iCap: Interactive Image Captioning with Predictive Text

Zhengxiong Jia and Xirong Li*
Key Lab of DEKE, Renmin University of China
Vistel AI Lab, Visionary Intelligence Ltd.
Beijing, China

ABSTRACT

In this paper we study a brand new topic of interactive image captioning with human in the loop. Different from automated image captioning where a given test image is the sole input in the inference stage, we have access to both the test image and a sequence of (incomplete) user-input sentences in the interactive scenario. We formulate the problem as *Visually Conditioned Sentence Completion (VCSC)*. For VCSC, we propose *ABD-Cap*, asynchronous bidirectional decoding for image caption completion. With ABD-Cap as the core module, we build iCap, a web-based interactive image captioning system capable of predicting new text with respect to live input from a user. A number of experiments covering both automated evaluations and real user studies show the viability of our proposals.

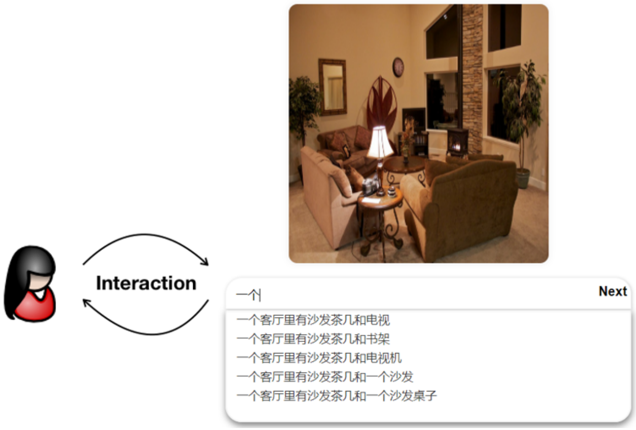
CCS CONCEPTS

• **Human-centered computing** → **Text input**.

KEYWORDS

Image captioning, human-computer interaction, sentence completion, deep learning

ACM Reference Format:
Zhengxiong Jia and Xirong Li. 2020. iCap: Interactive Image Captioning with Predictive Text. In *2020 International Conference on Multimedia Retrieval (ICMR'20), June 8–11, 2020, Dublin, Ireland.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3372278.3390697



一个

一个客厅里有沙发茶几和电视
一个客厅里有沙发茶几和书架
一个客厅里有沙发茶几和电视机
一个客厅里有沙发茶几和一个沙发
一个客厅里有沙发茶几和一个沙发桌子

Figure 1: User interface of the proposed interactive image captioning system. Given both the image and live input from a user, the system interacts with the user by predicting new text to complete the user input. Note that while we target at *Chinese*, which is the authors' first language, major conclusions of this research shall be language-independent.

1 INTRODUCTION

Automated description of visual content, let it be image or video, is among the core themes for multimedia content analysis and retrieval. Novel visual captioning models are being actively developed [5, 8, 10, 13] with increasing performance reported on public benchmark datasets such as MS-COCO [6] and MSR-VTT [24]. Nonetheless, the success of the state-of-the-art models largely depends on the availability of many well-annotated training examples in a specific domain. It is apparent that an image captioning model learned from MS-COCO does not work for medical images, while a model learned from annotations with respect to human activities is ineffective for describing general images [15]. It is also recognized that one cannot directly use a dataset described in one language to build models for another language [11, 14]. Therefore, in the days to come, when one wants to build an effective image captioning model for a domain where annotations are in short supply, manual annotation remains indispensable.

Writing image descriptions is known to be laborious, even for crowd sourcing. Thus an important research question arises: *Can manual annotation be performed in a more intelligent manner other than fully manual*? Li *et al.* [15] develop a recommendation assisted annotation system, where a user is presented with five sentences automatically recommended by the system based on the pictorial content. Cornia *et al.* [7] propose a framework that allows a user to supervise the caption generation process by specifying a set of detected objects as the control signal. Neither of them is *interactive*.

In this paper, we study a novel topic of *interactive image captioning* with human in the loop. Recall that in automated image captioning, a given test image is the sole input in the inference stage. By contrast, an interactive system has to consider live input from a specific user and respond on the fly, see Fig. 1. Thus, the interactive scenario is more challenging. A desirable system shall be capable of predicting new text that refines or completes the current text provided by the user. Note that such a task conceptually resembles to some extent query auto completion (QAC) [3] in information retrieval, as both perform text completion. Major difference is two-fold. First, QAC has no visual input. Second, different from search

[*] Corresponding author: Xirong Li (xirong.li@gmail.com)

ICMR '20, June 8–11, 2020, Dublin, Ireland

ACM ISBN 978-1-4503-7087-5/20/06…$15.00
https://doi.org/10.1145/3372278.3390697

queries that are typically keywords or phrases, image captions are natural language text that contains richer contextual information. We term the new task *Visually Conditioned Sentence Completion* (VCSC). To summarize, our contributions are as follows:

- To the best of our knowledge, this is the first work on interactive image captioning, where humans and deep captioning models interact with each other to accomplish the captioning task.
- We identify a key task in the interactive scenario, namely Visually Conditioned Sentence Completion (VCSC). We tackle the task by proposing asynchronous bidirectional decoding for image caption completion (ABD-Cap).
- We verify our proposal by developing iCap, a web-based interactive annotation system that responds to user input in real time. Both automated evaluations and real user studies justify the effectiveness of the proposed ABD-Cap model and iCap system.

