Prophet Attention: Predicting Attention with Future Attention

Fenglin Liu[1], Xuancheng Ren[2], Xian Wu[3], Shen Ge[3], Wei Fan[3], Yuexian Zou[1*], Xu Sun[2]
[1]ADSPLAB, School of ECE, Peking University
[2]MOE Key Laboratory of Computational Linguistics, School of EECS, Peking University
[3]Tencent [*]Also with Peng Cheng Laboratory, Shenzhen, China

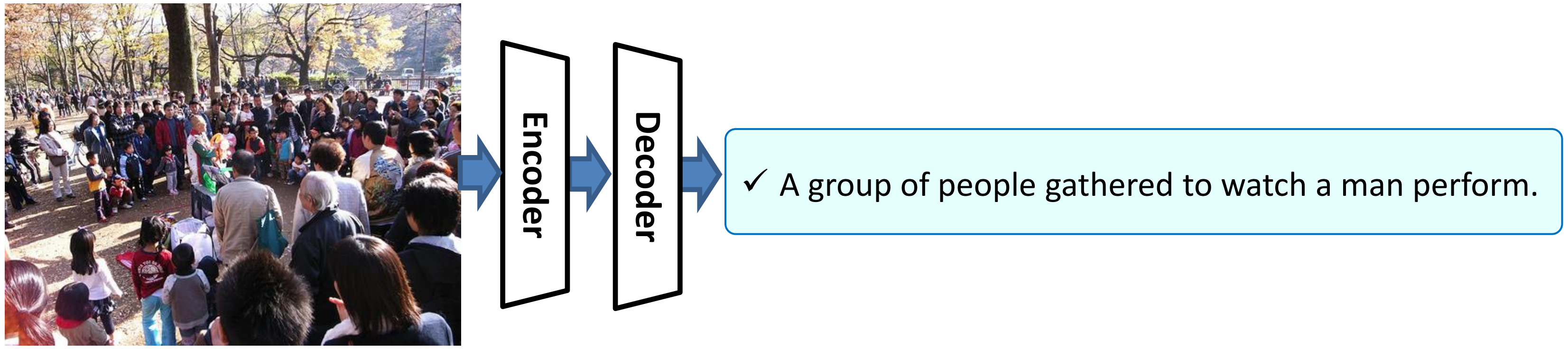
PEKING UNIVERSITY

目录

1. Introduction

Image Captioning

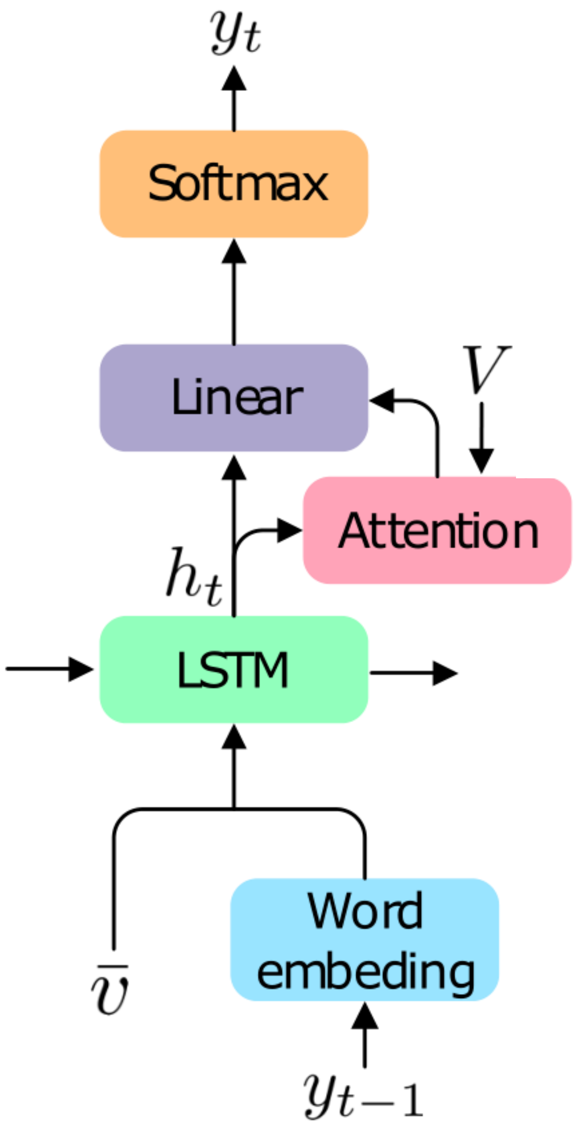
- Dataset: (V,S), where V and $S=\{s_1, s_2, \ldots, s_T\}$ represent the input image and the target sentence, respectively.
- Training Objective: The training objective is to minimize the cross entropy loss.



