



# Chinese Poetry Generation with a Salient-Clue Mechanism

Xiaoyuan Yi<sup>1</sup>, Ruoyu Li<sup>2</sup>, Maosong Sun<sup>1</sup>

<sup>1</sup>Tsinghua University

<sup>2</sup>6ESTATES PTE LTD

*CoNLL 2018*

# Poetry

江雪 柳宗元

千山鸟飞绝，  
万径人踪灭。  
孤舟蓑笠翁，  
独钓寒江雪。

From hill to hill no bird in flight;  
From path to path no man in sight.  
A lonely fisherman afloat,  
Is fishing snow in lonely boat.

(1) Concise language

(2) Exquisite expression

(3) Rich content

(4) Structural and phonological  
requirements



# Automatic Poetry Generation



A desirable **entry point** for  
automatic analyzing,  
understanding and utilizing  
literary text.



# Previous Models

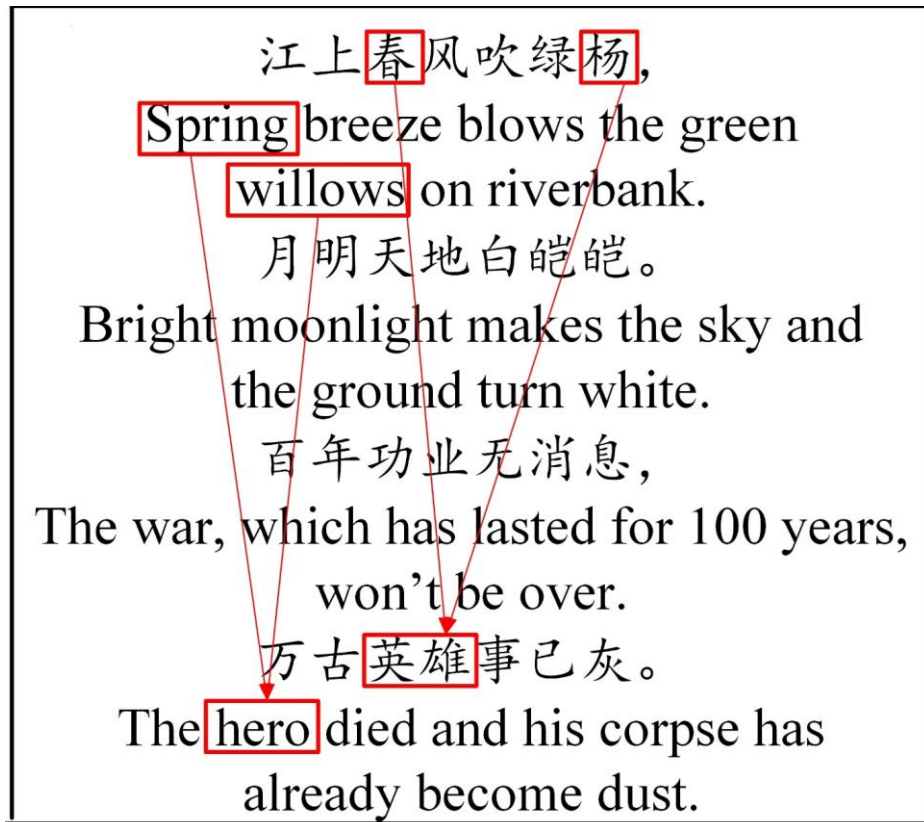
- **Innovation** (Zhang et al., 2017)
- **Rhythmic Constraints** (Ghazvininejad et al., 2016)
- **Keywords Insertion** (Wang et al., 2016)

...