2 RELATED WORK

Automated image captioning. A number of deep learning based methods have been proposed for automated image captioning [2, 19, 21]. These methods mainly follows an encoding-and-decoding workflow. A given image is encoded into a dense vector by first extracting visual features by a pre-trained convolutional neural network (CNN) and then reduce the features to the dense vector either by affine transformation [21] or by an spatially aware attention module [2]. Even though this line of work has achieved extraordinary performance on several benchmarking datasets, there are still limitations when applied in more complex scenarios especially when humans want to get control of the captioning process. Recent work by Cornia *et al.* [7] proposes a framework that allows human to control the caption generation process by specifying a set of detected objects as the control signal. Our work goes further in this direction, investigating interactive image captioning with human in the loop.

Interactive image labeling. In the context of semantic image segmentation, several works have been done to speed up the pixel labeling process using interactive techniques. Representative works are the polygon-rnn series including polygon-rnn [4], polygon-rnn++ [1] and curve-gcn [16], which focus on producing polygonal annotations of objects interactively with humans-in-the-loop. An interactive video captioning system is presented in [23], where a captioning agent asks for a short prompt from a human. As the prompt is treated as a clue of the human's expectation rather than an (incomplete) caption, that work does not consider sentence completion. To the best of our knowledge, we are the first to devise an interactive annotation tool for image captioning.

Bidirectional decoding. Research on utilizing bidirectional decoding for image captioning exists [20, 22]. In [20], a backward decoder is used only in the training stage to encourage the generative RNNs to plan ahead. As for [22], a bidirectional LSTM (Bi-LSTM) combining two separate LSTM networks is developed to capture long-term visual-language interactions. Predictions at a specific time step is fully determined by the forward and backward hidden states at that step. Different from Bi-LSTM, the two LSTMs used in our ABD-Cap model work in an asynchronous manner that is suited for the VCSC task. This design is inspired by ABD-NMT [25] in the machine translation field. We adapt the backward decoding process to generate a fixed-length sequence of backward hidden vectors and change the training procedure from joint training to training two decoders sequentially. We find in preliminary experiments that the adaption is better than the original ABD for image captioning.

3 HUMAN-IN-THE-LOOP IMAGE CAPTIONING

3.1 Problem Formalization

For writing a sentence to describe a given image I, a user typically conducts multiple rounds of typing and editing. In such a manual annotation session, one naturally generates a sequence of (incomplete) sentences, denoted as $\{S_i|i = 1, \ldots, T\}$, where S_i indicates the sentence after i rounds and T is the number of rounds in total. Accordingly, S_1 is the initial user input, while S_T is the final annotation. Our goal is to devise an image captioning (iCap) framework so that the user can reach S_T in fewer rounds and with reduced annotation workload.

Given the image I and the user input S_i, it is reasonable to assume that S_i is thus far the most relevant with respect to I. Hence, we shall take both into account and accordingly suggest k candidate sentences that complete S_i. Apparently, this scenario differs from automated image captioning where the image is the sole input. We term the new task as *Visually Conditioned Sentence Completion* (VCSC). Image captioning can be viewed as a special case of VCSC that has no user input.

Given an iCap system equipped with a VCSC model, a user annotates an image in an iCap session, which is illustrated in Fig. 2 and described as follows. A session starts once the user types some initial text, *i.e.,* S_1. At the i-th round ($i \geq 2$), the system presents to the user k candidate sentences $\{\hat{S}_{i,1}, \hat{S}_{i,2}, \ldots, \hat{S}_{i,k}\}$ generated by the VCSC model. The user produces S_i by either selecting one of the suggested sentence or revising S_{i-1}. The session closes when the user chooses to submit the last annotation S_T.

From the above description we see that a desirable VCSC model shall not only cope with the multi-modal input but also needs to respond in real time. Next, we propose a model that fulfills these two requirements.

3.2 Visually Conditioned Sentence Completion

We develop our model for VCSC based on Show-and-Tell [21], a classical model for automated image captioning. To make the paper more self-contained, we describe briefly how Show-and-Tell generates a sentence for a given image. Accordingly, we explain difficulties in directly applying the model for the VCSC task.