$$\text{Visual Enc.}: \mathcal{V} \rightarrow \hat{\mathcal{V}}; \quad \text{Target Dec.}: \hat{\mathcal{V}} \rightarrow \mathcal{S}.$$

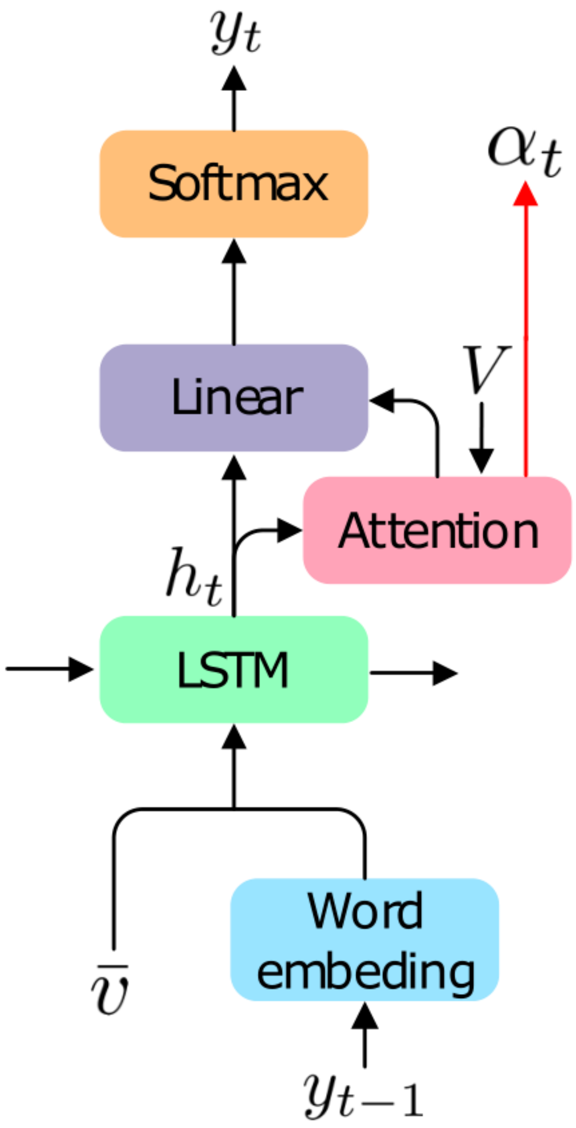
$$L_{CE}(\theta) = -\sum_{t=1}^{T} \log\left(p_\theta\left(s_t^* \mid s_{1:t-1}^*; \hat{\mathcal{V}}\right)\right)$$

Attention-Enhanced Encoder-Decoder Framework



- Visual Encoder: $V = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N\} \in \mathbb{R}^{d \times N}$
- Decoder-Input: $h_t = \text{LSTM}\left(h_{t-1}, [W_e y_{t-1}; \bar{v}]\right)$, $\bar{v} = \frac{1}{k}\sum_{i=1}^{k} v_i$
- Decoder-Attention: $\alpha_t = f_{\text{Att}}(h_t, V) = \text{softmax}\left(w_\alpha \tanh\left(W_h h_t \oplus W_V V\right)\right)$
  $c_t = V\alpha_t^{\mathrm{T}}$
- Decoder-Output: $y_t \sim p_t = \text{softmax}\left(W_p[h_t; c_t] + b_p\right)$
- Cross Entropy Loss: $L_{CE}(\theta) = -\sum_{t=1}^{T} \log\left(p_\theta\left(s_t^* | s_{1:t-1}^*; V\right)\right)$

