Neurocomputing 333 (2019) 86–100

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Phrase-based image caption generator with hierarchical LSTM network

Ying Hua Tan, Chee Seng Chan*

Center of Image and Signal Processing, Department of Artificial Intelligence, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603 Malaysia

ARTICLE INFO

Article history:
Received 1 July 2018
Revised 19 October 2018
Accepted 4 December 2018
Available online 28 December 2018

Communicated by Dr XIANG Xiang Bai.

Keywords:
Image captioning
Natural language processing
Long short-term memory
Deep learning

ABSTRACT

Automatic generation of caption to describe the content of an image has been gaining a lot of research interests recently, where most of the existing works treat the image caption as pure sequential data. Natural language, however possess a temporal hierarchy structure, with complex dependencies between each subsequence. In this paper, we propose a phrase-based image captioning model using a hierarchical Long Short-Term Memory (phi-LSTM) architecture to generate image description. In contrast to the conventional solutions that generate caption in a pure sequential manner, phi-LSTM decodes image caption from phrase to sentence. It consists of a phrase decoder to decode the noun phrases of variable length, and an abbreviated sentence decoder to decode the abbreviated form of the image description. A complete image caption is formed by combining the generated phrases with sentence during the inference stage. Empirically, our proposed model shows a better or competitive result on the Flickr8k, Flickr30k and MS-COCO datasets in comparison to the state-of-the art models. We also show that our proposed model is able to generate more novel captions (not seen in the training data) which are richer in word contents in all these three datasets.

1. Introduction

Automatic caption or description generation from images is a challenging problem that requires a combination of visual and linguistic information. In other words, it requires not only complete image understanding, but also sophisticated natural language generation [1,2]. This is what makes it such an interesting task that has been embraced by both the computer vision and natural language processing communities.

Over the past few years, one of the most common frameworks applied in this line of research is a neural network model that composed of two sub-networks [3–7], where a convolutional neural network (CNN) is used to encode an image into a feature representation; while a recurrent neural network (RNN) is applied to decode it into a natural language description. In particular, the Long Short-Term Memory (LSTM) model [8] has emerged as the most popular RNN architecture, as it has the ability to capture long-term dependency and preserve sequence. Recently, many variants of this LSTM framework were introduced and achieved good results, such as those with attention mechanism [9–12] and attributes [13–15]. However, we notice that most of these works decode image caption in a fully sequential word-by-word basis. Although sequential model is appropriate for processing sentential data, it does not take other syntactic structure of language into consideration in its modeling.

In fact, natural language is a sequential data that has temporal hierarchy, with information spread out over multiple time-scales [16]. Consider English as an example, the lowest level with the shortest time-scale is characters, followed by words, phrases, clauses, sentences to documents. Therefore, it is undeniable that sentence structure is one of the prominent characteristics of language. Victor Yngve, an influential contributor in linguistic theory stated in 1960 that "*language structure involving, in some form or other, a phrase-structure hierarchy, or immediate constituent organization*"[17]. Hence, forcing a generative model to train on flat sequences and then generate a high-level structure locally, in a step-by-step basis often results in limited performance [18]. For image caption in particular, it can be observed that there are at least two levels of structure in those human annotated captions provided in the public datasets such as Flickr8k, Flickr30k and MS-COCO. Within each of the caption, there are several phrases that describe the objects in an image. These phrases have equal time-scale resolution at the word level, and they are conditioned on both the image and short-term language structure during decoding. Thus, previous words in the caption excluding the phrase itself, encoded in the long term memory is redundant in its generation process. Moreover, the structure of caption across these phrases is more inter-dependent, and so it requires both the image

* Corresponding author.
E-mail addresses: tanyinghua@siswa.um.edu.my (Y.H. Tan), cs.chan@um.edu.my (C.S. Chan).

https://doi.org/10.1016/j.neucom.2018.12.026
0925-2312/

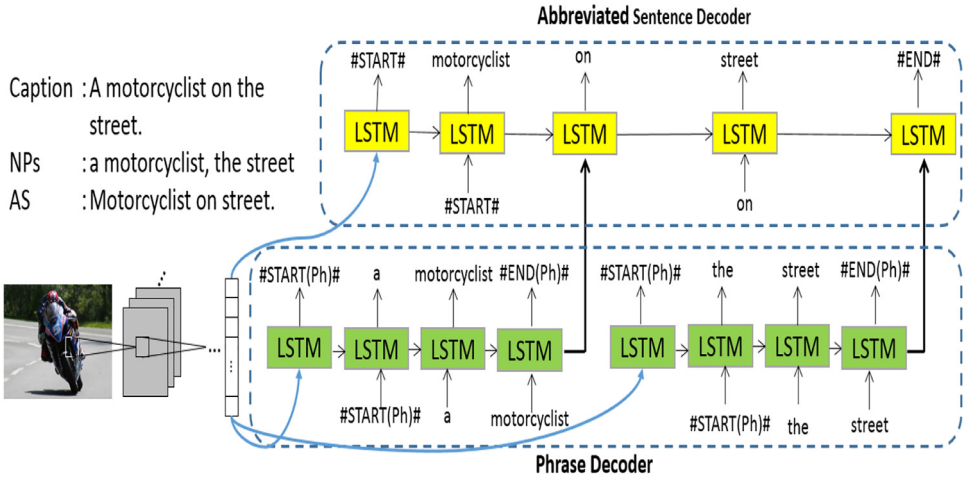


Fig. 1. The overall architecture of phi-LSTM model. It consists of a phrase decoder at the bottom hierarchy and an abbreviated sentence decoder at the upper hierarchy.

and all the previous sequences as a context to generate a correct description.

In this paper, we would like to investigate the capability of a phrase-based image captioning model that incorporates the observed structure in its modeling, as compared to a similar model trained on the flat sequences. To this end, we design a phrase-based image captioning model using a hierarchical LSTM architecture, namely **phi-LSTM** that consists of a phrase decoder and an abbreviated sentence (AS) decoder to generate image description from phrase to sentence. As illustrated in Fig. 1, given an image encoded with the CNN, the phrase decoder is first employed to decode the noun phrases (NPs) (i.e. *a motorcyclist, the street*) that describe the dominant entities within the image, using words as the atomic unit. At the same time, the phrase decoder also encodes each of the NP into a compositional vector representation, which will serve as an input to the AS decoder at the upper hierarchy. As such, the NPs will have an equal time step resolution as to the remaining words at the sentence level (i.e. *on*). Then, the AS decoder will decode an abbreviated form of the caption, which is made up from the last word of each NP (i.e. *motorcyclist, street*) and those remaining words that connect the phrases (i.e. *on*). Finally, a complete image caption (i.e. *A motorcyclist on the street*) is formed by combining the generated phrases with sentence gradually, during the beam search at the inference stage. Empirically, our proposed model shows a better or competitive results on Flickr8k [19], Flickr30k [20] and MS-COCO [21] datasets in comparison to the state-of-the art models.

As a summary, our contributions are two-folds:

1. We propose a novel phrase-based hierarchical LSTM model to decode image caption from phrase to sentence.
2. We show that the image caption generated with phi-LSTM is more accurate, novel (not seen in the training data), and richer in word content.