Show-and-Tell generates a sentence by iteratively sampling words from a pre-specified vocabulary of m distinct words $\{w_1, \ldots, w_m\}$. Note that in addition to normal words, the vocabulary contains three special tokens, *i.e., start*, *end* and *unk*, which indicate the beginning, the ending of a sentence and out-of-vocabulary (OOV) words. Let $p_t \in R^m$ be a probability vector, each dimension of which indicates the probability of the corresponding word to be sampled at the t-th iteration. Show-and-Tell obtains p_t using a Long-Short Term Memory (LSTM) network [9]. In particular, given $h_t \in R^d$ as

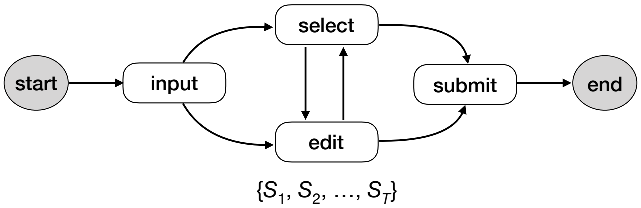


Figure 2: User state transitions in an iCap session. The session starts once a user types something as an initial input S_1. The iCap system tries to complete the input by suggesting k sentences, denoted as $\{\hat{S_1}, \ldots, \hat{S_k}\}$. The user performs the captioning task by switching between the *select* and *edit* states, resulting in a sequence of sentences to be completed $\{S_i | i = 2, 3, \ldots\}$. The session ends once the user *submits* the final annotation S_T.

the LSTM's hidden state vector, p_t is obtained by feeding h_t into a $d \times m$ fully connected (FC) layer followed by a softmax layer, *i.e.,*

$$\begin{array}{rcl} p_t & = & \text{softmax}(\text{FC}(h_t)), \\ \hat{w}_t & = & \text{argmax}_w(p_t). \end{array} \tag{1}$$

The word with the largest probability is sampled. Consequently, the hidden state vector is updated as

$$h_{t+1} = \text{LSTM}(h_t, \hat{x}_t), \tag{2}$$

where $\hat{x}_t$ is the embedding vector of $\hat{w}_t$. To make the generation process visually conditioned, let $v(I)$ be a visual representation of the given image extracted by a pre-trained CNN. LSTM is initialized with $h_0 = \mathbf{0}$ and $\hat{x}_0$ as the visual embedding vector obtained by feeding $v(I)$ into an FC layer.

3.2.1 Show-and-Tell for VCSC. Assume the user input S_i has n words $\{w_{i,1}, \ldots, w_{i,n}\}$. Let c be the user-specified cursor position, suggesting where the user wants to edit. Accordingly, S_i is divided into two substrings, *i.e.,* $S_{i,l} = \{w_{i,1}, \ldots, w_{i,c}\}$ and $S_{i,r} = \{w_{i,c+1}, \ldots, w_{i,n}\}$. As the sampling process of Show-and-Tell is unidirectional, $S_{i,r}$ has to be omitted. Note that such a deficiency cannot be resolved by reversing the sampling process, as $S_{i,l}$ will then be ignored. Consequently, Show-and-Tell is unable to fully exploit the user-provided information.

In order to inject the information of $S_{i,l}$ into the LSTM network, we update the hidden state vector by Eq. 2 except that the word sampled at each step is forced to be in line with $\{w_{i,1}, \ldots, w_{i,c}\}$. Afterwards, the regular sampling process as described by Eq. 1 is applied until the *end* token is selected. In order to obtain k candidates, a beam search of size k is performed. The best k decoded beams are separately appended to $S_{i,l}$ to form $\{\hat{S}_{i,1}, \ldots, \hat{S}_{i,k}\}$.

3.2.2 Proposed ABD-Cap for VCSC. In order to effectively exploit the user input on both sides of the cursor, we propose Asynchronous Bidirectional Decoding for image caption completion (ABD-Cap). The main idea is to deploy two asynchronous LSTM based decoders, see Fig 3. One decoder is to model the entire user input in a backward manner so that $S_{i,r}$ is naturally included. The other decoder is responsible for sentence generation similar to Show-and-Tell except that it receives information from the first decoder through a seq2seq attention module [17]. The two decoders are separately trained, and cooperate together in an asynchronous manner for sentence completion.

Backward decoder. Our backward decoder is trained using reserved image captions, and thus learns to generate backward hidden state vectors $\{\overleftarrow{h}_t\}$ from right to left, *i.e.,*

$$\begin{array}{rcl} \overleftarrow{h}_t & = & \text{LSTM}(\overleftarrow{h}_{t+1}, \hat{x}_{t+1}), \\ \overleftarrow{p}_t & = & \text{softmax}(\text{FC}(\overleftarrow{h}_t)), \\ \hat{w}_t & = & \text{argmax}_w(\overleftarrow{p}_t). \end{array} \tag{3}$$

In order to cope with user input of varied length, we sample N rounds, where N is the maximum sequence length. This results in a fixed-length sequence of backward hidden state vectors $\overleftarrow{H} = \{\overleftarrow{h}_0, \overleftarrow{h}_1, \ldots, \overleftarrow{h}_N\}$. For injecting the information of S_i into the sequence, we first conduct the regular sampling procedure for $N - n$ steps. Afterwards, the word sampled per step is forced to be in line with $\{w_{i,n}, \ldots, w_{i,1}\}$.

