Non-Autoregressive Coarse-to-Fine Video Captioning
Technical Appendix

Bang Yang,[1] Yuexian Zou, [2*] Fenglin Liu, [1] Can Zhang [1]
[1] ADSPLAB, School of ECE, Peking University, Shen Zhen, China
[2] Peng Cheng Laboratory
{yb.ece, zouyx, fenglinliu98, zhangcan}@pku.edu.cn

Detailed Formulation

Decoder

Our decoder consists of a multi-head self-attention, a multi-head inter-attention and a feed-forward network. Assuming that attention layer has H heads, each of which can be defined as follows (omitting biases for clarity):

$$f_{att}^{h}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}W_Q(\mathbf{K}W_k)^T}{\sqrt{d_k}}\right)\mathbf{V}W_V \quad (1)$$

where $\mathbf{Q} \in \mathbb{R}^{s_Q \times d_m}$ denotes query, $\mathbf{K} \in \mathbb{R}^{s_K \times d_m}$ denotes key, $\mathbf{V} \in \mathbb{R}^{s_V \times d_m}$ denotes value, and s_* ($* \in Q, K, V$) is the sequence length. $\{W_Q, W_K, W_V\} \in \mathbb{R}^{d_m \times d_k}$ are learnable weights and $d_k = d_m/H$. So multi-head attention (MHA) can be formulated as follows:

$$f_{MHA}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{LN}([f_{att}^1; f_{att}^2; \ldots; f_{att}^H]W_{MHA} + \mathbf{Q}) \quad (2)$$

where $[;]$ represents concatenation, LN denotes layer normalization and $W_{MHA} \in \mathbb{R}^{d_m \times d_m}$ is linear transformation. Following the MHA is a fully connected feed-forward network (FFN), which is defined as follows:

$$f_{FFN}(\mathbf{X}) = \text{LN}(\text{GELU}(\mathbf{X}W_{F_1})W_{F_2} + \mathbf{X}) \quad (3)$$

where GELU is short for Gaussian Error Linear Unit (Hendrycks and Gimpel 2016), $W_{F_1} \in \mathbb{R}^{d_m \times d_h}$ and $W_{F_2} \in \mathbb{R}^{d_h \times d_m}$ are the matrices for linear transformation. In practice, f_{MHA} and f_{FFN} in our decoder excluded layer normalization (LN) because we found the training procedure more stable by doing so. Given a input sequence Y of length N and video representation $\mathbf{R}$, the transformation of our decoder is achieved by the following formula:

$$f_{dec}(Y,\mathbf{R}) = f_{PJ}(f_{FFN}(f_{MHA}(f_{MHA}(E,E,E),\mathbf{R},\mathbf{R}))) \quad (4)$$

where E is the normalized input embeddings of Y ($e_n = \text{LN}(e_n^{tok} + e_n^{pos} + e^{src})$) and f_{PJ} maps the outputs into a distribution over the vocabulary:

$$f_{PJ}(\mathbf{X}) = \text{Softmax}(XW_{PJ}) \quad (5)$$

where $W_{PJ} \in \mathbb{R}^{d_m \times \mathcal{V}}$ and $\mathcal{V}$ is the vocabulary size.

*Corresponding author.

Length Beam and Teacher Rescoring

Following the common practice of noisy parallel decoding (Gu et al. 2018; Wang et al. 2019), we select top B length candidates with the highest probability from the predicted distribution L during inference, which is somewhat analogous to beam search in AR decoding. Then we decode the same example with different lengths $\{N_1, N_2, \ldots, N_B\}$ in parallel. Here we make $N_b \in [4, N_{max}]$ to ensure that a complete sentence can be generated. Finally, we select the sequence with the highest average log-probability as our hypothesis based on $C^{(T)}$:

$$\frac{1}{N_b}\sum_{n=1}^{N_b} \log c_n^{(T)} \quad (6)$$

Besides solely using $C^{(T)}$ to decide the best candidate, we also use the *teacher rescoring* technique (Gu et al. 2018; Wang et al. 2019; Mansimov, Wang, and Cho 2019) to evaluate the collected B captions. In this work, we simply utilize AR-B to give a mark Z to the generated $Y^{(T)}$:

$$z_n = p_\theta(y = y_n^{(T)} | y_{<n}^{(T)}, \mathbf{R}) \quad (7)$$

where z_n reveals the coherence between $y_n^{(T)}$ and previous generated words $y_{<n}^{(T)}$. Note that this rescoring operation is parallel because the whole sentence $Y^{(T)}$ is already known. Then we jointly consider $C^{(t)}$ and Z to decide the best caption candidate, so Eq. 6 is modified as follows:

$$\frac{1}{2N_b}\sum_{n=1}^{N_b}(\log c_n^{(T)} + \log z_n) \quad (8)$$

Reproducibility

To assess the reproducibility of our work, we here introduce details of our experiments besides providing our codes.

Vocabulary Processing

To extract vocabulary from the MSVD and MSR-VTT datasets, we first remove punctuation and convert all words to lower case. Then for MSVD we keep all words whereas we filter words that appear less than 3 times for MSR-VTT. After taking special tokens (*i.e.*, $[bos]$, $[eos]$, $[unk]$, $[mask]$, $[vis]$, $[pad]$) into account, the resulting vocabulary size of MSVD is 9,468, whereas that of MSR-VTT is 10,547.