## Context Coherence ?



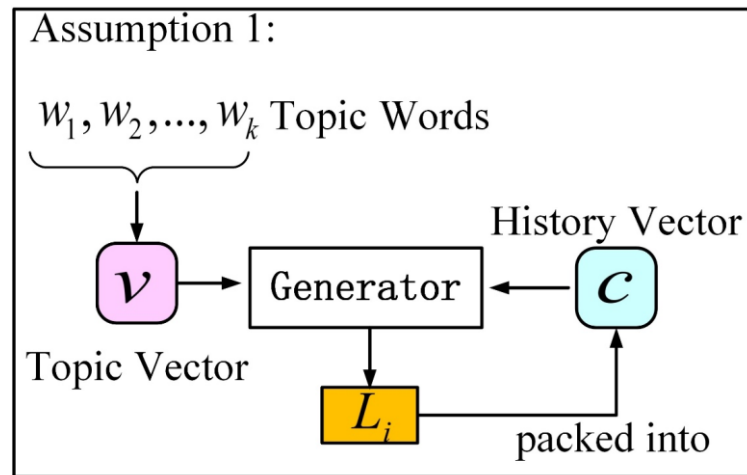
# Previous Models



A poem generated by (Zhang and Lapara, 2014). The input keyword is *chun feng* (spring breeze). Red boxes and arrows show the inconsistency.

# Previous Models

(1) *Single* history/topic vector (Zhang and Lapata 2014; Yan, 2016)



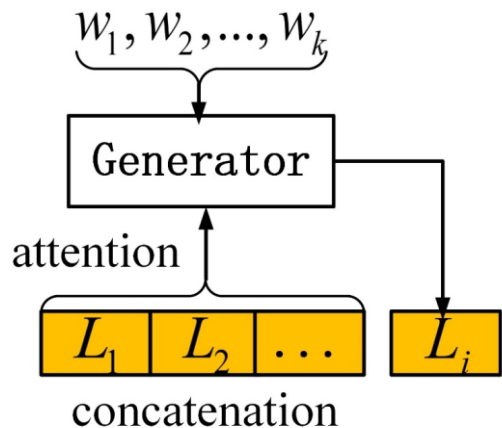
A graphical illustration of assumption 1.

- Insufficient capacity for maintaining the **full** history.
- Informative words and noises are **mixed** (e.g., stop words).
- Indiscriminate and inefficient use of the context.
- Low topic expression ratio (different topic words are mixed).

# Previous Models

(2) The ability of exploring *unlimited* history.  
(Wang et al., 2016; Zhang et al., 2017)

Assumption 2:



A graphical illustration of assumption 2.

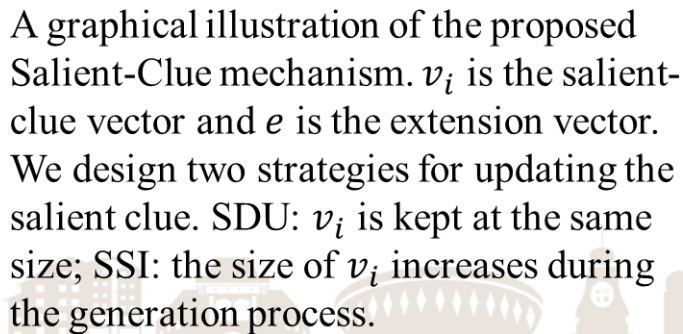
Keyword	The Preceding Text	Current Line
床 霜 明月 故乡	— 床前明月光 床前明月光; 疑是地上霜 <b>床前明月光; 疑是地上霜; 举头望明月</b>	床前明月光 疑是地上霜 举头望明月 低头思故乡

Training triples extracted from a quatrain in (Wang et al., 2016)

Too long input/output sequences

The performance of seq2seq model degrades, even with an attention mechanism.

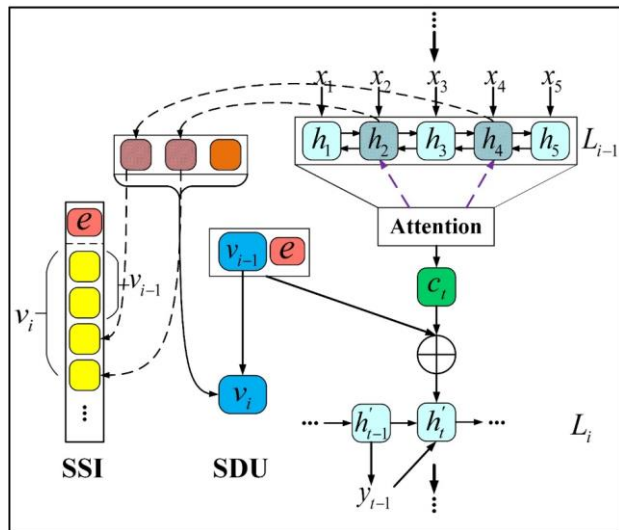
Design Philosophy: ignore the uninformative parts (e.g., stop words) and use **some salient** characters in context to represent the full context and form a **salient clue**, which is used to guide the generation process.