Forward decoder with attention. In order to utilize the information from the backward decoder, we employ the seq2seq attention mechanism [17]. In particular, $\overleftarrow{H}$ is converted to a vector m_t that encodes the backward contextual information. Then, by substituting m_t for $\hat{x}_t$ in Eq. 2, the backward information is embedded into the word sampling process. We express the above as

$$\begin{array}{rcl} a_t & = & \text{softmax}(\text{ReLU}(\text{FC}(\hat{x}_t, h_t)), \\ m_t & = & \sum_{i=1}^{N} a_{t,i} \cdot \overleftarrow{h}_i, \\ h_{t+1} & = & \text{LSTM}(h_t, m_t), \end{array} \tag{4}$$

where a_t is a N-dimensional attention weight vector.

For the rest of the VCSC task, we follow the same procedure as depicted in Section 3.2.1.

3.3 Training Models for VCSC

Since our target language is Chinese, we train the models on the COCO-CN dataset [15]. This public dataset contains 20,342 MS-COCO images annotated with 27,218 Chinese sentences. We follow the original data split, with 18,342 images for training, 1,000 images for validation and 1,000 images for test. For image features we use the provided 2,048-dim CNN features, extracted using a pretrained ResNeXt-101 model [18].

All models are trained in a standard supervised manner, with the cross entropy loss minimized by the Adam optimizer. The initial learning rate is set to be 0.0005. We train for 80 epochs at the maximum. Best models are selected based on their CIDEr scores on the validation set.

Note that in the interactive scenario, a user might provide OOV words. To alleviate the issue, different from previous works [14, 15] that perform word-level sentence generation, our models compose a sentence at the character level. As shown in Section 4.1, this choice substantially reduces the occurrence of OOV words.

For user study, we build a web-based iCap system, with its user interface shown in Fig. 1. Given a specific user input, it suggests $k = 5$ sentences in approximately 70 milliseconds, which is sufficiently fast for real-time interaction.

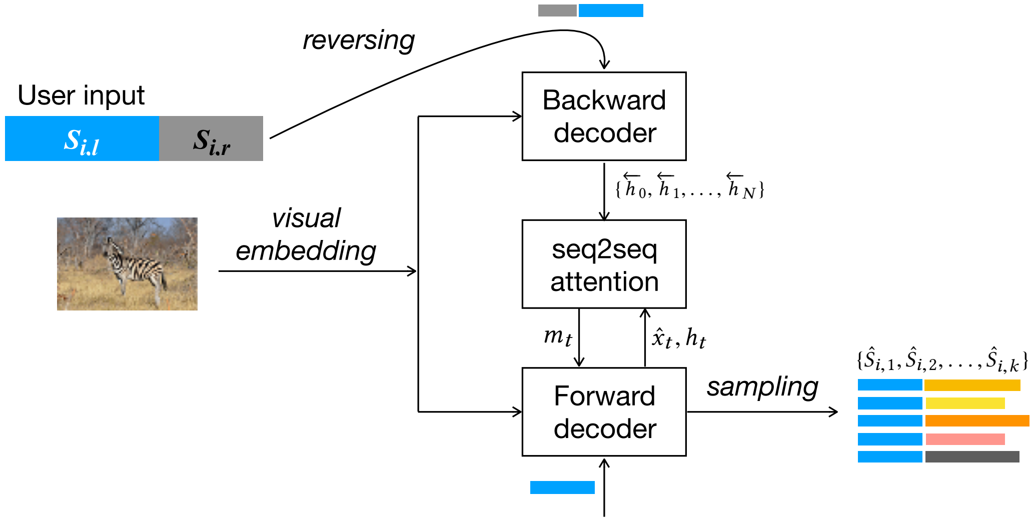

reversing
User input
$S_{i.l}$
$S_{i.r}$
visual embedding
Backward decoder
$\{\overleftarrow{h}_0, \overleftarrow{h}_1, \ldots, \overleftarrow{h}_N\}$
seq2seq attention
m_t
$\hat{x}_t, h_t$
Forward decoder
sampling
$\{\hat{S}_{i,1}, \hat{S}_{i,2}, \ldots, \hat{S}_{i,k}\}$

Figure 3: Diagram of the proposed ABD-Cap model for visually conditioned sentence completion. Given an image and a user-input sentence S_i which is separated into two substrings $S_{i,l}$ and $S_{i,r}$ by the user-specified cursor, the forward encoder exploits multi-source information (from the image and the backward decoder) to complete S_i.