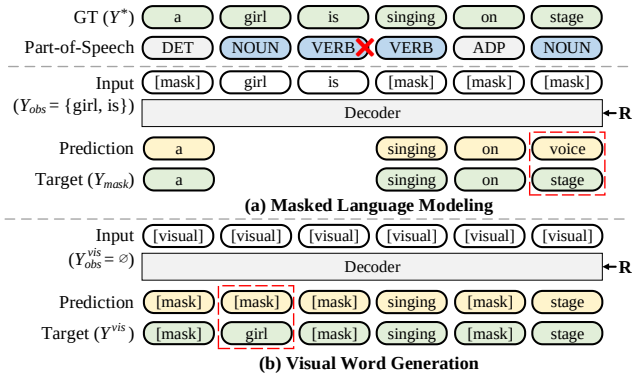


Figure 1: Illustration of training examples for (a) masked language modeling and (b) visual word generation. When generating visual words, we only focus on nouns and verbs, but ignore verbs like "is" and "are", which do not reveal the relevant visual contents.

Video Processing

To get sampled frames, we sample the video at 3 *fps* for MSVD whereas 5 *fps* for MSR-VTT, and set the maximum number of frames as 60. In terms of video clips, we sample the videos at 25 *fps* and extract features for every 16 consecutive frames with 8 frames overlap for both datasets. For features of each video, we divide them into $K = 8$ snippets and randomly/uniformly sample one frame per snippet in training/evaluation process.

Training

An intuitive example of training is given in Fig. 1. One may argue that masked language modeling (MLM) and visual word generation (VWG) belong to an inclusive relationship. Here are their differences. (1) MLM often processes partially-observed sentences while VWG always takes completed unobserved sequences as inputs. (2) MLM predicts any word in the vocabulary whereas VWG predicts either a [mask] token or a visual word (Fig. 1 (b)). Ideally, our NACF can be trained with two complementary settings, i.e., VWG and predicting unknown words conditioned on the visual words. Nevertheless, our method may fail to generate comprehensive visual words (see qualitative examples at the end of this document). Hence we adopt MLM and VWG for robustness. From another prospective, training MLM jointly with VWG is to some extent similar to attaching higher weights to those visual words when calculating the overall loss. Therefore, our proposed visual word generation can alleviate the insufficient training of meaningful words.

Evaluation Metrics

- **BLEU** (Papineni et al. 2002). BLEU@n measures the precision of n-grams between the ground-truth and generated sentences. This precision-based metric is generally believed to prefer short sentences, though it have introduced a penalty factor to alleviate that.
- **METEOR** (Banerjee and Lavie 2005) not only leverages a uni-grams-based weighted F-score and a penalty function to punish incorrect word order but also consider the matching of synonyms.
- **CIDEr-D** (Vedantam, Lawrence Zitnick, and Parikh 2015) adopts a voting-based approach, where TF-IDF is taken into account. Therefore, CIDEr-D prefers to punishing the often-seen words and awarding rare words. This metric is considered to be more robust to incorrect annotations and more relevant with human judgment.

Standard of Selecting the Best Model

Given the number of training epochs e, the score of a specific metric (*e.g.* BLEU@4) s_i at i-th epoch ($i \in [1, e]$) and the highest score of a specific metric s_i^{best} till i-th epoch, we calculate the relative score $rs = s_i / s_i^{best}$. For multiple metrics, we measure the summation of their relative scores. Finally, we select the model with the highest summation of relative scores on the validation set as our best model and report its performance on the testing set.

Experimental Details

Parameters. For our NACF, we set the number of iterations T to 5, the length beam size B to 6 and use the CT-MP algorithm unless otherwise specified. For NA-B, same T and B are adopted. But NAB does not support CT-MP algorithms because it excludes visual word generation. Hence NA-B use the MP algorithm by default. For autoregressive models (e.g., AR-B), beam size B is to 5, which has higher performance than $B = 6$ (see Fig. 5(b) in the submitted paper). All these parameters are determined by the performance on the validation set. Please note that B has different meanings for NA models and AR models.

Explanation of Fig. 4 in the submitted paper. Let denote the generated captions of NA-B on the MSR-VTT test set as C. From C, we can obtain a set of unique words W of size N, based on which we calculate the word frequency f_{w_i} for each $w_i \in W (i \in [1, N])$:

$$f_{w_i} = \frac{occurrence(w_i)}{\sum_{j=1}^{N} occurrence(w_j)} \tag{9}$$

Then the word frequencies $F = \{f_{w_1}, f_{w_2}, \ldots, f_{w_N}\}$ for NA-B is obtained. Similarly, we can get W_{vis} of size N_{vis} and F_{vis} for "NA-B w/ $\mathcal{L}_{vis}$". Given a word $w \in W_{vis} \cup W$, its relative growth rate of word frequency r_w is defined as:

$$r_w = \begin{cases} 1.0 & \text{if } w \in W_{vis} \ \& \ w \notin W \\ -1.0 & \text{if } w \notin W_{vis} \ \& \ w \in W \\ (f_w^{vis} - f_w)/f_w & \text{otherwise} \end{cases} \tag{10}$$

At last, given a specific type of words (*e.g.* visual words) W_{spe}, we measure $\frac{1}{|W_{spe}|} \sum_{w \in W_{spe}} r_w$, which is defined as **average relative growth rate of word frequency** in the submitted paper.