1. For each generated line: selects salient and informative characters to form the salient-clue.
2. When generating each line: utilizes the salient-clue.



# Salient-Clue Mechanism



A graphical illustration of the proposed Salient-Clue mechanism.  $v_i$  is the salient-clue vector and  $e$  is the extension vector. We design two strategies for updating the salient clue. SDU:  $v_i$  is kept at the same size; SSI: the size of  $v_i$  increases during the generation process.

Basic framework: Bidirectional LSTM  
Encoder-Decoder with attention mechanism  
(Bahdanau et al., 2015)

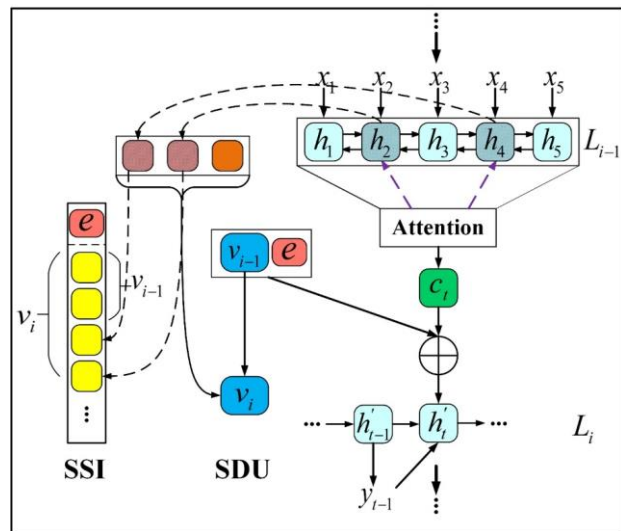
$$h'_t = LSTM(h'_{t-1}, emb(y_{t-1}), c_t), \quad (1)$$

$$p(y_t | y_{1:t-1}, L_{1:i-1}) = g(h'_t, emb(y_{t-1}), c_t, v), \quad (2)$$

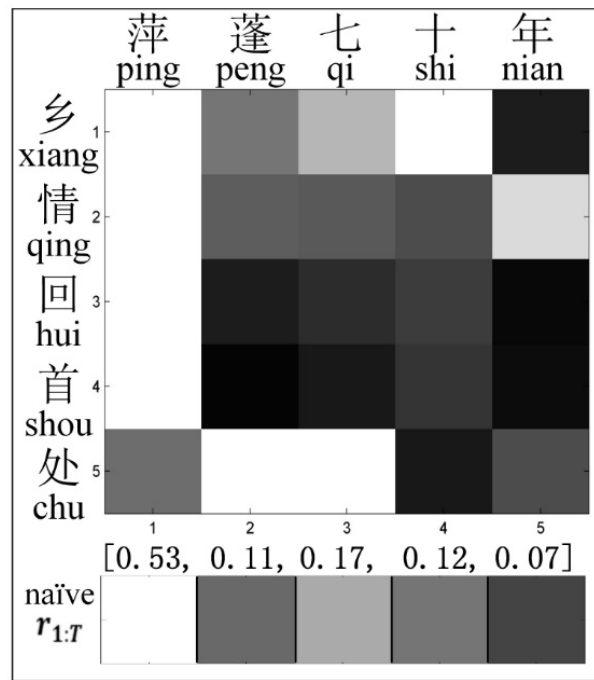
$$r_j = [(\mathbf{w}_{out} * \textcircled{A}) \odot \mathbf{w}_{in}]_j, \quad (3)$$