4 EVALUATION

Unlike automated image captioning, there lacks a well established evaluation protocol for interactive image captioning. We need to understand how a user interacts with the iCap system and evaluate to what extent the system assists the user. To that end, we propose a two-stage evaluation protocol as follows:

- **Stage I**. Performed before commencing a real user study, evaluations conducted at this stage are to assess the validity of major implementation choices of the module for VCSC used in the iCap system.
- **Stage II**. Evaluations at this stage are performed after running the iCap system for a while with adequate data collected to analyze the usability of iCap in multiple aspects.

4.1 Stage-I Evaluation

4.1.1 Setup. With no user input provided, the VCSC task is equivalent to automated image captioning. So we first evaluate the proposed ABD-Cap model in the automated setting.

Baselines. As described in Section 3.2, compared to the Show-and-Tell model [21] used in [15] for Chinese captioning, we make two main changes. That is, the proposed asynchronous bidirectional decoder and character-level sentence generation. In order to verify the necessity of these two changes, we compare with the following four baselines:

- Show-and-Tell with a word-level forward decoder [15].
- Show-and-Tell with a word-level backward decoder.
- Show-and-Tell with a character-level forward decoder.
- Show-and-Tell with a character-level backward decoder.

For a fair comparison, the four baselines are all trained in the same setting as ABD-Cap. All the models are evaluated on the 1,000 test images of COCO-CN [15], each of which is associated with five manually written Chinese captions.

Evaluation criteria. We report BLEU-4, METEOR, ROUGE-L and CIDEr, commonly used for automated image captioning. Note that computing the four metrics at the character level is not semantically meaningful. So for a sentence generated by a character-level decoder, we employ Jieba[1], an open-source toolkit for Chinese word segmentation, to tokenize the sentence to a list of words. The presence of the *unk* token in a generated sentence means an OOV word is predicted, which negatively affects user experience. Therefore, for each model we calculate the OOV rate, *i.e.,* the number of sentences containing *unk* divided by the number of generated sentences.

4.1.2 Results. The overall performance of each model on the COCO-CN test set is presented in Table 1. As we can see from the table, our proposed ABD-Cap model outperforms the baselines on all of the four caption evaluating metrics. Even more, the character-level models possess a zero OOV rate that is way less than the word-level models. The result justifies the superiority of the proposed character-level sentence generation for reducing OOVs. Therefore, the character-level ABD-Cap is deployed in the iCap system for the following real user study.

4.2 Stage-II Evaluation

We first collect real-world interaction data from a user study. We then analyze the data in details to understand how users and the system interacted and to what extent the system assisted users to accomplish the annotation task.

4.2.1 User Study. To avoid any data bias towards COCO-CN, we constructed our annotation pool by randomly sampling images from MS-COCO with COCO-CN images excluded in advance.

Nineteen members in our lab, 14 males and 5 females, participated as volunteers in this user study. While mostly majored in computer science, the majority of the subjects have no specific

[1] https://github.com/fxsjy/jieba

Table 1: Performance of different models for VCSC without any user input, *i.e.,* automated image captioning. *Char* and *Word* represent character-level and word-level decoders, respectively. Our proposed ABD-Cap model outperforms the baselines. Moreover, the character-level ABD-Cap has zero OOV rate.

	BLEU_4		METEOR		ROUGE_L		CIDEr		OOV rate (%)	
Model	*Char*	*Word*	*Char*	*Word*	*Char*	*Word*	*Char*	*Word*	*Char*	*Word*
Show-and-Tell	24.0	25.5	27.1	27.4	47.8	48.4	70.6	72.4	0	3.8
Show-and-Tell, backward decoder	23.0	23.9	26.6	27.0	47.0	48.2	68.2	71.5	0	5.0
proposed ABD-Cap	24.4	**26.0**	**27.6**	27.3	48.3	**48.6**	71.7	**73.7**	0	7.4

knowledge about the iCap project. So they can be considered as average users for interactive image captioning. Our subjects performed the annotation task in their spare time. User actions in each session such as the editing history of the input box and the selection operation from the drop-down list were logged for subsequent analysis.

The user study lasted for one month, with 2,238 images annotated and 2,238 sentences in total. Table 2 summarizes main statistics of the gathered data. Depending on whether the system-suggested sentences were selected, user annotation is divided into two modes, *i.e., fully manual* and *interactive*. For 793 out of the 2,238 sentences (35.4%), they were written in the interactive mode, *i.e.,* users selected the suggested sentences at least once. Moreover, the sentences written in the interactive mode tend to be longer and thus more descriptive than their fully manual counterparts. Also note the relatively smaller number of editing rounds in the interactive mode. These results encouragingly suggest that user-system interactions produce better annotations in a shorter time.