Extra Experiments and Analyses

Analysis on MSR-VTT Corpora

The division of part-of-speech (POS) tags is given in Table 1, where six subsets and their frequency on MSR-VTT are

Subsets	Part-of-Speech Tags	Corpus-Level Frequency	Vocab-Level Frequency	Per-Word Occurrence
Noun	NN, NNS	35.84	51.74	0.69
Verb	VB, VBD, VBG, VBN, VBP, VBZ	14.62	23.23	0.63
Adjective	JJ, JJR, JJS, AFX	12.67	19.23	0.66
Adverb	RB, RBR, RBS, WRB	6.08	1.06	5.74
Determiner	WDT, WP$, PRP$, DT, PDT	19.00	0.29	65.52
Others	IN, RP, CC, CD, . . . , "be" (is, are, was, . . .)	11.79	4.45	2.65

Table 1: The Part-of-Speech (POS) tags are obtained by the NLTK toolkit (Bird, Klein, and Loper 2009) and divided into six subsets, whose frequency (%) in corpus- and vocabulary-level on the whole MSR-VTT dataset is also presented.

Combination	Stage 1: Training			Stage 2: Training + Coarse-Grained Template		
	BLEU@4	METEOR	CIDEr-D	BLEU@4	METEOR	CIDEr-D
∅ (NA-B)	40.4	28.0	47.6	-	-	-
{Noun}	40.7	28.0	48.8	41.5	28.4	50.4
{Verb}	40.6	27.9	48.2	40.7	27.9	48.3
{Noun, Verb} (**NACF**)	**40.8**	28.2	49.4	**42.0**	28.7	**51.4**
{Noun, Adjective}	40.6	28.2	49.6	41.3	28.5	50.7
{Noun, Verb, Adjective}	**40.8**	**28.3**	**49.9**	**42.0**	**28.8**	51.3
{Noun, Verb, Adverb}	40.3	27.9	48.6	41.3	28.3	50.2
{Noun, Verb, Adjective, Adverb}	40.0	28.0	48.4	40.9	28.4	50.4

Table 2: Performance of our NACF (CT-MP) when using different combinations of POS subsets for visual word generation on MSR-VTT test set.

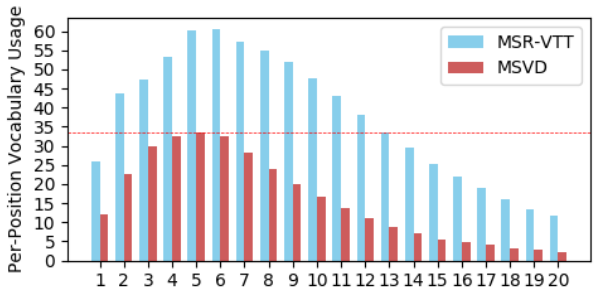


Figure 2: The statistics of training data about per-position vocabulary usage, which may reveal the model's perplexity on each position.

presented. By comparing the per-word occurrence of different subsets, we can observe that non-visual words (*e.g.* determiner) appear more frequently than visual words like nouns and verbs.

Why NA-B Behaves Poorly on MSR-VTT?

We suspect that NA-B confronts a more serious "multi-modality" problem (Gu et al. 2018) on MSR-VTT than MSVD. To test our hypothesis, we measure per-position vocabulary usage in training data, which may reveal the model's perplexity on each position, as a proxy metric for the "multi-modality" problem. The results shown in Fig. 2 match our hypothesis. For example, more than one-quarter words from the vocabulary can be placed at the first position (the start of the sentence) on MSR-VTT, more than twice the proportion on MSVD, making the NA-B more confused when predicting tokens individually. On the one hand, this suggests that linguistic diversity of a large-scale dataset present great challenges for NA decoding. On the other hand, the superior performance of our NACF demonstrates that decoding sentences under the guidance of some content related visual words can mitigate description ambiguity, i.e., alleviate the "multi-modality" problem.

Different Combinations of Part of Speech

Instead of only generating nouns and verbs (i.e., our proposal of visual word generation), we here analyze the effect of different combinations of POS subsets in Table 2 and have the following observations:

- Compared with "{Verb}", "{Noun}" performs better. On the one hand, nouns are more common than verbs in the corpus and vocabulary (see Table 1). On the other hand, nouns are highly associated with salient subjects or objects in video contents, which could be more distinguishable than verbs like "playing", "talking". So both enhancing the model's awareness towards nouns (stage 1) and taking generated nouns as the coarse-grained templates (stage 2) can yield better performance.
- The reason why "{Verb}" brings limited improvement can refer to Fig. 4 (b). As we can see, our model performs badly on identifying verbs, which is probably due to the lack of position modeling (Bao et al. 2019; Gu et al. 2018)

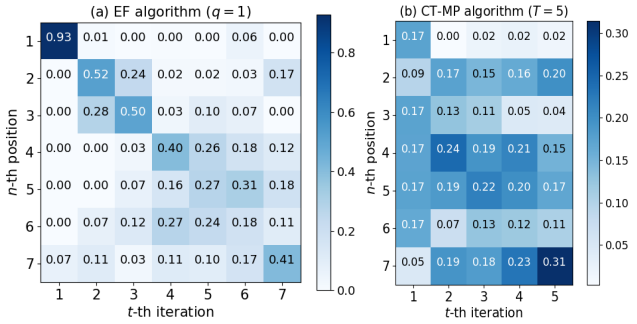


Figure 3: The possibility of generating new words at n-th position of the sentence at t-th iteration. Generated sentences of length 7 ($N = 7$) on MSR-VTT test set are studied here.