the attention alignment matrix in the attention mechanism

# Salient-Clue Mechanism

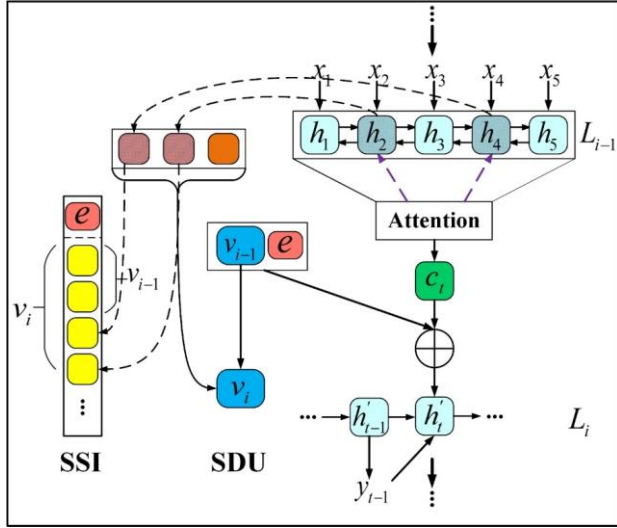


A graphical illustration of the proposed Salient-Clue mechanism.  $v_i$  is the salient-clue vector and  $e$  is the extension vector. We design two strategies for updating the salient clue. SDU:  $v_i$  is kept at the same size; SSI: the size of  $v_i$  increases during the generation process.



An example of calculating the saliency score of each character (in the x-axis) from the attention matrix (0:black, 1:white), in the naive Salient-Clue. The scores are normalized to interval [0,1] here.

# Salient-Clue Mechanism



A graphical illustration of the proposed Salient-Clue mechanism.  $v_i$  is the salient-clue vector and  $e$  is the extension vector. We design two strategies for updating the salient clue. SDU:  $v_i$  is kept at the same size; SSI: the size of  $v_i$  increases during the generation process.

---

## Algorithm 1 Saliency Selection Algorithm

---

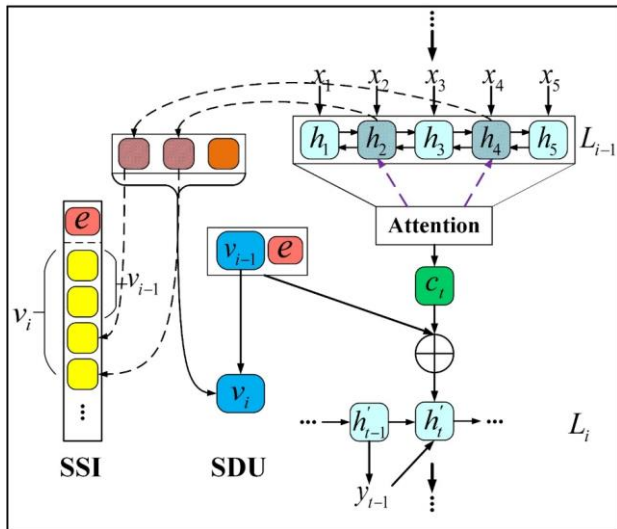
**Inputs:** The saliency scores of characters in the preceding line,  $r_{1:T}$ ;  $K$ ;

**Outputs:** The number of finally selected salient characters,  $N$ ; The indices of selected characters in the preceding line,  $m_{1:N}$ ;

- 1: Calculate the mean value of  $r_{1:T}$ ,  $avg$ ;
  - 2: Calculate the standard deviation of  $r_{1:T}$ ,  $std$ ;
  - 3: Get sorted indices  $i_{1:T}$  in descending order of  $r_{1:T}$ ;
  - 4:  $k = 1$ ;  $val = avg + 0.5 * std$ ;
  - 5: **while** ( $r_{i_k} \geq val$ ) and ( $k \leq K$ ) **do**
  - 6:    $m_k = i_k$ ;  $val = val * 0.618$  (the golden ratio);  $k = k + 1$ ;
  - 7: **end while**
  - 8:  $N = k - 1$ ;
  - 9: **return**  $N, m_{1:N}$ ;
- 

$$N, m_{1:N} = SSsal(r_{1:T}, K), \quad (4)$$

# Salient-Clue Mechanism



A graphical illustration of the proposed Salient-Clue mechanism.  $v_i$  is the salient-clue vector and  $e$  is the extension vector. We design two strategies for updating the salient clue. SDU:  $v_i$  is kept at the same size; SSI: the size of  $v_i$  increases during the generation process.