4.2.2 Analysis of an iCap Session. We now analyze user-system interactions at the session level. The editing history of a specific user was recorded as a sequence of sentences $\{S_1, \ldots, S_T\}$, generated by scanning the input box every 0.2 second. If the text in the input changes between two consecutive scans, we consider an editing operation occur. To quantize the changes, we compute the Levenshtein distance[2] (LevD) [12] between S_i and S_{i+1}. On the basis of the pair-wise LevD, we derive the following two metrics:

- *Accumulated LevD*, computed over the sequence $\{S_1, \ldots, S_T\}$ by summing up all pair-wise LevD values. This metric reflects the overall amount of editing conducted in an iCap session. Smaller is better.
- $LevD(S, S_T)$ between an (incomplete) sentence S and the final annotation. This metric estimates human workload. Specifically, $LevD(\varnothing, S_T)$ measures human workload required in the fully manual mode. Smaller is better.

As Table 2 shows, the sentence sequences in the interactive mode has an averaged *Accumulated LevD* of 13.9, clearly smaller than that of the fully manual mode.

User behaviors. For the 793 interactive annotations, a total number of 996 selections were recorded, meaning 1.2 selections per session on average. As shown in Fig. 4, more than half of the selections are made on the top-1 suggested sentences. Meanwhile, the other suggestions also get selected but with their chances decreasing along their ranks. The result demonstrates the effectiveness of beam search as well as the necessity of presenting multiple suggestions to the user.

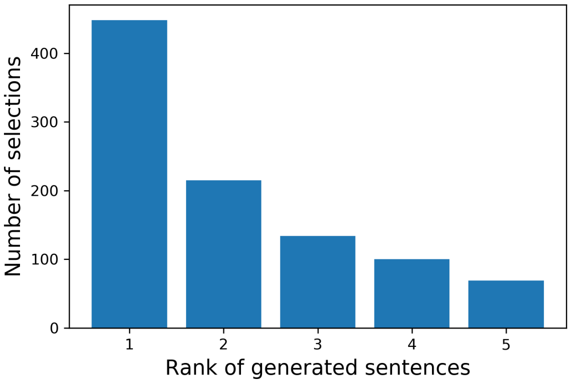


Figure 4: Distribution of selection with respect to the ranks of generated sentences. Sentences ranked at the top are more likely to be selected by users.

Influence of sentence completion. To study the direct influence of the sentences completed by the ABD-Cap model, let S_s be the completed sentence selected by a given user at a specific editing round. By definition, the pre-completion sentence is obtained as S_{s-1}. We then calculate $LevD(S_{s-1}, S_T)$ and $LevD(S_s, S_T)$, which respectively measures the distance of the pre-completion and post-completion sentences to the final annotation. The distribution curves of these two type of distances are shown in Fig. 5. The green curve which shows the distribution of $LevD(S_s, S_T)$ is skewed towards the left side. The result clearly shows that the sentence completion module helps the user input to converge to the final annotation, and thus reduces human workload.

4.2.3 ABD-Cap versus Show-and-Tell for VCSC. With the interaction data collected, we now perform a simulated experiment to compare two distinct instantiations for the sentence completion module, namely ABD-Cap *versus* Show-and-Tell.

Again, let S_s be the candidate sentence selected by a given user at a specific editing round, and accordingly we have access to S_{s-1} which is the user input of the sentence completion model to generate S_s. Now, instead of ABD-Cap, we use Show-and-Tell to generate

[2] The Levenshtein distance between two Chinese sentences is the minimum number of single Chinese character edits (insertions, deletions or substitutions) required to change one sentence into the other

Table 2: Major statistics of the interaction data collected in our user study. Smaller sequence length T, *Accumulated edits*, and Levenshtein distance (*LevD*) suggest less amount of human workload.

		Statistics per sentence		Statistics per iCap session			
Annotation mode	**Sentences**	*Num. words*	*Num. chars*	*Num. selections*	T	*Accumulated edits*	*Accumulated LevD*
Fully manual	1,445	9.0	15.3	0	11.6	10.6	18.9
Interactive	793	9.8	16.7	1.2	9	8	13.9

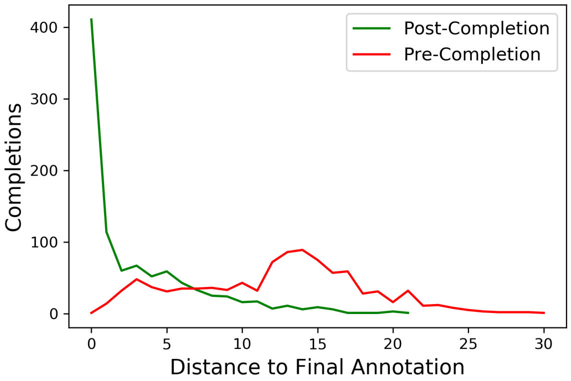


Figure 5: Distribution of pre-completion / post-completion sentences's LevD to the final annotation. Smaller LevD means post-completion sentences are more close to the final annotation.

five candidate sentences. Comparing these candidates to the final annotation S_T allows us to conclude which model provides better suggestions.