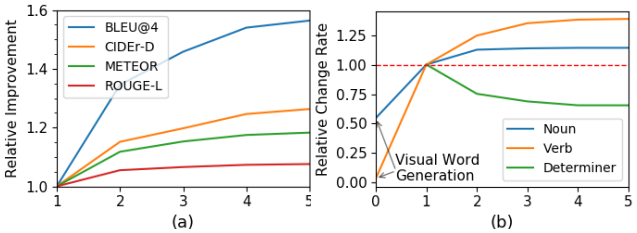


Figure 4: When using NACF (CT-MP), the relative improvement on different metrics (a) at t-th iteration and the relative change rate of part of speech (b) at t-th iteration ($T = 5$) on MSR-VTT test set.

that has been proved to be effective for non-autoregressive decoding.

- Among all combinations, both "{Noun, Verb, Adjective}" and "{Noun, Verb}" are promising. Although "{Noun, Verb, Adjective}" performs better than "{Noun, Verb}" at stage 1, they achieve similar performance at stage 2. This indicates that nouns and verbs in the generated coarse-grained templates already provide rich contextual information. Thus, we follow (Sigurdsson, Russakovsky, and Gupta 2017) and treat nouns and verbs as visual words in our paper.

More Insights of Decoding Algorithms

We here investigate the preference of different decoding algorithms during decoding. As shown in Fig. 3 (a), EF is sure about the leftmost word ($n = 1$) at the first iteration ($t = 1$) but the rest words are generated adaptively. Based on the superior performance of (CT-)EF over (CT-)L2R, we conclude that adaptive rather than monotonic generation is requisite for our NACF to generate plausible descriptions. We also visualize the refinement order of CT-MP in Fig. 3 (b) and have the following observations. (1) Given sentences of length 7, visual words are often generated at second and rightmost positions. (2) Guided by the visual words, the leftmost, third and penultimate words are less likely to be reconsidered in later iterations ($t \geq 2$), which are usually determiners, verb "be" and prepositions (i.e., non-visual words). (3) The model, in turn, refines those visual words repeatedly in later

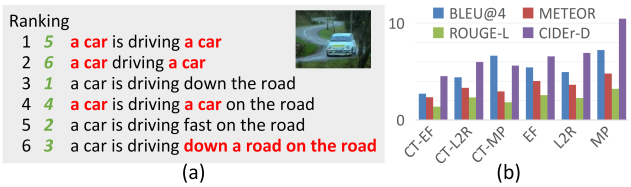


Figure 5: An example of picking the best caption among six caption candidates ($B = 6$) is shown in (a), where the numbers in green denote the ranking affected by teacher rescoring (TR) and the words in red are repeated contents. In (b), relative improvements (%) of different decoding algorithms when using TR on MSR-VTT are given.

exp	Model	B4	M	C_D
1	NACF	**42.0**	**28.7**	**51.4**
2	NACF w/o e^{src}	**42.0**	28.5	50.6
3	NACF w/o Ca.	41.5	28.5	49.8

Table 3: Effect of the copied source information (e^{src}) and the category tags (Ca.) on the performance on MSR-VTT test set in terms of BLEU@4 (B4), METEOR (M) and CIDEr-D (C_D).

iterations, which is expected because nouns and verbs form the major part (roughly 70%) of the vocabulary. Given that visual words are highly associated with the caption quality, how to utilize more semantic information to guide the refinement process deserves further consideration.

What makes our method different from previous ones is the flexible decoding paradigm, *i.e.*, words can be modified based on bidirectional context at a small cost of inference speed. We here measure the relative performance improvement during the intermediate process of iterative refinement. As shown in Fig. 4(a), all metrics are boosted, especially the BLEU@4 metric. This shows that refinement can make sentences more fluent and add more precise visually-grounded details. Therefore, we hope that this work will foster further research in flexible paradigms for caption generation.

Effect of Teacher Rescoring

We here present an example in Fig. 5 (a) to show how teacher rescoring (TR) influences the process of picking the best caption from B caption candidates. As we can observe, although two of six candidates ($B = 6$) are free from the repeated translation errors, without TR, the model fails to select either of them as the best caption due to the insensitivity to sentence fluency, which is resulted by the removal of sequential dependency in NA decoding. Considering the scoring information of an autoregressive counterpart, which measures how coherent a token is with its previous tokens, can mitigate this problem. So it is not surprising to see that in Fig. 5 (b), TR matters on performance , which is consistent with the observations in NMT (Gu et al. 2018; Mansimov, Wang, and Cho 2019; Wang et al. 2019). Specifically, the relative improvements brought by TR for CT based algorithms are less than that of original ones, indicating that

GT: a gold fish in a fish tank moving its mouth

_ _ _ _ _ _ _ aquarium
a fish is swimming around in the aquarium

_ _ _ _ _ _ _ ***tank***
a fish is swimming around in a tank

_ _ ***fish*** _ _ _ _ aquarium
a **gold fish** is swimming in the aquarium

_ _ ***man*** _ ***surfing*** _ _ ***sea***
(t = 1) a small man is surfing in the sea
(t > 1) a **gold fish** is swimming in the aquarium

The original results of our NACF
The results after manual control

Figure 6: Different descriptions can be generated after manually controlling the results of visual word generation.

the utilization of coarse-grained templates improves the sentence fluency to some extent. In short, both results demonstrate that retrieving sequential information to NA decoding is crucial, which is left to our future study. More qualitative examples are given in Fig. 7.