A preliminary version of this work was presented in [22], whereas the present work adds to the initial version in significant ways. First, the phrase selection objective is replaced with the prediction of the last word of each NP with the AS decoder for training simplicity. Secondly, we apply length normalization during the inference stage at both of the phrase and sentence levels, in order to generate a longer caption. Thirdly, we further improve the outputs of the parsing tool with a phrase refinement strategy. Finally, considerable new analysis and intuitive explanations are added to our results. We also extend our experiment to include the MS-COCO dataset [21], and evaluate our results on four additional evaluation metrics (i.e. METEOR [23], ROUGE [24], CIDEr [25] and SPICE [26]).

2. Related works

The image description generation approaches are differed in terms of i) how the context in which the description is derived from is represented, and ii) how a sentence is generated.

2.1. Context representation

To encode visual information, earlier works rely on multiple visual detectors and classifiers to capture different aspects of an image, such as objects, attributes, relations and scene [27–33]. The outputs of these detectors and classifiers usually form a set of tuples [27–31], in which the description is built upon. Such approach generally fixes the number of classes for each aspect of the image. Since the unprecedented success of CNN in image classification and object detection tasks, a growing number of works start to use different variants of CNN to encode a whole image [3,4,6,7,9,11,15,34–39], or multiple image regions [5,10,12,13,40–42]. Given the CNN encoded image and its description, many works train a multimodal embedding space using various language models [3–7,9,11,15,34,36–40,43] to decode image caption. Alternatively, Fang et al. [41], Wu et al. [13], Yao et al. [14], and You et al. [15] trained a set of "visual word detectors" on the training data to encode image into a semantic space, named as the attributes.

Besides that, there are works that rely on retrieval approach to generate image description. By retrieving and re-ranking the caption of similar images from the training sets [34,35,40,44,45], a query image can be described with human written caption that is most relevant to its content. However, this method is incapable of describing an image with unseen composition of objects correctly. Thus, some of the works in this line of approach retrieve a set of tuples [27] or text snippets [32,33,46] to form and re-rank novel captions. On the other hand, Mun et al [47] built an attention map on the feature of query image using the retrieved caption as guidance to form the context representation.

2.2. Description generation

Given various contexts described above, several approaches are developed to generate image description, which are i) template-based, ii) composition-based, and iii) language model-based.

2.2.1. Template-based

This approach generates sentence using a pre-defined template with open-slots to be filled with image entities [27,28,30,46]. It is mostly used by works that represent visual content as a set of tuples. Description generated this way is usually syntactically correct, but rigid and not flexible.

2.2.2. Composition method

This approach stitches up text snippets retrieved [32,33] or entities detected [29,31] to form an image description. It requires sophisticated pre-defined rules to decide the set of text snippets or entities to be used for generating a complete caption, their orders and the gluing words in between them. Description generated in such manner is broader and more expressive compared to the template-based approach, but is also computationally expensive at test time due to its nonparametric nature.

2.2.3. Language model-based

Most recent works jointly embed image and language into a multimodal embedding space with neural network based language model to generate image caption [3–7,36]. For instance, Kiros et al. [36] proposed a multimodal log-bilinear neural language model which is biased by image feature to decode image caption. Chen et al. [48] used RNN to build a dynamic visual representation of generated words to aid the next word prediction during caption generation. Mao et al. [3] and Karpathy & Li [5] used RNN to decode caption of varying length, while LSTM was implemented in [4,7,13,38,49] to decode image description from their respective context. The context for caption generation can be any of those described in Section 2.1, or a combination of several types. For example, Jia et al. [38] used both CNN encoded image and semantic embedding learned with normalized Canonical Correlation Analysis as inputs to their LSTM decoder. Moreover, Xu et al. [9], Fu et al. [12], Li et al. [10], and Yang et al. [11] incorporated attention mechanism with the LSTM decoder to attend to various parts of image during the caption generation process. On the other hand, You et al. [15] implemented attention mechanism over semantic space instead of multimodal space when generating image caption.

2.3. Relation to our work

Similarly, our model employs the LSTM to decode image caption using the CNN encoded image as context. However, instead of using tokenized words as the atomic unit to a pure sequential LSTM, we introduce a hierarchical LSTM structure to decode image description from phrase to sentence. Thus, the input of our model at sentence level is a sequence of combination of words and phrases.

In terms of extending the LSTM from sequential data to graph-structured data, our model is slightly similar to Graph-LSTM [50] for semantic object parsing. However, the Graph LSTM [50] model is used to update information of each graph node based on their neighboring nodes while keeping the structure of each graph topology. On the other hand, our model aims to construct graph-structured data (natural language description) from a number of unorganized nodes (NPs), where the graph topology is unknown during inference.

Also, our work is different from the phrase-based approaches that use retrieval of text snippets paired with template or composition method to generate caption [32,33,46], as we do not rely on retrieval. Other phrase-based approaches place more emphasis on phrase learning and use a simple language model to decode sentence. For example, Lebret et al. [37] and Ushiku et al. [39] extracted various types of phrase from image description. The former trained phrase relevancy with image with negative sampling, and decoded a sequence of phrases using a tri-gram language model conditioned on the chunking tag of each phrase. The latter proposed a subspace-embedding method for phrase learning and generated sentence from estimated phrases using a combinatorial optimization. Our work differs from them in terms of i) the type of phrase extracted, ii) phrase learning approach, and iii) sentence decoding method.

First, we only extract NPs with intuition of having each phrase equivalent to an entity within the image. Moreover, we train both of our phrase and AS decoder using the LSTM, which are linked hierarchically as shown in Fig. 1. Thus, our phrase representation is learned from the backpropagation of AS decoder at sentence level. Lastly, we generate a complete caption by decoding AS while progressively replace the inferred noun with generated phrases.

A recent work, Skeleton-Key [51] designed a course-to-fine image caption decoder consists of two submodels, where Skel-LSTM learns to generate skeleton sentence made up of original caption with each NP replaced with its last word, while Attr-LSTM learns to decode the NPs. Their work designed a top-down model, where a skeleton sentence is first generated, followed by decoding each of the skeletal word to form the attribute sub-sequences.

3. phi-LSTM architecture

The main idea of the proposed phi-LSTM is decoding the image caption from phrase to sentence. It consists of a phrase decoder and an abbreviated sentence decoder. Given an image-sentence pair in the training set, NPs that are equivalent to i) the entities within the image and ii) made up of at least two words, are first chunked from the sentence (S), using a phrase chunking algorithm described in Section 5. Then, an AS is formed by replacing each of the NP in the caption with the last word of the chunked phrase as shown in the example below:

S: *The man in the gray shirt and sandals is pulling the large tricycle.*
NPs: *the man, the gray shirt, the large tricycle*
AS: *Man in shirt and sandals is pulling tricycle.*

We decompose each of the caption in the training data into an AS-NPs pair, such that the AS and NPs are processed with two decoders that are linked hierarchically. This decomposition alters the sequence order in the human annotated caption, and thus we will have different ground truth sequence (GTS) during the training stage as compared to the conventional RNN models. To this end, the GTS of our phrase decoder is the NPs, while the GTS of our AS decoder is the AS.