Table 3: LevD between sentences produced by distinct models to the final annotation. Lower LevD means a suggested sentence is more close to the final annotation, and thus implies lessened human interaction.

	Rank				
Model for VCSC	1	2	3	4	5
Show-and-Tell	5.69	5.93	6.05	6.21	6.43
ABD-Cap	4.99	5.17	5.28	5.56	5.76

Quantitative results. Table 3 shows LevD between sentences generated by the two models to the final annotation. Lower LevD means the generated sentences are more close to the final annotation. The proposed ABD-Cap outperforms Show-and-Tell at all ranks. Moreover, the LevD increases as the rank goes up. This result confirms our previous finding (Fig. 4) that the sentences ranked at the top describe images better and thus more likely to be selected by the user.

Qualitative results. To better understand the differences between the two models for sentence completion, we present some typical results in Table 4. In particular, the first two rows show cases where the user cursor is placed at the end of the current input text, *i.e.,* $S_{i,r}$ is empty. The last two rows show cases where the user cursor is placed in the middle. Compared to Show-and-Tell which considers only text at the left-hand side of the cursor, our ABD-Cap model exploits texts at both sides, and thus suggest sentences more close to the final annotation.

Limitation of this study. The current system assumes user input preceding the cursor to be noise free. The robustness of iCap to user input with noises such as typos and grammar errors needs further investigation. While sentences written in the interactive mode are longer and presumably more descriptive, their effectiveness as a better alternative to train an image captioning model has not been verified.

5 CONCLUSIONS

In this paper we have made a novel attempt towards image captioning with human in the loop. We develop iCap, an interactive image captioning system. We conduct both automated evaluations and user studies, allowing us to draw conclusions as follows. With the assistance of visually conditioned sentence completion, better image captions can be obtained with less amount of human workload. Moreover, asynchronous bidirectional decoding is found to be important for effectively modeling live input from a user during the interaction.

ACKNOWLEDGMENTS

This research was supported in part by National Natural Science Foundation of China (No. 61672523), Beijing Natural Science Foundation (No. 4202033), and the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (No. 18XNLG19).

REFERENCES

[1] D. Acuna, H. Ling, A. Kar, and S. Fidler. 2018. Efficient Interactive Annotation of Segmentation Datasets With Polygon-RNN++. In *CVPR.*

[2] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *CVPR.*

[3] F. Cai and M. Rijke. 2016. A Survey of Query Auto Completion in Information Retrieval. *Foundations and Trends in Information Retrieval* (2016).

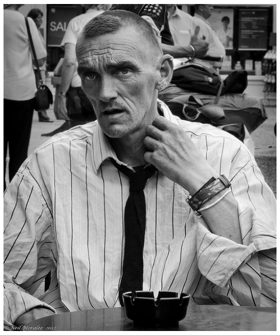
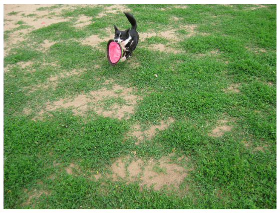
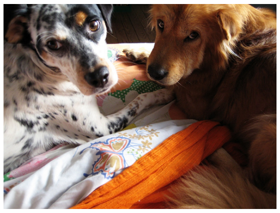
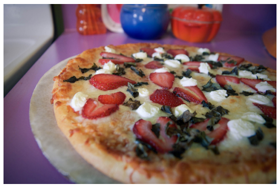
[4] L. Castrej, K. Kundu, R. Urtasun, and S. Fidler. 2017. Annotating Object Instances with a Polygon-RNN. In *CVPR.*

[5] S. Chen, T. Yao, and Y.-G. Jiang. 2019. Deep Learning for Video Captioning: A Review. In *IJCAI.*

[6] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and L. Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *CoRR* abs/1504.00325 (2015).

[7] M. Cornia, L. Baraldi, and R. Cucchiara. 2019. Show, Control and Tell: A Framework for Generating Controllable and Grounded Captions. In *CVPR.*

Table 4: Some qualitative results showing how our iCap system responses to user input in a specific annotation session. The cursor, marked out by ∧, indicates the position where the user wants to edit. Text completed by a specific model is highlighted in color. Different from the baseline model (Show-and-Tell) which considers only text at the left-hand side of the cursor, our *ABD-Cap* model exploits texts at both sides, and thus suggests sentences more close to final annotations. Texts in parentheses are English translations, provided for non-Chinese readers.