Effect of Auxiliary Information

We ablate the performance to measure the effect of the copied source information e^{src} and the category tags of MSR-VTT. As shown in Table 3, exp2 is worse than exp1, showing that enhancing the inputs of NA decoders is important (Guo et al. 2019). Comparing exp3 with exp1, we can conclude that the category information is complementary to visual features. Since such prior information is not always available, an auxiliary task of classifying videos can be integrated into the model in the future.

Robustness of NACF

Since the proposed visual word generation (VWG) is critical in our NACF, what if VWG captures irrelevant details from the video content? Here, we manually control (modify) the generated coarse-grained templates of VWG to figure out what will happen. As we can see in Fig. 6, before manual control, our NACF generates a visual word "aquarium", and the final description is related to the scene. Then at the first case, we change the word "aquarium" to "tank". As we can see, the captioning result ("swimming in a tank") is affected by our modification but it is still reasonable. At the second case, we add the word "fish" at the third position. Surprisingly, the final description contains a more precise detail, i.e., "gold fish". At the last case, we make the coarse-grained template completely irrelevant to the scene. We can see that our NACF is robust to re-convert these mistakes.

References

Banerjee, S.; and Lavie, A. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *ACL workshop*.

Bao, Y.; Zhou, H.; Feng, J.; Wang, M.; Huang, S.; Chen, J.; and Li, L. 2019. Non-autoregressive Transformer by Position Learning. *arXiv preprint arXiv:1911.10677* .

Bird, S.; Klein, E.; and Loper, E. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Gu, J.; Bradbury, J.; Xiong, C.; Li, V. O.; and Socher, R. 2018. Non-autoregressive neural machine translation. In *ICLR*.

Guo, J.; Tan, X.; He, D.; Qin, T.; Xu, L.; and Liu, T.-Y. 2019. Non-autoregressive neural machine translation with enhanced decoder input. In *AAAI*.

Hendrycks, D.; and Gimpel, K. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* .

Mansimov, E.; Wang, A.; and Cho, K. 2019. A generalized framework of sequence generation with application to undirected sequence models. *arXiv preprint arXiv:1905.12790* .

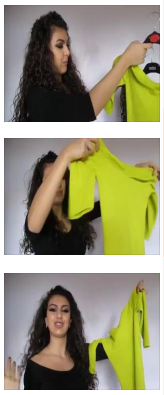
Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

Sigurdsson, G. A.; Russakovsky, O.; and Gupta, A. 2017. What actions are needed for understanding human actions in videos? In *ICCV*.

Vedantam, R.; Lawrence Zitnick, C.; and Parikh, D. 2015. Cider: Consensus-based image description evaluation. In *CVPR*.

Wang, Y.; Tian, F.; He, D.; Qin, T.; Zhai, C.; and Liu, T.-Y. 2019. Non-autoregressive machine translation with auxiliary regularization. In *AAAI*.

Figure 7: Examples of NACF (CT-MP) for picking the best caption among B candidates ($B = 6$), where the numbers in black denote the original ranking while in green denote the ranking affected by teacher rescoring. We mark the errors in captions in red. As we can observe, teacher rescoring is beneficial for picking a suitable description that sounds natural in most cases (the first three examples).



GT: a woman wearing a black shirt show us her clothes
AR-B: a woman is showing how to fold a piece of jeans
NA-B (init.): _ _ _ _ _
(t = 1): ***a woman is a dress***
(t = 2): a ***woman is a clothes***
(t = 3): a woman ***is a clothes***
(t = 4): a woman is ***talking something***
(t = 5): a woman is ***explaining*** something

NACF(CT): _ woman _ _ _ _ clothes — CT-MP
(t = 1): ***a woman is a a a clothes***
(t = 2): a woman is ***talking a a clothes***
(t = 3): a woman is ***talking a her*** clothes
(t = 4): a woman is talking ***about her*** clothes
(t = 5): a woman is ***talking*** about her clothes

NACF(CT): _ woman _ _ _ _ clothes — CT-EF
(t = 1): ***a*** woman _ _ _ _ clothes
(t = 2): a woman ***is*** _ _ _ clothes
(t = 3): a woman is ***talking*** _ _ clothes
(t = 4): a woman is talking ***about*** _ clothes
(t = 5): a woman is talking about ***her*** clothes
(t = 6): a ***woman*** is talking about her ***clothes***

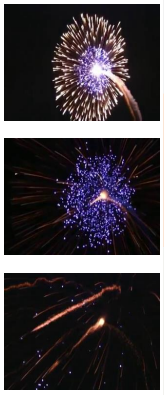
NACF(CT): _ woman _ _ _ clothes — CT-L2R
(t = 5): a ***woman*** is talking about ***clothes***

GT: a lego set is being filmed by a human being
AR-B: a person is playing a video game
NA-B (init.): _ _ _ _ _ _ _
(t = 1): ***a lego is a a a car***
(t = 2): a ***lego is is with*** a ***car***
(t = 3): a ***man*** is ***driving with a*** car
(t = 4): a man is ***talking with*** a car
(t = 5): a man is talking ***about*** a car

NACF(CT): _ _ _ _ _ _ _ _ _ _ — CT-MP
(t = 1): ***a lego of a a a a a a car***
(t = 2): a lego ***car is a is in*** a a ***car***
(t = 3): a ***lego truck*** is ***being around*** in ***a*** a car
(t = 4): a lego ***car*** is being ***driven in*** a a car
(t = 5): a lego car is being driven in a ***lego*** car