## Saliency Dynamic Update (SDU) V.S. PFC

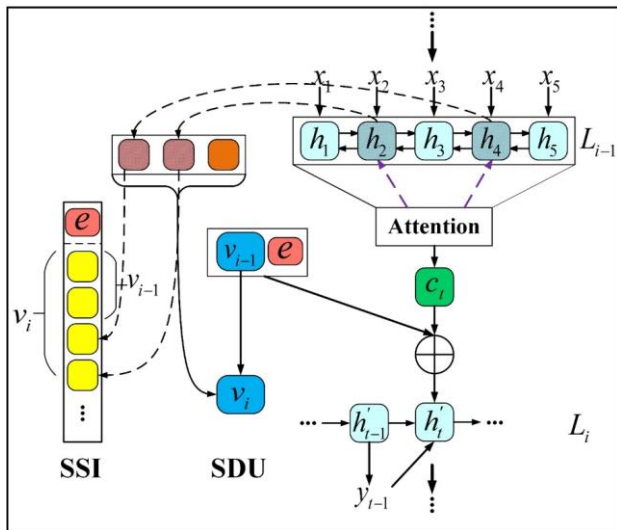
$$s = \frac{\sum_{k=1}^N r_{m_k} * h_{m_k}}{\sum_{k'=1}^N r_{m_{k'}}}, \quad (5)$$

$$v_i = \sigma(v_{i-1}, s), v_0 = \vec{0}, \quad (6)$$

## Saliency Sensitive Identity (SSI) V.S nLto1L

$$v_i = [v_{i-1}; h_{m_1}; \dots; h_{m_N}], \quad (7)$$

# Salient-Clue Mechanism



A graphical illustration of the proposed Salient-Clue mechanism.  $v_i$  is the salient-clue vector and  $e$  is the extension vector. We design two strategies for updating the salient clue. SDU:  $v_i$  is kept at the same size; SSI: the size of  $v_i$  increases during the generation process.

## Extensions of Salient-Clue

$$p(y_t | y_{1:t-1}, L_{1:i-1}) = g(h'_t, \text{emb}(y_{t-1}), c_t, [v_{i-1}; e]). \quad (8)$$

- Intent Salient-Clue**

Feed the keyword into Encoder, then vector  $e$  is calculated by a non-linear transformation of the average of their hidden states.

- Style Salient-Clue**

Simply use a style embedding as the vector  $e$ . Use LDA to train the whole corpus. For three main styles, Pastoral, Battlefield and Romantic, find the corresponding topics manually. Then all poems are labeled by LDA inference.

# Experimental Results

	Models	Wujue	Qijue
Different Models	Planning	0.460	0.554
	iPoet	0.502	0.591
	seq2seqPG	0.466	0.620
	SC	<b>0.532</b>	<b>0.669</b>
Different Strategies of SC	naive-TopK-SDU	0.442	0.608
	naive-SSal-SDU	0.471	0.610
	tfidf-SSal-SDU	<b>0.533</b>	0.648
	tfidf-SSal-SSI	0.530	0.667
	tfidf-SSal-SSI-intent	0.532	<b>0.669</b>