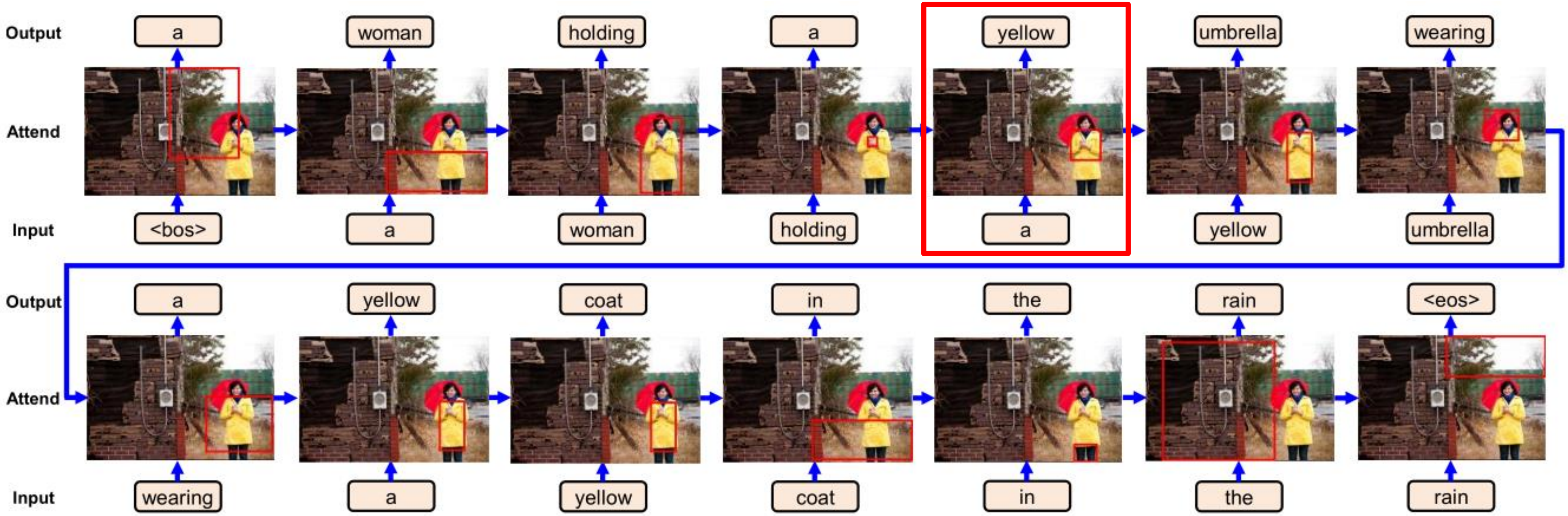
Motivations



$$\alpha_t = f_{\text{Att}}(\boxed{h_t}, V) = \text{softmax}\left(w_\alpha \tanh\left(W_h \boxed{h_t} \oplus W_V V\right)\right)$$

- Many sequence-to-sequence learning systems, including machine translation, have proven the importance of the attention mechanism in generating meaningful sentences. Especially for image captioning, the attention model can ground the salient image regions to generate the next word in the sentence.
- Current attention model attends to image regions based on current hidden state, which contains the information of past generated words. It means that the attention model has to predict attention weights without knowing the word it should ground.
- Thus we find that current attention models have a "**deviated focus**" problem, that they calculate the attention weights based on previous words instead of the one to be generated, impairing the performance of both grounding and captioning.