NACF(CT): _ _ _ _ _ _ vehicles — CT-EF
(t = 1): ***a*** _ _ _ _ _ vehicles
(t = 2): a _ ***is*** _ _ _ vehicles
(t = 3): a _ is _ _ ***lego*** vehicles
(t = 4): a _ is _ ***a*** lego vehicles
(t = 5): a ***person*** is _ a lego vehicles
(t = 6): a person is ***playing*** a lego vehicles
(t = 7): a person is playing a lego ***car***

NACF(CT): _ _ _ _ _ cars — CT-L2R
(t = 6): a lego car is being ***played***



GT: fireworks explode in the sky and the crowds cheer
AR-B: a person is playing a video game
NA-B (init.): _ _ _ _ _ _ _ _
(t = 1): ***a is is a a a a screen***
(t = 2): a ***man is a walking in*** a ***city***
(t = 3): a ***person*** is ***being around*** in a ***city***
(t = 4): a person is ***walking*** around in ***the sky***
(t = 5): a person is walking around in the ***sky***

NACF(CT): _ _ _ _ _ _ _ _ _ — CT-MP
(t = 1): ***a is of a a a a the sky***
(t = 2): a ***video*** of ***fireworks fireworks fireworks in*** the ***sky***
(t = 3): a video ***of*** fireworks ***is fireworks*** in ***the*** sky
(t = 4): a video of fireworks ***are shown*** in the ***sky***
(t = 5): a video of fireworks are ***shown*** in the sky

NACF(CT): _ _ _ _ _ _ sky — CT-EF
(t = 1): ***a*** _ _ _ _ _ sky
(t = 2): a _ _ _ _ ***the*** sky
(t = 3): a _ _ _ ***in*** the sky
(t = 4): a _ ***is*** _ in the sky
(t = 5): a ***person*** is _ in the sky
(t = 6): a person is ***walking*** in the sky
(t = 7): a person is walking in the ***sky***

NACF(CT): _ _ _ _ _ _ sky — CT-L2R
(t = 7): a man is talking about the ***sky***

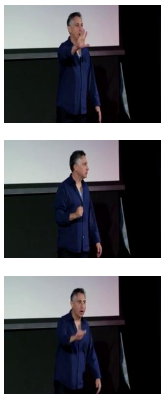
GT: girl in pink dress fashion model walking in ramp
AR-B: there is a woman is walking on the floor
NA-B (init.): _ _ _ _ _ _ _ _ _
(t = 1): ***a woman is a a a a a stage***
(t = 2): a ***woman is a a talking in*** a ***stage***
(t = 3): a woman is ***talking a stage in*** a ***stage***
(t = 4): a woman is talking a ***song on a*** stage
(t = 5): a woman is ***singing*** a song on a stage

NACF(CT): _ _ _ _ _ _ _ show — CT-MP
(t = 1): ***a woman is a a a fashion*** show
(t = 2): a ***woman is walking in*** a fashion ***show***
(t = 3): a ***woman is talking in*** a fashion show
(t = 4): a woman ***is walking*** in a fashion show
(t = 5): a ***woman*** is walking in a fashion show

NACF(CT): _ _ _ _ _ _ _ show — CT-EF
(t = 1): ***a*** _ _ _ _ _ _ show
(t = 2): a _ _ _ _ ***a*** _ show
(t = 3): a _ _ _ _ a ***fashion*** show
(t = 4): a _ _ _ ***in*** a fashion show
(t = 5): a _ ***is*** _ in a fashion show
(t = 6): a ***woman*** is _ in a fashion show
(t = 7): a woman is ***walking*** in a fashion show
(t = 8): a woman is walking in a fashion ***show***

NACF(CT): _ _ _ _ _ _ _ show — CT-L2R
(t = 8): a woman is talking about a fashion ***show***

Figure 8: Qualitative results on MSR-VTT, where ground-truth sentences (GT), captions generated by two baselines (AR-B and NA-B) and our NACF are presented. The generated process of NA-B (MP), NACF (CT-MP) and NACF (CT-EF) is highlighted, i.e., the words in bold and italic denote the update at each iteration. We mark the errors in read and keywords in green.

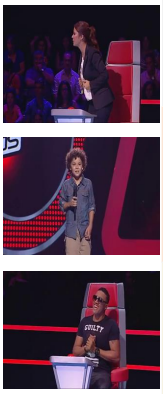


GT: a man in a blue shirt asks an audience to say
AR-B: a man is giving a lecture
NA-B (init.): _ _ _ _ _
(t = 1): ***a man is a presentation***
(t = 2): a ***man is a presentation***
(t = 3): a man ***is a presentation***
(t = 4): a man is ***talking something***
(t = 5): a man is ***explaining*** something

NACF(CT): _ man _ _ _ _ _ _ _ _ CT-MP
(t = 1): ***a man in a a a a a a stage***
(t = 2): a man in a ***blue shirt talking talking a talking***
(t = 3): a man in a ***blue*** shirt ***is talking a talking***
(t = 4): a man in a blue ***shirt*** is ***giving*** a ***room***
(t = 5): a man in a blue shirt is giving a ***presentation***

NACF(CT): _ man _ _ _ lecture CT-EF
(t = 1): ***a*** man _ _ _ lecture
(t = 2): a man _ _ ***a*** lecture
(t = 3): a man ***is*** _ a lecture
(t = 4): a man is ***giving*** a lecture
(t = 5): a ***man*** is giving a ***lecture***