Test image	Texts in an iCap session
	***User input at a specific moment***: 一个∧ (A ∧) ***Top-1 sentence generated by a specific model***: Baseline ⇒ 一个戴眼镜的男人正在打电话 (A man with glasses is on the phone) *ABD-Cap* ⇒ 一个穿着西装的男人的黑白照片 (A man in a suit in a black-and-white photograph) ***Final annotation***: 一个穿着西装打着领带的男人坐在街道上的黑白照片 (A man in a suit sit on the street in a black-and-white photograph)
	***User input at a specific moment***: 一只狗∧ (A dog ∧) ***Top-1 sentence generated by a specific model***: Baseline ⇒ 一只狗在草地上奔跑 (A dog is running on the grass) *ABD-Cap* ⇒ 一只狗在草地上玩飞盘 (A dog is playing frisbee on the grass) ***Final annotation***: 一只狗站在绿色的草地上玩飞盘 (A dog is playing frisbee on the green grass)
	***User input at a specific moment***: 一只黑白相间的斑点狗和∧躺在沙发上 (A black and white spotted dog and ∧ lie on the sofa) ***Top-1 sentence generated by a specific model***: Baseline ⇒ 一只黑白相间的斑点狗和一只狗 (A black and white spotted dog and a dog) *ABD-Cap* ⇒ 一只黑白相间的斑点狗和一只棕色的狗趴在沙发上 (A black and white spotted dog and a brown dog lie on the sofa) ***Final annotation***: 一只黑白相间的斑点狗和一只棕色的狗趴在床上往回看向镜头 (A black and white spotted dog and a brown dog lie on the bed and look back at the camera)
	***User input at a specific moment***: 桌子上放着一个草莓∧披萨饼 (On the table sits a strawberry ∧ pizza) ***Top-1 sentence generated by a specific model***: Baseline ⇒ 桌子上放着一个草莓和一个比萨饼 (On the table sits a strawberry and a pizza) *ABD-Cap* ⇒ 桌子上放着一个草莓和奶酪的比萨饼 (On the table sits a strawberry and cheese pizza) ***Final annotation***: 桌子上放着一个草莓和奶酪的比萨饼 (On the table sits a strawberry and cheese pizza)

[8] H. Ge, Z. Yan, K. Zhang, M. Zhao, and L. Sun. 2019. Exploring Overall Contextual Information for Image Captioning in Human-Like Cognitive Style. In *ICCV*.

[9] S. Hochreiter and J. Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[10] L. Huang, W. Wang, J. Chen, and X.-Y. Wei. 2019. Attention on Attention for Image Captioning. In *ICCV*.

[11] W. Lan, X. Li, and J. Dong. 2017. Fluency-Guided Cross-Lingual Image Captioning. In *ACMMM*.

[12] V. I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady* 10 (Feb. 1966), 707.

[13] X. Li, L. Gao, X. Wang, W. Liu, X. Xu, H.-T. Shen, and J. Song. 2019. Learnable Aggregating Net with Diversity Learning for Video Question Answering. In *ACMMM*.

[14] X. Li, W. Lan, J. Dong, and H. Liu. 2016. Adding Chinese Captions to Images.

[15] X. Li, C. Xu, X. Wang, W. Lan, Z. Jia, G. Yang, and J. Xu. 2019. COCO-CN for Cross-Lingual Image Tagging, Captioning and Retrieval. *T-MM* (2019).

[16] H. Ling, J. Gao, A. Kar, W. Chen, and S. Fidler. 2019. Fast Interactive Object Annotation With Curve-GCN. In *CVPR*.

[17] T. Luong, H. Pham, and C. D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*.

[18] P. Mettes, D. Koelma, and C. Snoek. 2016. The ImageNet Shuffle: Reorganized Pre-training for Video Event Detection. In *ICMR*.

[19] S. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. 2017. Self-Critical Sequence Training for Image Captioning. In *CVPR*.

[20] D. Serdyuk, N. Rosemary Ke, A. Sordoni, A. Trischler, C. Pal, and Y. Bengio. 2018. Twin Networks: Matching the Future for Sequence Generation. In *ICLR*.

[21] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. 2015. Show and Tell: A Neural Image Caption Generator. In *CVPR*.

[22] C. Wang, H. Yang, and C. Meinel. 2018. Image Captioning with Deep Bidirectional LSTMs and Multi-Task Learning. *TOMM* (2018).

[23] A. Wu, Y. Han, and Y. Yang. 2019. Video Interactive Captioning with Human Prompts. In *IJCAI*.

[24] J. Xu, T. Mei, T. Yao, and Y. Rui. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *CVPR*.

[25] X. Zhang, J. Su, Y. Qin, Y. Liu, R. Ji, and H. Wang. 2018. Asynchronous Bidirectional Decoding for Neural Machine Translation. In *AAAI*.