3.1. Phrase decoder

The phrase decoder in this work has two roles, which are

i) to decode an image representation into multiple NPs that describe the entities in the image, and
ii) to encode each of the NPs into a compositional vector representation, which serves as an input to the AS decoder.

Given an image I, a CNN pre-trained on ImageNet is applied to encode an image into a D-dimensional image feature, which is then transformed into a K-dimensional vector with image embedding matrix, $\mathbf{W_{ip}} \in \mathbb{R}^{K\times D}$ and bias $\mathbf{b_{ip}} \in \mathbb{R}^{K}$. A LSTM model similar to [4] is used to decode it into each of the NPs.

To train an LSTM model to decode the ith NP of length L_i, the embedded image feature, followed by a start-word token $\mathbf{x_{sp}} \in \mathbb{R}^{K}$ indicates the translation process, and each word in the NP are input to a sequence of LSTM blocks in a step-by-step manner, as shown in Fig. 2. Hence, the phrase decoder inputs $\mathbf{x^i_{t_p}}$ at each time

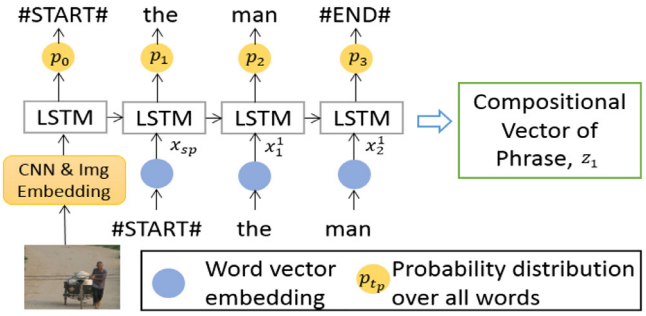


Fig. 2. The phrase decoder is trained to generate NPs and encode each NP into a compositional vector.

step of phrase, t_p are:

$$\mathbf{x}^{\mathbf{i}}_{\mathbf{t_p}} = \begin{cases} \mathbf{W_{ip}}\text{CNN}(I) + \mathbf{b_{ip}}, & \text{for } t_p = -1 \\ \mathbf{x_{sp}}, & \text{for } t_p = 0 \\ \mathbf{W_{ep}} w^i_{t_p}, & \text{for } t_p = 1...L_i, \end{cases} \tag{1}$$

where $\mathbf{W_{ep}} \in \mathbb{R}^{K\times V}$ is the trainable word embedding matrix of NPs, where each word in the vocabulary of size V is represented as a K-dimensional vector, and $w^i_{t_p}$ is a one-hot vector indicating the location of current input word in the vocabulary at time step t_p of phrase i.

For a LSTM block at time step t_p, let $\mathbf{i}_{t_p}, \mathbf{f}_{t_p}, \mathbf{o}_{t_p}, \mathbf{c}_{t_p}$ and $\mathbf{h}_{t_p}$ denote the input gate, forget gate, output gate, memory cell and hidden state at the time step. Thus, the LSTM transition equations omitting the phrase index i are:

$$\mathbf{i}_{t_p} = \sigma(\mathbf{W_i}\mathbf{x}_{t_p} + \mathbf{U_i}\mathbf{h}_{t_p-1} + \mathbf{b_i}), \tag{2}$$

$$\mathbf{f}_{t_p} = \sigma(\mathbf{W_f}\mathbf{x}_{t_p} + \mathbf{U_f}\mathbf{h}_{t_p-1} + \mathbf{b_f}), \tag{3}$$

$$\mathbf{o}_{t_p} = \sigma(\mathbf{W_o}\mathbf{x}_{t_p} + \mathbf{U_o}\mathbf{h}_{t_p-1} + \mathbf{b_o}), \tag{4}$$

$$\mathbf{u}_{t_p} = tanh(\mathbf{W_u}\mathbf{x}_{t_p} + \mathbf{U_u}\mathbf{h}_{t_p-1} + \mathbf{b_u}), \tag{5}$$

$$\mathbf{c}_{t_p} = \mathbf{i}_{t_p} \odot \mathbf{u}_{t_p} + \mathbf{f}_{t_p} \odot \mathbf{c}_{t_p-1}, \tag{6}$$

$$\mathbf{h}_{t_p} = \mathbf{o}_{t_p} \odot tanh(\mathbf{c}_{t_p}), \tag{7}$$

$$\mathbf{p}_{t_p+1} = \text{softmax}(\mathbf{h}_{t_p}). \tag{8}$$

Here, σ denotes the logistic sigmoid function while $\odot$ denotes elementwise multiplication. The LSTM parameters $\{\mathbf{W_i}, \mathbf{W_f}, \mathbf{W_o}, \mathbf{W_u}, \mathbf{U_i}, \mathbf{U_f}, \mathbf{U_o}, \mathbf{U_u}\}$ are all matrices with dimension of $\mathbb{R}^{K\times K}$. Intuitively, each gating unit controls the extent to which information is updated, forgotten and forward-propagated while the memory cell holds the unit internal memory regarding the information processed up to current time step. The hidden state is therefore a gated, partial view of the memory cell of the unit.

The output of the LSTM at each time step, $\mathbf{p}_{t_p+1} \in \mathbb{R}^V$ is equivalent to the conditional probability of a word given the previous words and image, $P(w_{t_p}|w_{1:t_p-1}, I)$. Its ground truth is the input word of next time step, and an end-word token at the last time step to indicate the end of a NP. The hidden state of the last time step is employed as the compositional vector representation of the NP, where

$$\mathbf{z}_i = \mathbf{h}_{\mathbf{L_i}}, \qquad \mathbf{z} \in \mathbb{R}^K. \tag{9}$$

This is served as the input to the AS decoder described next.

3.2. Abbreviated sentence(AS) decoder

The AS decoder has a similar design as the phrase decoder, except the inputs, outputs and GTS, as shown in Fig. 3. The input of the AS decoder is a complete caption describing the image, with each NP (e.g. *the man*) and the remaining words in the caption (e.g. *in*) are encoded as input in a single time step. Let t_s denotes the time step of the AS decoder and N is the length of the caption considering each NP as a unit, the input of AS decoder $\mathbf{y_{t_s}}$ is:

$$\mathbf{y_{t_s}} = \begin{cases} \mathbf{W_{is}}\text{CNN}(I) + \mathbf{b_{is}}, & \text{for } t_s = -1 \\ \mathbf{x_{ss}}, & \text{for } t_s = 0 \\ \left.\begin{array}{ll} \mathbf{W_{es}} w_{t_s}, & \text{if input is word} \\ \mathbf{z}_i, & \text{if input is phrase } i \end{array}\right\} & \text{for } t_s = 1...N. \end{cases} \tag{10}$$

The $\mathbf{W_{is}} \in \mathbb{R}^{K\times D}$, $\mathbf{b_{is}} \in \mathbb{R}^K$, $\mathbf{x_{ss}} \in \mathbb{R}^K$ and $\mathbf{W_{es}} \in \mathbb{R}^{K\times V}$ are another set of trainable parameters for image embedding, start-word token and word embedding matrix of AS, while w_{t_s} is the one-hot vector indicator of current input word of time step t_s.