Examples

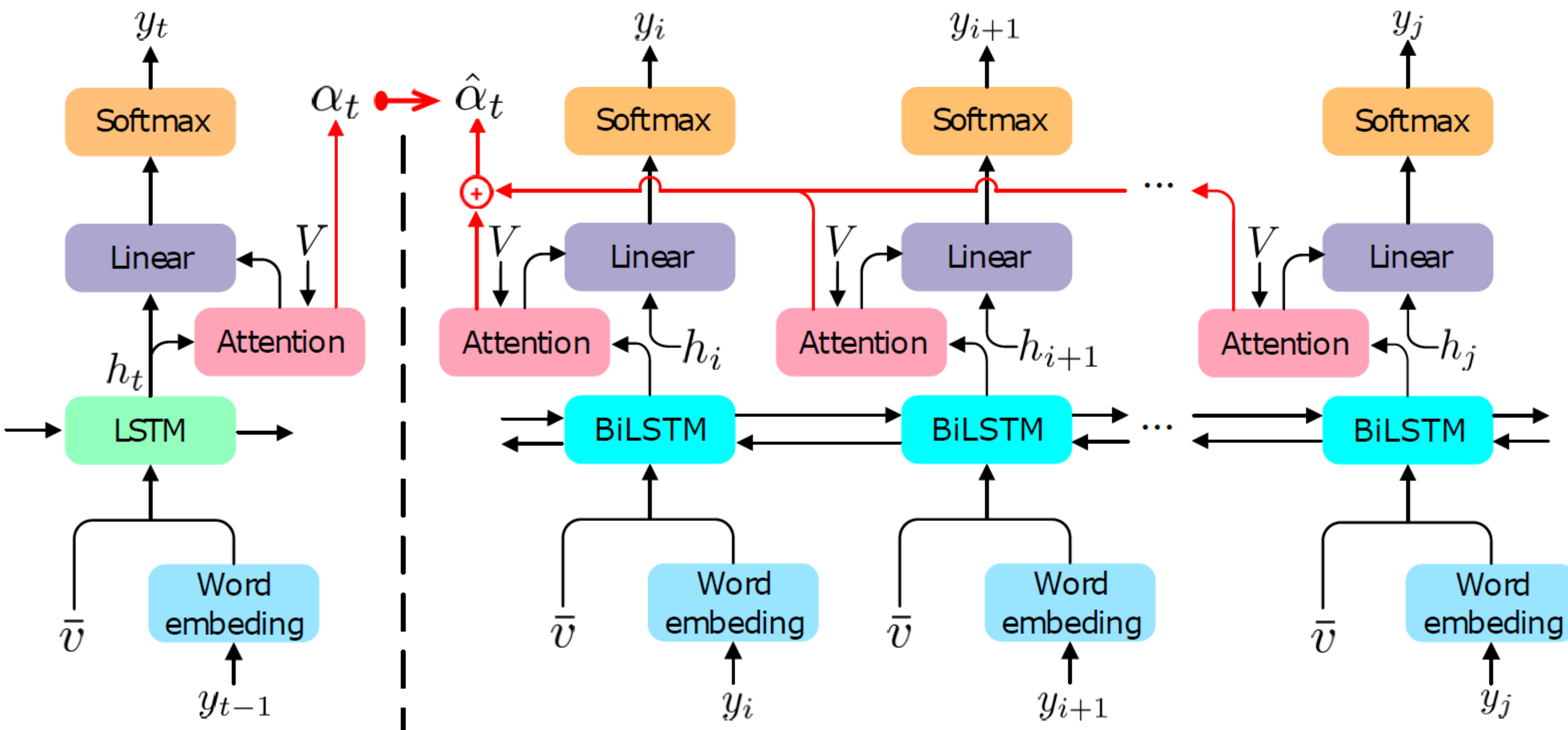


- At the time step to generate the 5^{th} word *yellow*, the attended image region is the *woman* instead of the *umbrella*. As a result, the incorrect adjective yellow is generated rather than the correct adjective *red*. This is mainly due to the "**focus**" of the attention is "**deviated**" several steps backwards and the conditioned words are *woman* and *holding*;

