fluency	Coherence	G-Score
Seq2Seq	7.54	4.98	6.13
+Skeleton Extraction Module	7.26	4.32	5.60
+Reinforcement Learning	8.69	5.62	**6.99**

Table 6: Human evaluations of the key components.

DIXIT: INTERACTIVE VISUAL STORYTELLING VIA TERM MANIPULATION
WWW 2018

Framework

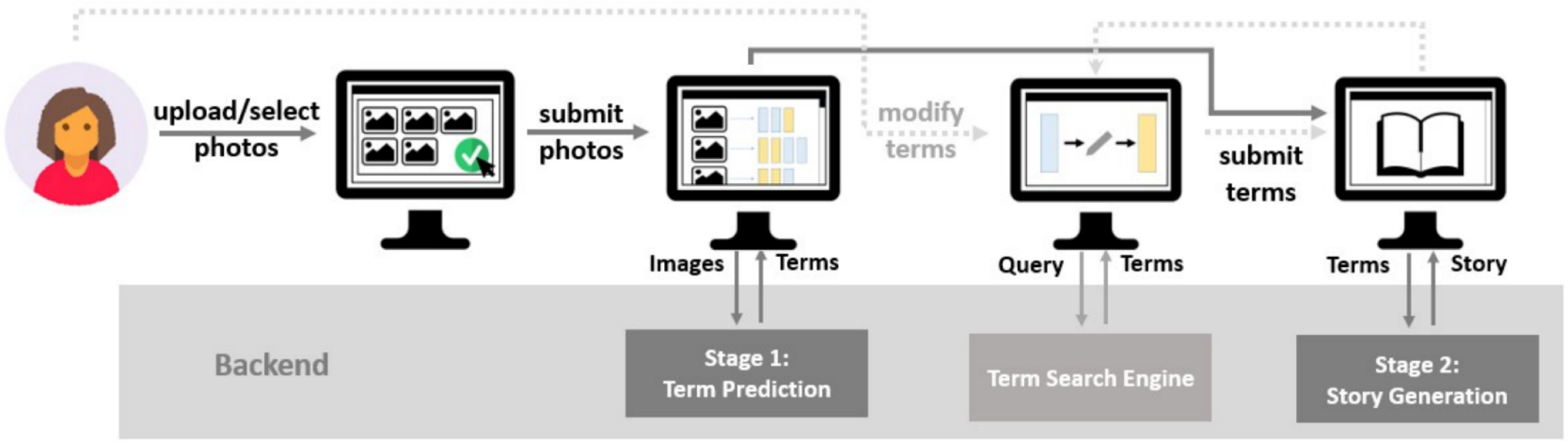


Figure 1: System architecture. The user initiates the process by uploading a sequence of photos. Dixit first extracts text *terms* from each photo which describe the objects or actions in the photo, and then allows the user to add new terms or remove existing terms. Dixit then generates a short story based on these terms.

Framework

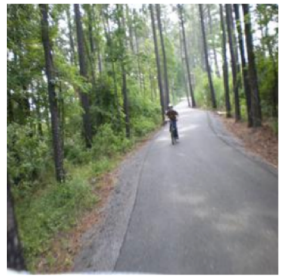
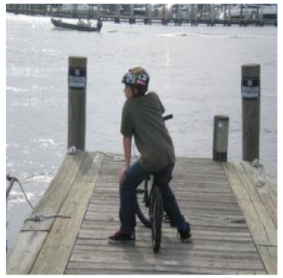
					
Predicted Terms	"man", "Placing", "bike"	"motorcycle", "rider", "Emptying", "bike"	"trees", "sign", "Preventing_or_letting",	"man", "Placing", "bench"	"boy", "seat", "Placing"
Story	the man was sitting on his bike.	he was riding his bike with a rider on his motorcycle.	he stopped at a stop sign from a trees.	the man sat on the bench.	the boy sat in his seat.
Modified Terms	"boy", "Placing", "bike"	"rider", "bike"	"Seeking", "trees", "forest"	"bike", "dock"	"Sleep", "seat"
New Story	the boy sat down at his bike.	he was riding his bike like a rider.	he went into the forest looking for trees.	he threw his bike at the dock.	he went back to his seat to sleep.

Table 1: Via term adjustment, story generation is focused on specific objects or actions

Model

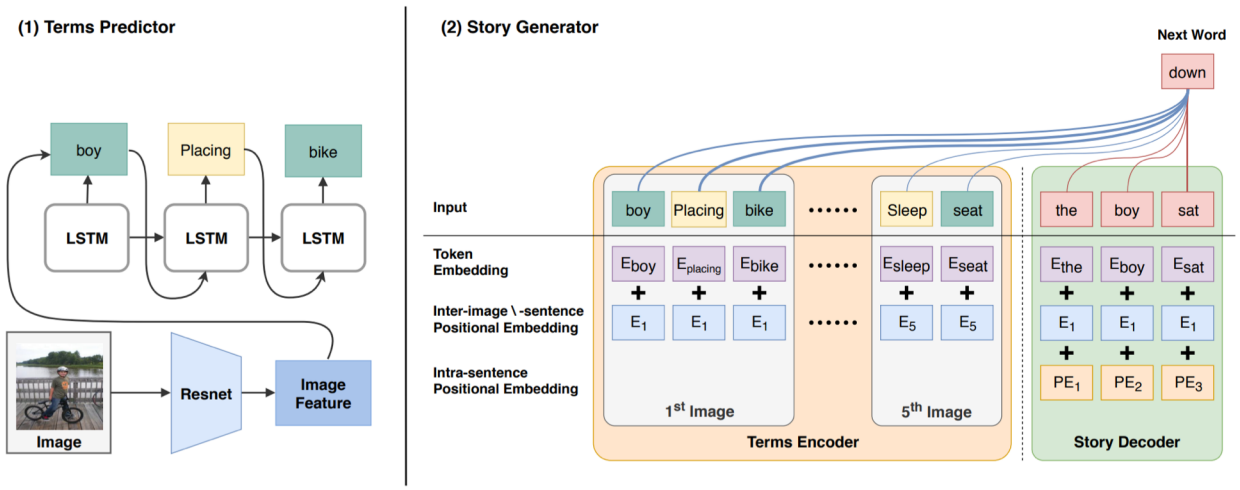


Figure 3: Architecture of (1) term prediction model and (2) story generation model. Predicted terms in green boxes denote noun terms, and terms in yellow boxes denote verb frames. In the training process, which uses the a single vocabulary, we treat both noun terms and verb frames equally. Term box colors are used here to emphasize their different origins.

- The InterPE is randomly initialized and updated during the training procedure. Similar to the segment embedding in BERT [2], we differentiate sentences/images by adding E_s to token embeddings of the s-th sentence/image; s denotes the order of a sentence/image in a story. $s \in \{1, 2, 3, 4, 5\}$.
- For IntraPE, we follow the Transformer implementation:

$$\text{IntraPE}_{(pos,\,2i)} = sin(pos/10000^{2i/d_{model}})$$

$$\text{IntraPE}_{(pos,\,2i+1)} = cos(pos/10000^{2i/d_{model}})$$

where pos is the position and i is the dimension. d_{model} denotes the dimension of the input and output token. The IntraPE parameters are fixed, and the sinusoidal representation allows the model to extrapolate the sentence length that is longer than the training instances. The IntraPE PE_s is added to the s-th token embedding in a sentence. $s \in \{1, 2, ..., n\}$ and n is the max sentence length.

Q & A

THANK YOU !