Two outputs are produced by the LSTM model at each time step in the AS decoder, which are i) a binary indicator that determines if the next input is either a phrase or a word (i.e. phrase indication), and ii) a softmax prediction of the next word in the sequence of AS (i.e. word prediction).

The ground truth of the second output at each time step is either the last word of next phrase or the next word itself, formulated as:

$$GTS_{t_s} = \begin{cases} w_{t_s+1}, & \text{if next input is word} \\ w^i_{L_i}, & \text{if next input is phrasei} \\ \text{end-word token}, & \text{when } t_s = N. \end{cases} \tag{11}$$

In our preliminary work [22], we used a phrase token for the phrase indication, which resulted in a limitation of unable to discern on the appropriateness of different NP inputs during decoding. As a compensation, a phrase selection objective was introduced to solve this limitation. However, it has a complicated training procedure, because it is optimized over multiple randomly selected NPs input at each time step when the input is a NP. To simplify the training process, herein, we replace the phrase token and the phrase selection objective with phrase indication and softmax prediction of the last word of each NP (i.e. Eq. (11), if the next input is a phrase) respectively.

3.3. Training the phi-LSTM model

The objective function of our model is a log-likelihood cost function computed from the perplexity of word prediction summed with a loss from the phrase indication prediction. That is, given an image I and its description S, let R be the number of phrases of the sentence, while $\mathbf{p}_{\mathbf{t}p}$ and $\mathbf{p}_{\mathbf{t}s}$ are the probability output of LSTM block at time step $t_p - 1$ and $t_s - 1$ respectively. The perplexity of sentence S conditioned on an image I is

$$\log_2 \mathcal{PPL}(S|I) = -\frac{1}{M}\left[\sum_{t_s=1}^{N+1} \log_2 \mathbf{p}_{\mathbf{t}s} + \sum_{i=1}^{R}\left[\sum_{t_p=1}^{L_i+1} \log_2 \mathbf{p}_{\mathbf{t}p}\right]\right], \tag{12}$$

where $M = N + 1 + \sum_{i=1}^{R}(L_i + 1)$.

We use hinge loss as the phrase indication objective to classify the next input of the AS decoder into either phrase or word. The cost function of the classifier is

$$\mathcal{C}_{PI} = \sum_{t_s=1}^{N} \kappa_{t_s} \sigma(1 - y_{t_s}\mathbf{h_{t_s}}\mathbf{W_{ps}}), \tag{13}$$

where $\mathbf{h_{t_s}}$ is the hidden state output of the LSTM block at time step t_s, $\mathbf{W_{ps}} \in \mathbb{R}^K$ is trainable parameters for the classifier. y_{t_s} is +1 if the next input to the AS decoder is a phrase or -1 otherwise. Here, κ_{t_s} normalizes the objective based on the number of

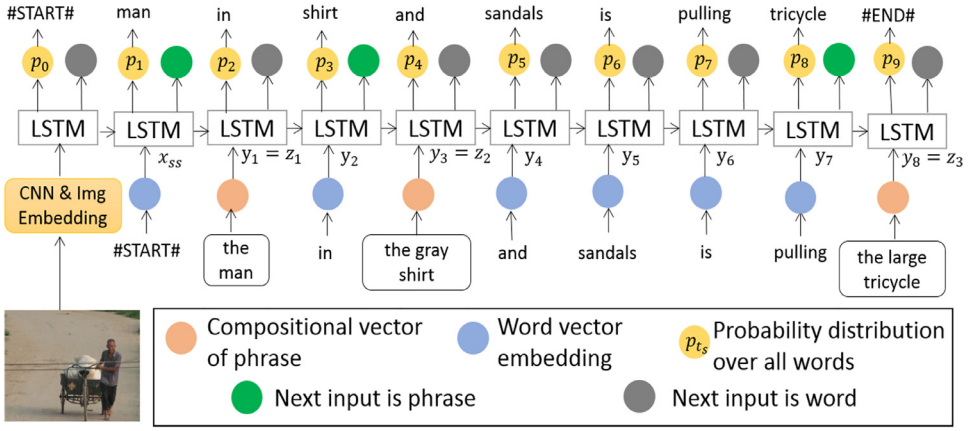


Fig. 3. Abbreviated sentence decoder: the input sequence is a complete caption, with each NP occupies only one time step, while the ground truth sequence is the abbreviated sentence of the caption. It also predicts whether the next input is either a phrase or a word.

phrases and words in the AS. Thus, $\kappa_{t_s} = 1/R$ if $y_{t_s} = 1$ or $1/(N-R)$ otherwise.

Hence, with P number of training samples, the overall objective function of our model is:

$$\mathcal{C}(\theta) = -\frac{1}{Q}\sum_{j=1}^{P}\left[M_j \log_2 \mathcal{PPL}(S_j|I_j) + \mathcal{C}_{PI}\right] + \lambda_\theta \cdot \| \theta \|_2^2, \tag{14}$$

where $Q = P \times \sum_{j=1}^{P} M_j$. It is equivalent to the average log-likelihood of a word given their previous context and the image described, summed with a regularization term, $\lambda_\theta \cdot \| \theta \|_2^2$, average over the number of training samples. Here, θ is all the trainable parameters of the model.

In summary, the proposed phi-LSTM architecture is optimized to predict i) the next word given all the previous words in each NP, ii) the next word of AS given all the previous words and phrases, and iii) if the next input is a phrase. This objective function allows the model to be trained end-to-end.

4. Image caption generation

The phi-LSTM model generates image caption in a two-steps manner, where a list of NP candidates are first generated followed by the complete caption, both using beam search algorithm. The beam size for phrase and sentence generation are b_p and b_s respectively.

Generation of the NPs in this work is similar to [4], where a given image encoded with the CNN followed by a start-word token are input to the model, acting as the initial context of the phrase decoder to generate NPs. At every time step, b_p words with the highest probability are sampled and input to the decoder at the next time step to infer the subsequent words. A set of b_p best sequences generated up to time step t_p are kept as potential candidates for inference of the next word iteratively, until all the candidates infer an end-word token. A score is then computed for each of the NP candidate by summing up the log probability of each word normalized by the length of NP, including the end-word token:

$$S_p = \frac{1}{L+1}\left[\sum_{t_p=1}^{L+1} \log_2 \mathbf{p}_{t_p}\right], \tag{15}$$

The generated NP candidates are then grouped according to their last word, and the candidates with score lower than the threshold value T will be discarded. This is in order to improve the quality of the image description formed. Nonetheless, at least one candidate (of the highest score) will be remained for each NP group regardless of its score. Following this, a total of b_s complete captions will be generated from the list of NP candidates, as illustrated in Fig. 4. The AS decoder produces two outputs at each time step, which are i) the next word prediction and ii) the phrase indication of next input. Thus, when the model infers that the next input is a phrase, each of the b_s word candidates inferred (e.g. *dogs, dog, a, two, brown* in Fig. 4) is compared with the list of NP candidates. Those NPs with the last word matches to the inferred words (e.g. *a brown dog, two dogs, two brown dogs*) are attached to the list of beam candidates at the current time step, replacing the inferred words (e.g. beam that infers '*dog*' will use NP '*a brown dog*' as next input instead). The inferred words without any NP alternative (e.g. *a, two, brown*) will remain in the list of beam candidates, for case where the phrase decoder does not generate an appropriate NP (e.g. single word noun or very small object). Once all candidate sentences infer an end-word token, the score of each caption is computed as:

$$S_s = -\log_2 \mathcal{PPL}(S|I), \tag{16}$$

and the sentence obtains the highest score, S_s is chosen.