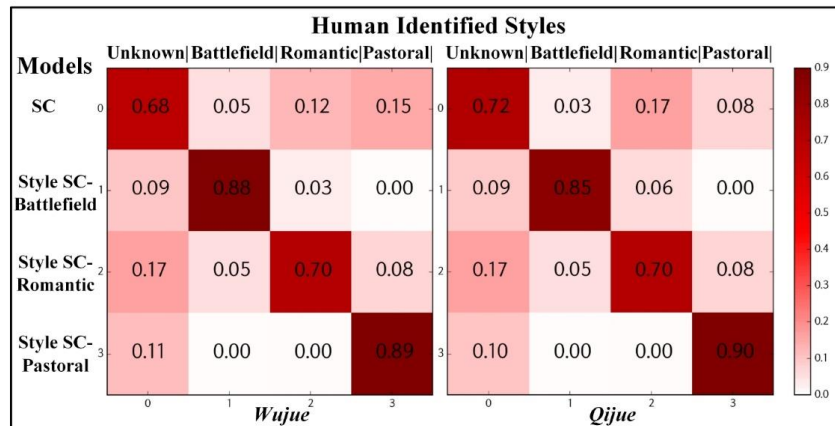
Table 1: BLEU evaluation results. The scores are calculated by the multi-bleu.perl script.

Models	Fluency		Coherence		Meaningfulness		Poeticness		Entirety	
	Wujue	Qijue	Wujue	Qijue	Wujue	Qijue	Wujue	Qijue	Wujue	Qijue
Planning	2.56	2.84	2.50	2.64	2.49	2.64	2.59	2.88	2.39	2.66
iPoet	3.13	3.45	2.89	2.91	2.60	2.80	2.79	3.05	2.54	2.85
seq2seqPG	3.54	3.65	3.31	3.16	3.15	3.01	3.26	3.29	3.06	3.08
SC	4.01**	4.04**	3.85**	3.86**	3.55**	3.63**	<b>3.74**</b>	<b>3.69*</b>	3.63**	<b>3.70**</b>
Style-SC	<b>4.03**</b>	<b>4.16**</b>	<b>3.90**</b>	<b>4.01**</b>	<b>3.68**</b>	<b>3.75**</b>	3.61*	3.68*	<b>3.65**</b>	<b>3.70**</b>
Human	4.09	4.43	3.90	4.33 <sup>+</sup>	3.94	4.35 <sup>++</sup>	3.83	4.24 <sup>++</sup>	3.81	4.24 <sup>++</sup>

Table 2: Human evaluation results. Diacritics \* ( $p < 0.05$ ) and \*\* ( $p < 0.01$ ) indicates SC models significantly outperform the three baselines; + ( $p < 0.05$ ) and ++ ( $p < 0.01$ ) indicates Human is significantly better than all the five models. The Intraclass Correlation Coefficient of the four groups of scores is 0.596, which indicates an acceptable inter-annotator agreement.



# Experimental Results



Style control evaluation results. The values are ratios that generated poems are identified as different styles by human experts.

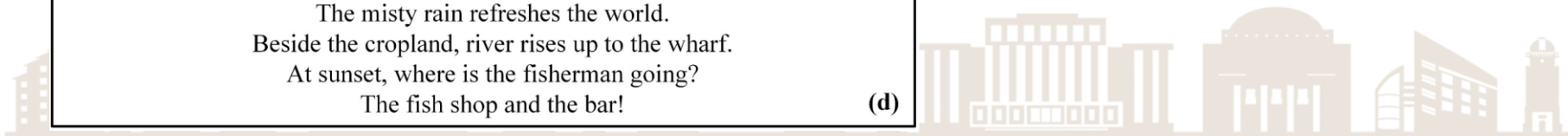
Models	<i>Wujue</i>	<i>Qijue</i>
Random	0.271	0.247
tf-idf	0.417	0.378
naive-TopK SC	0.347	0.415
naive-SSal SC	0.431	0.441
tfidf-SSal SC	<b>0.525</b>	<b>0.461</b>

Table 3: Saliency selection results. Random: randomly select K characters for three times and use the average Jaccard values. tf-idf: directly select K characters in terms of tf-idf, without SC.