2. Prophet Attention

Approach

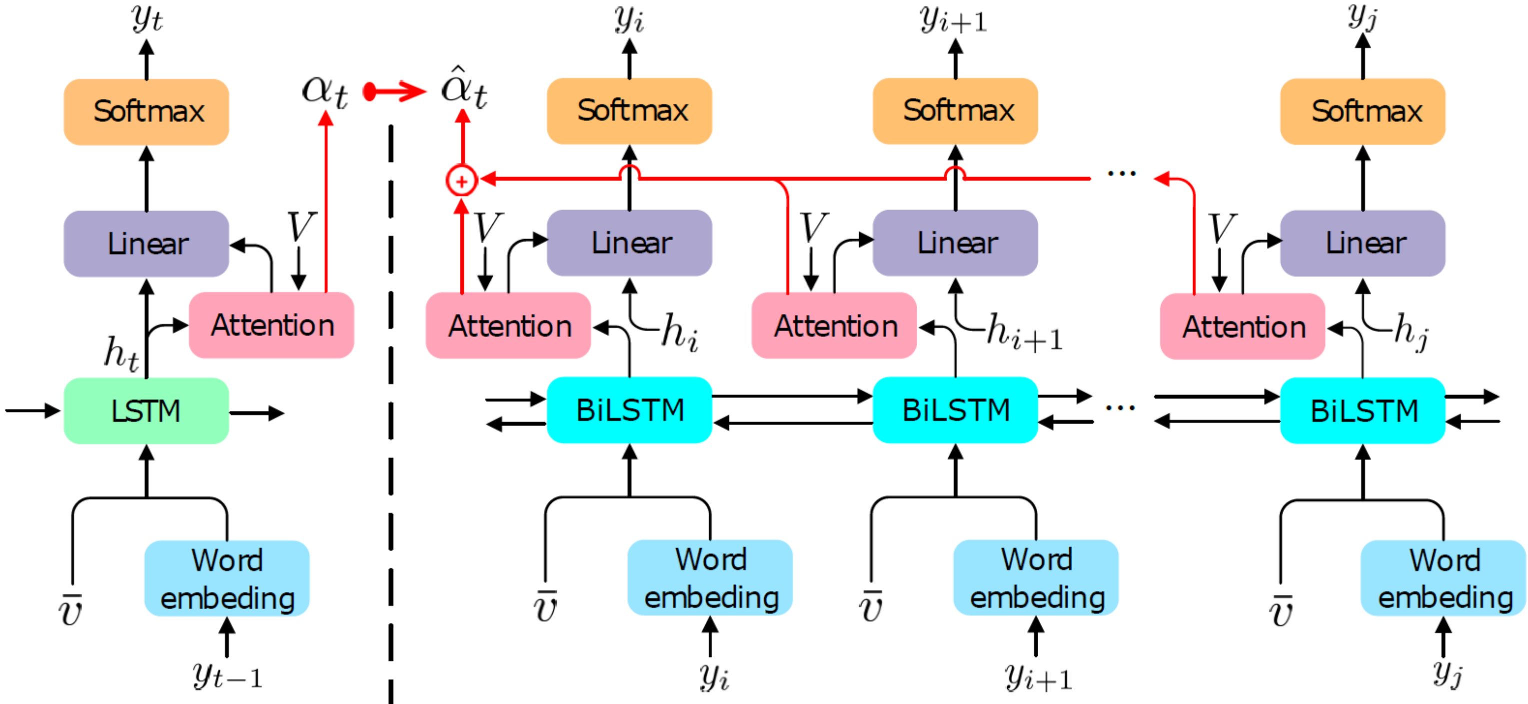
- In the training stage, our method utilizes the words that will be generated in the future to calculate the "**ideal**" attention weights towards image regions. These "**ideal**" attention weights are further used to **regularize** the "**deviated**" attention, which based on the input words that have already been generated. In this manner, image regions are grounded with the correct words.



(a) Visual Attention (b) Prophet Attention

Formulation

- "**Deviated**" weights: $\alpha_t = f_{\text{Att}}(h_t, V) = \text{softmax}\left(w_\alpha \tanh\left(W_h h_t \oplus W_V V\right)\right)$
- "**Ideal**" weights: $y_{i:j}\ (j \geq t) \longrightarrow h'_{i:j}$, $\hat{\alpha}_t = f_{\text{Prophet}}(h'_{i:j}, V) = \frac{1}{j-i+1}\sum_{k=i}^{j} f_{\text{Att}}(h'_k, V)$
- **Regularization**: $\mathcal{L}_{\text{Att}}(\theta) = \sum_{t=1}^{T} \|\alpha_t - \hat{\alpha}_t\|_1$



(a) Visual Attention (b) Prophet Attention

Constant Prophet Attention (CPA)