5. Phrase chunking, limitations and refinement

Phrase chunking is a natural language process that separates and segments a sentence into its subconstituents, such as noun, verb, and prepositional phrases. A quick overview on the structure of image descriptions reveals that the key elements which compose the majority of the captions are usually those NPs that describe the dominant entities in an image. It can be either an object, group of objects or scene. These entities have equivalent abstract level as the output of a CNN encoder, and are linked with verb and prepositional phrase. Thus, NP essentially covers over half of the corpus in a language model trained to generate image description.[1] Therefore, in this paper, we partition the learning of the NP and sentence structure so that they can be processed more evenly, compared to extract all the phrases without considering their part of speech (POS) tag.

[1] This is confirmed by computing the number of words which are NPs and non-NPs using the phrase chunking algorithm described in Section 5.1 on MS-COCO dataset. We found out that there are over four millions of words which are NPs and over two millions of words which are not.

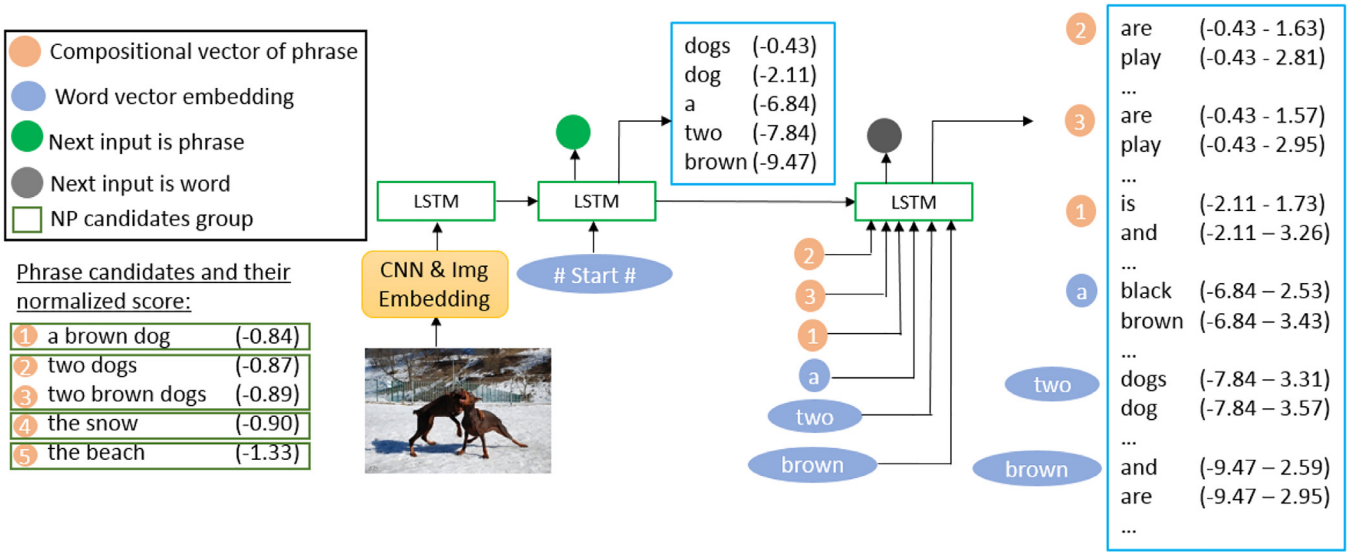


Fig. 4. Example of image caption generation given a set of generated NPs ($b_s = b_p = 5, T = -0.9$ in this example). Best viewed in color.

This section describes i) the parsing algorithm we applied to obtain the AS-NPs pair, ii) problems arose from the limitation of current parsing tool and our proposed solution, and iii) a measure we embark to reduce the influence of these limitations on the training of our image captioning model.

5.1. Phrase chunking

To identify NPs from a training caption, we adopt the dependency parse of Stanford CoreNLP tool [52], which forms a structural relation tree over a sentence by providing structural relationships between words. Though it does not chunk sentence directly as to the constituency parser and other chunking tools, the extracted pattern of the NP is more flexible as we can select the desirable structural relations. The relations we had selected are:

- determiner relation (*det*),
- numeric modifier (*nummod*),
- adjectival modifier (*amod*),
- compound (*compound*),
- adverbial modifier (*advmod*), only selected when the meaning of adjective term is modified, e.g. "*dimly lit room*",
- nominal modifier for 'of' & possessive alteration (*nmod:of* & *nmod:poss*), with case 'of' included.

In general, the dependency parser extracts several triplets, each made up of a governor word, a dependent word and a relation that links them, in the form of ⟨*relation (governor, dependent)*⟩, from a sentence. In order to form phrase chunks with the dependency parser, a simple post-processing step as illustrated in Fig. 5 is carried out. That is, triplets with the same governor or dependent word which are also consecutive in the complete caption (e.g. *amod(shirt, gray)* and *det(shirt, the)*) are grouped together as a single NP. The same applies for the consecutive triplet (e.g. *det(man, the)*), while the standalone word (e.g. 'in') remains as a unit in the AS.

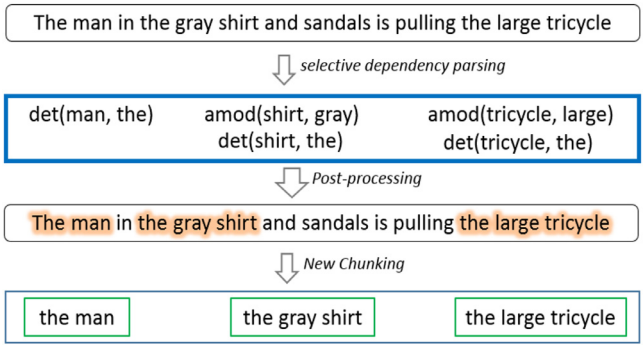


Fig. 5. An example of phrase chunking from the dependency parse.

5.2. Limitation of parsing tool

Due to the substantial ambiguity in linguistic structure, the parsing of natural language data is still an ongoing research with no perfect solution. As a result, there are always some unavoidable errors from the parser output, regardless of the chunking tool used. Asides from the dependency parser, we have tested phrase chunking with a constituency parser. The constituency parser outputs subject and predicate of a sentence directly, and we chunk the NP constituents at the lowest level. In this section, we will compare the AS-NPs pair formed by chunking using both of the parsers[2]. The NPs are showed in the left column while the AS is showed in the right column. The examples (a1-d1) given below are labelled with *S*, *DP*, *CP*, and *DP(R)*, which represent complete sentence, AS-NPs pair formed by chunking with dependency parser, constituency parser, and dependency parser with further refinement respectively. An underlined text indicates that the AS-NPs pair contains error.