# Case study

<p>濛濛烟雨暗江天，一片青山水自妍。为问故人何处所，夕阳依旧绿杨边。</p> <p>Misty rain darkens the sky. The green mountains and the rivers are so beautiful. Where is my old friend? I know, he is standing beside green willows at sunset.</p>	(a)
<p>茫茫烟雨不胜愁，千古英雄泪眼流。一代兴亡今已矣，百年遗恨付沧洲。</p> <p>There is endless sadness in the misty rain. Tears at last welled from the hero's eyes. The war has ended now, I leave for my hometown with a century-old hate.</p>	(b)
<p>濛濛烟雨湿衣裳，满地春风十里香。一曲琵琶无处觅，隔帘明月是何郎。</p> <p>The misty rain wets my dress. Spring wind brings the fragrance of flowers. Playing pi-pa (a kind of Chinese musical instrument), the boy out of the curtain, I want to know who you are.</p>	(c)
<p>濛濛烟雨一番新，野水平畴绿涨津。何处渔郎归去晚，数家沽酒卖鱼人。</p> <p>The misty rain refreshes the world. Beside the cropland, river rises up to the wharf. At sunset, where is the fisherman going? The fish shop and the bar!</p>	(d)

Four *Qijues* generated with the same keyword “烟雨” (misty rain) as input. (a) non-style by SC. (b) Battlefield style by Style-SC. (c) Romantic style by Style-SC. (d) Pastoral style by Style-SC.





# Case Study

## seq2seqPG

一夜扬州月, The moon in Yangzhou city makes me depressed,  
凄凉万里心。 since I am far away from it.  
故乡无限意, The missing for my hometown is endless.  
惆怅暮云阴。 It seem the cloud is also sad at sunset.

## SC

忆昔扬州月, I recall the past moon in Yangzhou city.  
于今又一秋。 Now, another autumn has come.  
故人何处是, Where can I find my old friends?  
落叶满汀洲。 Maybe on the shoal covered with fallen leaves.

于 今 又 一 秋  
salience at now again one autumn  
scores: [0.084, 0.120, 0.121, 0.058, 0.616]

Two *Wujues* generated with the same input. Green boxes and arrows show consistencies, and the red ones show inconsistencies.

Automatically selected charcters by SC are underlined.

# Weakness

## 1. Inflexible selection method

SC

细雨凉风吹客袂，  
Breeze is blowing and drizzle wets the sleeves of a traveler.

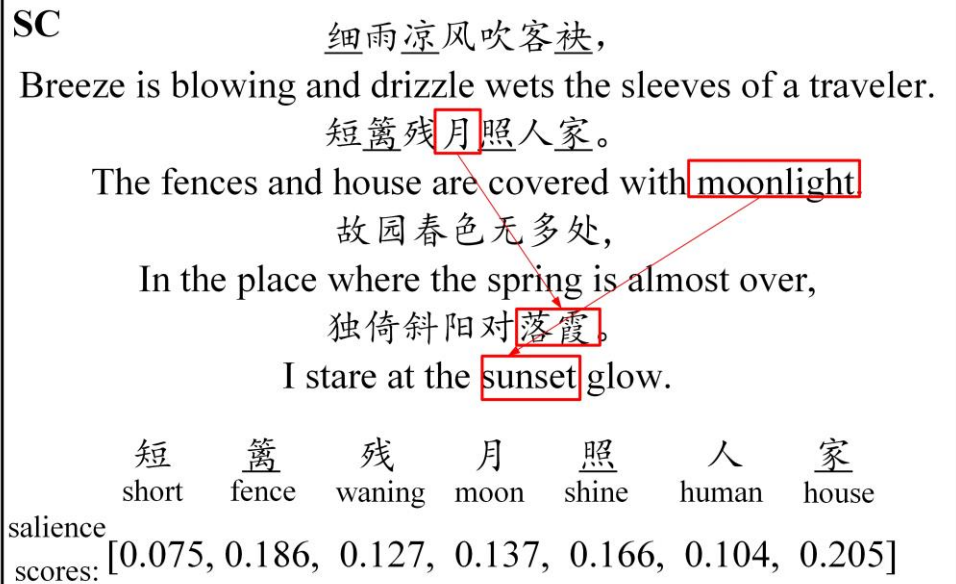
短篱残月照人家。  
The fences and house are covered with moonlight

故园春色无多处，  
In the place where the spring is almost over,

独倚斜阳对落霞。  
I stare at the sunset glow.