NACF(CT): _ man _ _ _ _ _ _ _ _ CT-L2R
(t = 10): a ***man*** in a blue shirt is talking about something

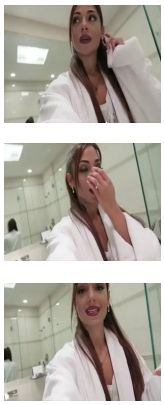


GT: judges expresses how much they liked a contestant on the voice
AR-B: a man is talking to a woman
NA-B (init.): _ _ _ _ _ _
(t = 1): ***a man is on a show***
(t = 2): a ***man is on*** a ***stage***
(t = 3): a man is ***talking a stage***
(t = 4): a man is talking ***about stage***
(t = 5): a man is talking ***on*** stage

NACF(CT): _ _ _ _ _ _ _ _ _ _ voice CT-MP
(t = 1): ***a man is a a a a a the*** voice
(t = 2): a ***man*** is ***talking to a talking on*** the voice
(t = 3): a man ***is*** talking ***to a judges*** on the ***voice***
(t = 4): a man is ***talking*** to a ***contestant on*** the voice
(t = 5): a man is talking to a ***contestant*** on the voice

NACF(CT): _ _ _ _ _ _ voice CT-EF
(t = 1): ***a*** _ _ _ _ _ voice
(t = 2): a _ _ _ _ ***the*** voice
(t = 3): a _ _ _ ***on*** the voice
(t = 4): a _ ***is*** _ on the voice
(t = 5): a _ is ***talking*** on the voice
(t = 6): a ***man*** is talking on the voice
(t = 7): a man is talking on the voice

NACF(CT): _ _ _ _ _ _ _ _ voice CT-L2R
(t = 9): a man is talking to a panel of ***judges***

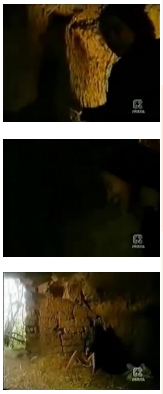


GT: brunette woman in white clothes is talking in the bathroom
AR-B: there is a woman is talking to a man
NA-B (init.): _ _ _ _ _ _ _
(t = 1): ***a woman is a a a dress***
(t = 2): a woman ***is a a a hair***
(t = 3): a woman is ***talking a a hair***
(t = 4): a woman is talking ***about*** a ***woman***
(t = 5): a woman is talking about ***a*** woman

NACF(CT): _ _ _ _ _ _ _ _ _ _ _ _ CT-MP
(t = 1): ***a*** woman is a a a and a in a camera
(t = 2): a woman ***in*** a ***black and a talking in*** a ***room***
(t = 3): a woman in a ***black is is talking*** in a ***room***
(t = 4): a woman in ***a*** black ***dress*** is talking in ***a*** room
(t = 5): a woman in a ***white*** dress is talking in a room

NACF(CT): _ woman _ _ _ _ _ CT-EF
(t = 1): ***a*** woman _ _ _ _ _
(t = 2): a woman ***is*** _ _ _ _
(t = 3): a woman is ***talking*** _ _ _
(t = 4): a woman is talking ***about*** _ _
(t = 5): a woman is talking about ***her*** _
(t = 6): a woman is talking about her ***hair***
(t = 7): a ***woman*** is talking about her hair

NACF(CT): _ woman _ _ _ _ _ CT-L2R
(t = 7): a ***woman*** is talking about her hair



GT: a person is walking through a cave
AR-B: a man is talking about a movie
NA-B (init.): _ _ _ _ _ _ _
(t = 1): ***a man is a a a forest***
(t = 2): a ***man is walking in*** a ***tree***
(t = 3): a man is ***walking a a forest***
(t = 4): a man is walking ***in*** a ***forest***
(t = 5): a man is walking in a ***forest***

NACF(CT): _ _ _ _ _ _ _ _ cave CT-MP
(t = 1): ***a man is walking a in a*** cave
(t = 2): a man ***is a walking in*** a ***cave***
(t = 3): a ***man*** is ***walking something in*** a cave
(t = 4): a man is walking ***around in*** a cave
(t = 5): a man is walking ***around*** in a cave

NACF(CT): _ _ _ _ _ _ _ _ cave CT-EF
(t = 1): ***a*** _ _ _ _ _ _ _ cave
(t = 2): a ***man*** _ _ _ _ _ cave
(t = 3): a man ***is*** _ _ _ _ cave
(t = 4): a man is _ _ _ ***a*** cave
(t = 5): a man is ***walking*** _ _ a cave
(t = 6): a man is walking _ ***in*** a cave
(t = 7): a man is walking ***around*** in a cave
(t = 7): a man is walking around in a ***cave***

NACF(CT): _ _ _ _ _ _ _ forest CT-L2R
(t = 9): a man is walking in the ***woods***

Figure 9: Qualitative results on MSR-VTT, where ground-truth sentences (GT), captions generated by two baselines (AR-B and NA-B) and our NACF are presented. The generated process of NA-B (MP), NACF (CT-MP) and NACF (CT-EF) is highlighted, i.e., the words in bold and italic denote the update at each iteration. We mark the errors in read and keywords in green.