One of the common errors found in the output of any of the parsers is incorrect recognition of a verb as a noun. As a result, AS with missing object is formed, as shown in the examples (a1 - d1, right column). Moreover, there are NPs that do not describe any entity in an image, such as '*the one*' in example (c1).

[2] Both parsers used throughout this work are in the package of Standford CoreNLP version 3.6.0. The type of dependency parser applied is collapsed-ccprocessed-dependencies.

(a1)	*S*:	A man in a blue shirt standing in a garden.	
	DP & CP:	*a man, a blue shirt standing, a garden*	Man in standing in garden.
(b1)	*S*:	A group of young people preparing to go skiing.	
	DP:	*a group of young people preparing*	Preparing to go skiing.
	CP:	*a group, young people preparing*	Group of preparing to go skiing.
(c1)	*S*:	Two men look toward the camera, while the one in front points his index finger.	
	DP & CP:	*two men, the camera, the one, front points, his index finger*	Men look toward camera, while one in points finger.
(d1)	*S*:	Two men and a woman on chairs outside near water.	
	DP & CP:	*two men, a woman, near water*	Men and woman on chairs outside water.

From our observation, both parsers seem to give relatively similar NP outputs. The reasons that we chose the dependency parser over the constituency parser are:

1. to chunk NPs with higher constituent level, it is more intuitive to select specific dependency relation such as '*nmod:of*', than specify the level of constituent NP in its parse tree.
2. there are some cases where a past tense verb is a part of the attributes of a noun, and the dependency parser has a higher chance to recognize it as adjective. For example:

(a2)	*S*:	Two snow covered benches sit in a snow covered field.	
	DP:	*two snow, a snow covered field*	Snow covered benches sit in field.
	CP:	*two snow, a snow*	Snow covered benches sit in snow covered field.
(b2)	*S*:	A red truck speeds down a tree lined street.	
	DP:	*a red truck, a tree lined street*	Truck speeds down street.
	CP:	*a red truck, a tree*	Truck speeds down tree lined street.

In order to parse NPs at a higher constituent level, we include the nominal modifier for 'of' and possessive alteration (*nmod:poss* & *nmod:of*) among our selected dependency relations. Most of the NPs chunked under these relations correspond to an entity or a group of entities within an image as we intended, as shown in example (a3). Nonetheless, there are still ambiguity for the NPs chunked from *nmod:of* relation, on either the whole phrase should be splitted into two NPs or remained as a single NP. Example (b3) shows the case where an '*of*' relation is not necessary, while example (c3) shows another case when the necessity of the relation is ambiguous.

(a3)	*S*:	A bird washes itself in a body of water.	
	DP:	*a bird, a body of water*	Bird washes itself in water.
	CP:	*a bird, a body*	Bird washes itself in body of water.
(b3)	*S*:	A lunch box is full of a variety of foods.	
	DP:	*a lunch box, full of a variety of foods*	Box is foods.
	CP:	*a lunch box, a variety of foods*	Box is full of foods.
(c3)	*S*:	A group of men and women walk down the center of a tree-lined street.	
	DP:	*a group of men and women, the center of a tree-lined street*	Women walk down street.
	CP:	*a group, the center, a tree-lined street*	Group of men and women walk down center of street.

5.3. Refinement of NPs

The limitations of parser have created unnecessary variations across the training data, which in turn has affected the training effectiveness of our image captioning model. In order to reduce the effects of the incorrect parsing on our model, we introduce a refinement strategy between the training of our phrase decoder and the AS decoder, where it will update the AS-NPs pair based on the local statistic of the training data. That is, the phrase decoder is first trained before the overall model with early stopping technique applied on the perplexity value of all NPs. Then, the model with the best validation set performance is used to generate a set of NPs from each of the training image. Next, the components of the AS-NPs pair of the training captions will be modified based on the generated NPs, by gradually restoring the non-inferred first word into its AS, followed by the non-inferred last word. The details of our proposed refinement algorithm are shown in Fig. 6, together with an example for better understanding. Given an image in the training data, a total of b_p NPs are generated. G_s and G_e are the set of first words and last words of all b_p generated NPs respectively, while K is a chunked NP (from parser) starts with word W_s and end with word W_e, with a length of $|K|$ words. This refinement is carried out for all of the chunked NPs in a sentence.

The examples below show the difference between the AS-NPs pair formed from our proposed phrase chunking approach described earlier, before and after the refinement strategy. Example (a4) shows where the RS1 comes into play, as none of the generated NPs start with word '*full*' but some start with word '*a*'. Example (b4) is corrected with RS2, as the word '*standing*' is not inferred as the last word of any of the generated NPs. In example (c4), phrases *the one*, *front points* and *his index finger* are restored to its AS, because our phrase decoder which uses image alone as its context will not generate NPs end with word '*one*', '*points*' and '*finger*'. These three phrases do not correspond to any dominant entities within the image, and thus it will seldom occur among the captions of similar images. In fact, '*the one*' cannot be generated from the image content alone, as it needs its subject ('*two men*') as the previous context. On the other hand, the word '*camera*' is inferred even though the object is not visible in the image due to the statistic of training data, as there are a lot of captions end with '*looking at the camera*' for images showing the frontal view of human. Example (d4) shows the case where our trained phrase decoder automatically decides which entity to be kept based on the statistic of the training data.

(a4)	*S*:	A lunch box is full of a variety of foods.	
	DP:	*a lunch box, full of a variety of foods*	Box is foods.
		a lunch box, a variety of foods	Box is full of foods.
	DP(R):		
(b4)	*S*:	A man in a blue shirt standing in a garden.	
	DP:	*a man, a blue shirt standing, a garden*	Man in standing in garden.
		a man, a blue shirt, a garden	Man in shirt standing in garden.
	DP(R):		
(c4)	*S*:	Two men look toward the camera, while the one in front points his index finger.	
	DP:	*two men, the camera, the one, front points, his index finger*	Men look toward camera, while one in points finger.
		two men, the camera	Men look toward camera, while the one in front points his index finger.
	DP(R):		
(d4)	*S*:	A group of men and women walk down the center of a tree-lined street.	
	DP:	*a group of men and women, the center of a tree-lined street*	Women walk down street.
		a group of men, the center of a tree-lined street	Men and women walk down street.
	DP(R):		

With this refinement strategy, the AS decoder will be trained fully on the refined AS, while the phrase decoder is fine-tuned on the refined NPs when the overall model is trained. On top of reducing the influence of error caused by the parser, we customize the AS-NPs pairs such that they are more appropriate for the image captioning task instead of linguistic task. Moreover, less dominant objects that need more informative prior context (long term memory) for its generation, such as *finger* in example (c4) will be learned by the AS decoder with the refinement applied.

Annotated Caption:
Two men look toward the camera, while the one in front points his index finger.
Chunked NPs:
two men, the camera, the one, front points, his index finger
Abbreviated Sentence (AS):
Men look toward camera, while one in points finger.