短 篱 残 月 照 人 家  
short fence waning moon shine human house

salience  
scores: [0.075, 0.186, 0.127, 0.137, 0.166, 0.104, 0.205]



A negative example. A *Qijue* generated by our SC model. Red box and arrow show the inconsistency.

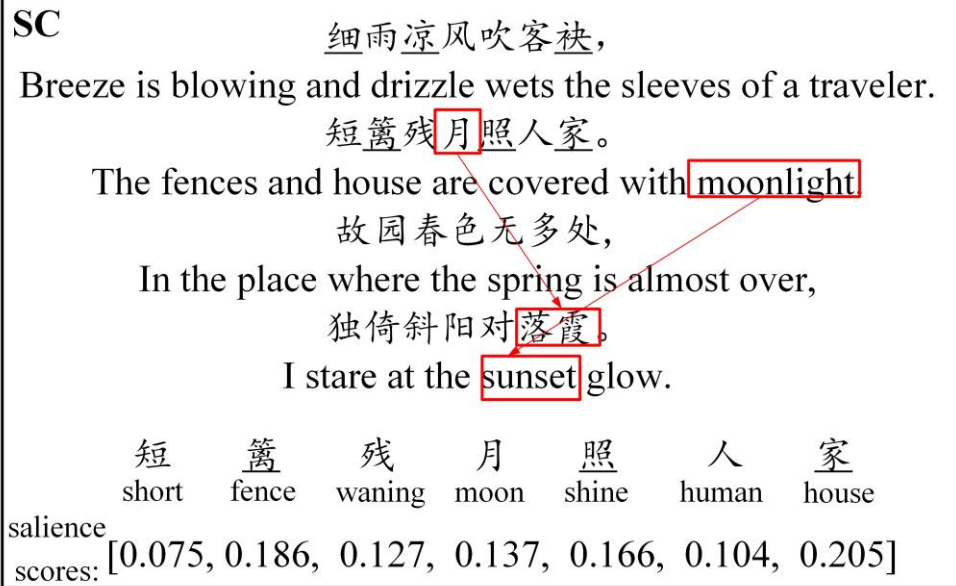
# Weakness

## 1. Inflexible selection method

SC                      细雨凉风吹客袂，  
Breeze is blowing and drizzle wets the sleeves of a traveler.  
                            短篱残月照人家。  
The fences and house are covered with moonlight  
                            故园春色无多处，  
In the place where the spring is almost over,  
                            独倚斜阳对落霞。  
I stare at the sunset glow.

短    篱    残    月    照    人    家  
short   fence   waning   moon   shine   human   house

salience  
scores: [0.075, 0.186, 0.127, 0.137, 0.166, 0.104, 0.205]



A negative example. A *Qijue* generated by our SC model. Red box and arrow show the inconsistency.

## 2. Supervised learning for style transfer

- Expensive labeled data
- Losing some fluency and Poeticness

- A novel model for poetry generation – Salient-Clue Mechanism
  - Inspired by human writing manner
  - With salient characters to form a vital clue
  - Better coherence
- Flexible structure to combine extra information
  - Combining user intent and style control
  - Further enhancing coherence





Jiuge (九歌), a Chinese poetry generation system developed by THUNLP&CSS lab.

- Support most popular genres of Chinese poetry
- Online generation interface
- Page View > 1 million

**The proposed model has been integrated into Jiuge!**

<http://jiuge.thunlp.org/>



# Thanks!

Oral Presentation by Xiaoyuan Yi(16:30, 3 Nov, Copper Hall):  
Automatic Poetry Generation with Mutual Reinforcement  
Learning

Poster Presentation by Cheng Yang(09:00, 4 Nov, Grand Hall 2):  
Stylistic Chinese Poetry Generation via Unsupervised Style  
Disentanglement



<http://jiuge.thunlp.org/>