GT: a celebrity talking on a tv show
AR-B: two men are talking to each other
NA-B (init.): _ _ _ _ _
(t = 1): ***a man is a show***
(t = 2): a ***man is a interviewed***
(t = 3): ***a*** man is ***talking interviewed***
(t = 4): a man ***is being*** interviewed
(t = 5): a man is being ***interviewed***

NACF(CT): _ men _ _ _ _ _ show — CT-MP
(t = 1): ***two*** men ***are on on a talk*** show
(t = 2): two ***men are talking on*** a ***talk*** show
(t = 3): two men ***are*** talking ***on a talk*** show
(t = 4): two men are talking ***on*** a ***talk*** show
(t = 5): two men are ***talking*** on a talk show

NACF(CT): _ men _ _ _ _ _ show — CT-EF
(t = 1): ***two*** men _ _ _ _ _ show
(t = 2): two men _ _ _ ***a*** _ show
(t = 3): two men _ ***talking*** _ a _ show
(t = 4): two men ***are*** talking _ a _ show
(t = 5): two men are talking _ a ***talk*** show
(t = 6): two men are talking ***on*** a talk show
(t = 7): two ***men*** are talking on a talk ***show***

NACF(CT): _ men _ _ _ _ _ show — CT-L2R
(t = 7): two ***men*** are talking on a tv ***show***

GT: a man is performing a science experiment
AR-B: a woman is talking about a product
NA-B (init.): _ _ _ _ _ _ _
(t = 1): ***a man is a a a experiment***
(t = 2): a man ***is a a a experiment***
(t = 3): a man is ***talking a a experiment***
(t = 4): a man is ***talking about*** a experiment
(t = 5): a man is talking about a ***science***

NACF(CT): _ man _ _ _ _ _ lab — CT-MP
(t = 1): ***a*** man ***is about about in a*** *lab*
(t = 2): a man is ***talking about a a lab***
(t = 3): a man is ***talking*** about ***a science science***
(t = 4): a man is talking about a ***science experiment***
(t = 5): a man is talking about a science ***experiment***

NACF(CT): _ man _ _ _ _ experiment — CT-EF
(t = 1): ***a*** man _ _ _ _ experiment
(t = 2): a man ***is*** _ _ _ experiment
(t = 3): a man is _ ***about*** _ experiment
(t = 4): a man is ***talking*** about _ experiment
(t = 5): a man is talking about ***a*** experiment
(t = 6): a ***man*** is talking about a *science*

NACF(CT): _ _ _ _ _ experiment — CT-L2R
(t = 6): a man is talking about ***science***

GT: a man is working on a roof
AR-B: a man is showing how to use a bicycle
NA-B (init.): _ _ _ _ _ _
(t = 1): ***a man is a a floor***
(t = 2): a man ***is a a floor***
(t = 3): a man is ***doing a bike***
(t = 4): a man is ***holding*** a ***bicycle***
(t = 5): a man is holding a ***bicycle***

NACF(CT): _ man _ _ _ _ roof — CT-MP
(t = 1): ***a*** man ***is a on a*** roof
(t = 2): a man is ***doing a a roof***
(t = 3): a man is ***working on*** a ***roof***
(t = 4): a man is ***working*** on a ***roof***
(t = 5): a man is working on ***a*** roof

NACF(CT): _ man _ _ _ _ roof — CT-EF
(t = 1): ***a*** man _ _ _ _ roof
(t = 2): a man ***is*** _ _ _ roof
(t = 3): a man is _ _ ***a*** roof
(t = 4): a man is _ ***on*** a roof
(t = 5): a man is ***working*** on a roof
(t = 6): a ***man*** is working on a ***roof***

NACF(CT): _ man _ _ _ _ _ _ _ — CT-L2R
(t = 7): a ***man*** is doing a bicycle on a roof

GT: army helicopters bombing the terror regions
AR-B: there is a **man is flying in the sky**
NA-B (init.): _ _ _ _ _ _ _
(t = 1): ***a is is a a sky***
(t = 2): a ***man is on*** a ***plane***
(t = 3): a man is ***flying a plane***
(t = 4): a man is ***throwing*** a ***plane***
(t = 5): a man is ***throwing*** a plane

NACF(CT): _ _ _ _ _ sky — CT-MP
(t = 1): ***a is is on a*** sky
(t = 2): a ***man is on*** a ***helicopter***
(t = 3): a ***man*** is ***flying*** a ***helicopter***
(t = 4): a man is ***flying*** a ***helicopter***
(t = 5): a man is flying a ***helicopter***

NACF(CT): _ _ _ _ _ _ _ _ sky — CT-EF
(t = 1): ***a*** _ _ _ _ _ _ _ sky
(t = 2): a _ ***is*** _ _ _ _ _ sky
(t = 3): a ***man*** is _ _ _ _ _ sky
(t = 4): a man is _ _ _ _ ***a*** sky
(t = 5): a man is _ ***a*** _ _ a sky
(t = 6): a man is ***flying*** a _ _ a sky
(t = 7): a man is flying a _ ***in*** a sky
(t = 8): a man is flying a ***helicopter*** in a sky
(t = 9): a man is flying a helicopter in a ***helicopter***

NACF(CT): _ _ _ _ _ _ sky — CT-L2R
(t = 6): a man is flying on the ***sky***

Figure 10: Qualitative results on MSR-VTT, where ground-truth sentences (GT), captions generated by two baselines (AR-B and NA-B) and our NACF are presented. The generated process of NA-B (MP), NACF (CT-MP) and NACF (CT-EF) is highlighted, i.e., the words in bold and italic denote the update at each iteration. We mark the errors in read and keywords in green.