Generated NPs:
a man, a woman, the street, the camera, two men, an Asian man, a young man, a person, two people, the sidewalk, a sign, a store, a crowd, a city, a black jacket, a black shirt .....
List of generated first word, G_s:
two, the, a, an
List of generated last word, G_e:
men, man, shirt, people, jacket, person, street, woman, city, crowd, sidewalk, store, sign, camera

(a) Example of AS-NPs pair, generated NPs, G_s & G_e given an image.

Algorithm 1: NP refinement algorithm.

Data: K, W_s, W_e, G_s, G_e
Result: AS, K
/* Refinement step 1 (RS1) */
while $W_s \not\subset G_s$ **and** $|K| > 0$ **do**
  remove W_s from K and update W_s;
end
if $|K|=1$ **then**
  restore K completely into AS;
else if $|K|>1$ **then**
  restore the removed words from original K into AS.
end
/* Refinement step 2 (RS2) */
while $W_e \not\subset G_e$ **and** $|K| > 0$ **do**
  remove W_e from K and update W_e;
end
if $|K|<2$ **then**
  restore K completely into AS;
else
  restore the removed words from original K into AS.
end

(b) Algorithm

Initial:
Let K = '*the one*', W_s = '*the*', W_e = '*one*', $|K| = 2$
During RS1:
$W_s \in G_s$ → skip while loop → removed words = ∅
During RS2:
$W_e \notin G_e$ & $|K| > 2$
Iteration 1 of while loop :
K = '*the*', W_e = '*the*', removed words = '*one*', $|K| = 1$
Iteration 2 of while loop :
K = [ ], W_e = [ ], removed words = '*the one*', $|K| = 0$
$|K| < 2$ → '*the one*' is restored to AS completely
AS = *Men look toward camera, while the one in points finger.*

(c) Pseudo-code example

Fig. 6. NP refinement strategy.

6. Experiment

6.1. Datasets

The proposed phi-LSTM model is tested on three benchmark datasets - Flickr8k [19], Flickr30k [20], and MS-COCO [21]. These datasets consist of 8000, 31000 and 123,287 images respectively, each annotated with at least five image descriptions prepared by human from crowd sourcing. We follow the publicly available dataset splits[3] used in [5]. That is, the validation and testing set each contains 1000 images for Flickr8k & Flickr30k datasets, and 5000 images for MS-COCO dataset. The rest of the images are used for training.

6.2. Evaluation metrics

We employ five automatic metrics, including BiLingual Evaluation Understudy (BLEU) [53], Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [24], Metric for Evaluation of Translation with Explicit ORdering (METEOR) [23], Consensus-based Image Description Evaluation (CIDEr) [25] and Semantic Propositional Image Caption Evaluation (SPICE) [26] to evaluate the quality of the generated image captions. BLEU metric measures the precision of n-grams matching between a generated caption and all reference sentences, while ROUGE metric measures the recall instead of precision. Here, we only reported ROUGE-L which uses the longest common sequence instead of n-grams. METEOR aligns generated caption and reference string by mapping each unigram using three different modules, which are "exact", "porter stem" and "WordNet synonymy" modules. The final score is the F-mean computed from the number of unigram mapping. CIDEr metric combines the average cosine similarity of each n-gram between the generated caption and references. It gives lower weight to n-grams that commonly occur across all reference captions in the dataset. Lastly, SPICE metric parses image caption and its references into a scene graph to form tuples for each semantic proposition. Then, it computes the F-score defined over the conjunction of all logical tuples.

6.3. Experimental details

Aside from our proposed phi-LSTM model, we have conducted experiment on a baseline model which processes image caption as a sequence of words. It is basically a reimplementation of work described in [4], but without ensemble multiple trained models and using VGGnet [54] instead of GoogleLeNet [55] to encode image for a fair comparison with our model. All experimental settings in the baseline model and ours are the same unless stated otherwise.

During the training stage, we use raw caption without any preprocessing as input to the language parser in order to get a more appropriate AS-NPs pair. Then, all the words in the AS-NPs pair are converted to lower case, with punctuations removed, and word that occurs less than 5 times in the training data discarded, so that the tokenization of our image captions are consistent with that of [5]. To avoid gradient explosion due to overlength caption (relative to average length of all training data), we truncate the sentence as specified in Table 1. For the overlength NPs, we truncate the first few words instead of last few words, because the latter part of NPs usually holds more significant semantic content. The length of the AS-NPs pair considered are those after the refinement described in Section 5.3. The truncate length is decided such that the number of captions affected are less than 0.5% of the whole training data.

The CNN encoder used in this paper is the VGG-16 [54] pretrained on ImageNet, but without fine-tuning the CNN parameters, for a fair comparison with the preliminary version of this work [22]. We also include the quantitative results obtained by using the *pool5* feature of ResNet-152 [56] as the image encoder for references. The LSTM decoder with hidden size of K=256 (Flickr8k) and K=512 (Flickr30k & MS-COCO) is employed. Our model is

[3] http://cs.stanford.edu/people/karpathy/deepimagesent/.

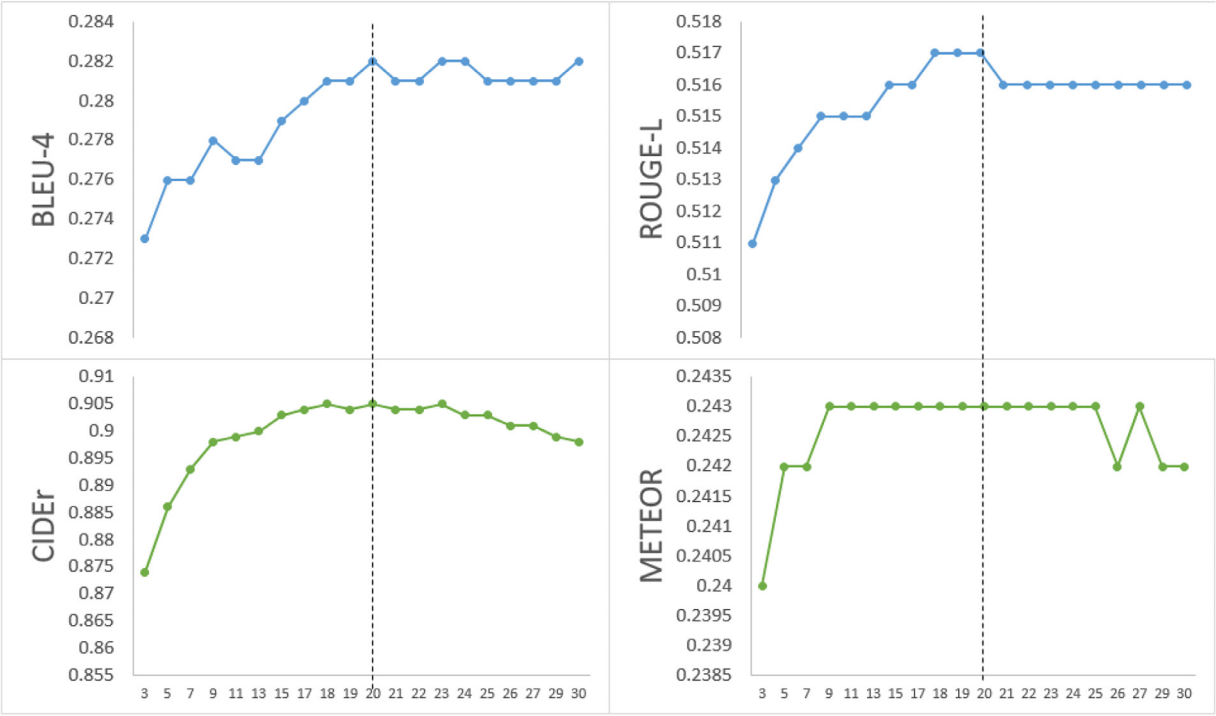


Fig. 7. Effect of the AS decoder's beam size, b_s on different metrics in MS-COCO dataset, using ResNet-152 as the image encoder. b_s=20 is the optimum value according to the findings.