- Since the attention weight is mainly determined by the single word that is to be generated at the current time step, the intuition is to set i = j = t. In this manner, the CPA only uses the word y_t to be generated to calculate the **ideal** attention weights:

$$\hat{\alpha}_t = f_{\text{Prophet}}(h'_{i:j}, V) = f_{\text{Att}}(h'_t, V)$$

- **Confusing Attended Image Regions:** "*black* " for the "*a black shirt*" and "*black pants*".
- **Non-Visual Word:** e.g., *of* and *the*, -> no suitable visual information at all -> remove (mask) the Prophet Attention -> prevent it from affecting the learning of the captioning model.

Dynamic Prophet Attention (DPA)

- **Confusing Attended Image Regions:** *a black shirt* -> a whole phrase instead of individual words -> y_t belongs to a noun phrase (NP) -> adopt all the words in the noun phrase to calculate the **ideal** attention weights.
- **Non-Visual Word:** non-visual (NV) word -> remove (mask) Prophet Attention -> $\lambda = 0$.
- **Remaining:** Following the CPA -> i = j = t + 1 .

$$\hat{\alpha}_t = f_{\text{Prophet}}(h_{i:j}, V) = \begin{cases} \frac{1}{n-m}\sum_{k=m}^{n} f_{\text{Att}}(h_{k+1}, V) & \text{if } y_t \in \text{NP: } y_{m:n} \\ \text{MASK} & \text{if } y_t \in \text{NV: } \{y_{\text{NV}}\} \\ f_{\text{Att}}(h_{t+1}, V) & \text{otherwise} \end{cases}$$

- In all, through our Prophet Attention, the attention model can learn to ground each output word y_t to image regions without the ground-truth of grounding annotation.

3. Experiments

Online Evaluation

Table 2: Highest ranking published image captioning results on the online MSCOCO test server. c5 and c40 mean comparing to 5 references and 40 references, respectively. ‡ is defined similarly to Table 1. We outperform previously published work on major evaluation metrics. At the time of submission (2 June 2020), we also outperformed all unpublished test server submissions in terms of CIDEr-c40, which is the default ranking score, and rank the 1st.

Methods	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE-L		CIDEr	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Up-Down [2]	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
GLIED [27]	80.1	94.6	64.7	88.9	50.2	80.4	38.5	70.3	28.6	37.9	58.3	73.8	123.3	125.6
SGAE [54]	81.0	95.3	65.6	89.5	50.7	80.4	38.5	69.7	28.2	37.2	58.6	73.6	123.8	126.5
GCN-LSTM [55]	-	-	65.5	89.3	50.8	80.3	38.7	69.7	28.5	37.6	58.5	73.4	125.3	126.5
AoANet [20]	81.0	95.0	65.8	89.6	51.4	81.3	39.4	71.2	29.1	38.5	58.9	74.5	126.9	129.6
$\mathcal{M}^2$ Trans. [10]‡	81.6	96.0	66.4	90.8	51.8	82.7	39.7	72.8	29.4	39.0	59.2	74.8	129.3	132.1
X-Trans. [37]‡	**81.9**	95.7	**66.9**	90.5	**52.4**	82.5	**40.3**	72.4	**29.6**	39.2	**59.5**	75.0	**131.1**	133.5
Ours	81.8	**96.3**	66.5	**91.2**	51.9	**83.2**	39.8	**73.3**	**29.6**	**39.3**	59.4	**75.1**	130.4	**133.7**

4. Conclusions

Conclusions

- In this work, we focus on correctly grounding the image regions with generated words in the attention model.
- To this end, we propose the Prophet Attention, which is similar to the form of self-supervision for calculating attentional weights based on future information, and force the attention model to learn to correctly ground each output word to proper image regions.
- We evaluate Prophet Attention for image captioning on the Flickr30k Entities and MSCOCO datasets. We achieve the **1st** place on the leaderboard of the MSCOCO online server benchmark.

☺

Thank you for your attention!

The code is available at https://github.com/fenglinliu98/ProphetAttention
If you have any questions about our paper, you can send an email to fenglinliu98@pku.edu.cn