Table 1
Caption truncation setting.

Dataset	Model	Truncate length	Captions affected (%)
Flickr8k	Baseline	24	0.25
	phi-LSTM (AS)	20	0.24
	phi-LSTM (NP)	7	0.12
Flickr30k	Baseline	36	0.25
	phi-LSTM (AS)	30	0.29
	phi-LSTM (NP)	7	0.12
MS-COCO	Baseline	23	0.26
	phi-LSTM (AS)	18	0.35
	phi-LSTM (NP)	7	0.36

optimized with RMSprop, using a minibatch of 300(Flickr8k), 500(Flickr30k) and 700(MS-COCO) image-sentence pair per iteration. The learning rate is set to 0.001, and dropout regularization is employed to avoid overfitting.

During the testing stage, we found that our proposed model generates better caption with large beam size, as shown in Fig. 7, while the baseline model's performance drops when large beam size is used. According to Vinyals et al., a well trained model should yield better result with larger beam size, and getting best performance with a relatively small beam size is an indication of model overfitting [57]. Nevertheless, we compare our model using beam size of b_p=30 and b_s=20, with the baseline model tested with beam size of b=3 and b=20.

Our model generates caption in a two-stage manner, from NP to complete caption as described in Section 4. We show some examples of the generated NPs in Fig. 8. To choose an appropriate value of threshold T, we examine the changes of several metrics and sentence uniqueness on the generated captions using a varying value in each dataset. The test result of MS-COCO dataset is shown in Fig. 9. It is observed that all the n-grams metrics (BLEU, CIDEr, METEOR and ROUGE-L) gradually increase with the threshold, and reach an optimum at T=-1.6 for Flickr8k and Flickr30k datasets, and T=-1.5 for MS-COCO dataset. Further increment of T yields different effect on different n-grams metrics, where BLEU and CIDEr decrease while METEOR and ROUGE-L fluctuate irregularly. Besides that, the sentence uniqueness constantly reduces with the increment of T as a result of less choice of NP candidates. We also notice that there are not much changes in the SPICE metric, where the score fluctuates within the range of 0.163–0.165 across varying value of T. This shows that the threshold value T only affects words' order and does not help much in predicting the correct objects, attributes and relations.

6.4. Comparison with state-of-the-art models

Tables 2,3,4 show the performance of our model in comparison with the state-of-the-art models, while Table 5 reports the performance of our model compared with the baseline model evaluated with MS-COCO online test server. B-n, MT, RG, CD and SP stands for n-gram BLEU, METEOR, ROUGE-L, CIDEr and SPICE respectively. † indicates that the results is obtained by ensembling multiple trained models, while (w.r) and (w.o.r) refer to with and without phrase refinement respectively. (RN) is our complete model trained with ResNet-152 as image encoder. Refer to[4] for ⋆.

When compared with the methods that only use the CNN as encoder, our model performs better or comparable to all other state-of-the-art models, including the phrase-based models proposed by Lebret et al. [37] and Ushiku et al. [39]. Note that our current model has a lower BLEU-1 and BLEU-2 score but a higher BLEU-3 and BLEU-4 score compared to our preliminary results published in [22]. This is because lower order of the BLEU

[4] The score reported here is cited from [9], in which the authors claimed that they obtained the missing metrics from authors of [4] through personnel communications.

Table 2
Performance of phi-LSTM and other state-of-the-art methods on Flickr8k dataset.

Models	B–1	B–2	B–3	B–4	MT	RG	CD	SP
DeepVS [5]	57.9	38.3	24.5	16.0	16.7	–	0.318	–
NIC [4]†⋆	63.-	41.-	27.-	–	–	–	–	–
Baseline, $b = 3$	57.6	39.2	26.1	17.5	19.1	43.6	0.422	0.128
Baseline, $b = 20$	56.2	38.0	25.3	16.7	19.0	43.2	0.410	0.129
phi-LSTM [22]	**63.6**	43.6	27.6	16.6	–	–	–	–
phi-LSTM (w.o.r)	61.5	43.1	29.6	19.7	19.9	44.5	0.502	0.140
phi-LSTM (w.r)	62.7	**44.4**	**30.7**	**20.8**	**20.2**	**45.4**	**0.516**	**0.141**
phi-LSTM (RN)	*65.2*	*46.7*	*32.8*	*22.5*	*20.8*	*47.1*	*0.567*	*0.146*
With attention mechanism								
Soft-Atten [9]	67.0	44.8	29.9	19.5	18.9	–	–	–
Hard-Atten [9]	67.0	45.7	31.4	21.3	20.3	–	–	–
With extra information / extra information+attention mechanism								
g-LSTM [38]	64.7	45.9	31.8	21.6	20.2	–	–	–
ACVT [13]	74.-	54.-	38.-	27.-	–	–	–	–
Reg-Atten[12]†	63.9	45.9	31.9	21.7	20.4	47.0	0.538	–

Table 3
Performance of phi-LSTM and other state-of-the-art methods on Flickr30k dataset.

Models	B-1	B-2	B-3	B-4	MT	RG	CD	SP
mRNN [3]	60.-	41.-	28.-	**19.-**	–	–	–	–
DeepVS [5]	57.3	36.9	24.0	15.7	15.3	–	0.247	–
LRCNN [7]	58.7	39.1	25.1	16.5	–	–	–	–
NIC [4]†⋆	66.3	42.3	27.7	18.3	–	–	–	–
PbIC [37]	59.-	35.-	20.-	12.-	-	-	-	-
Baseline, $b = 3$	57.0	38.5	25.9	17.3	17.3	41.2	0.333	0.117
Baseline, $b = 20$	57.0	38.3	25.7	17.3	17.8	41.7	0.349	0.122
phi-LSTM [22]	**66.6**	**45.8**	28.2	17.0	–	–	–	–
phi-LSTM (w.o.r)	60.6	41.2	27.8	18.6	18.1	41.8	0.394	0.123
phi-LSTM (w.r)	61.5	42.1	**28.6**	**19.3**	**18.2**	**42.4**	**0.399**	**0.125**
phi-LSTM (RN)	*64.2*	*45.4*	*31.7*	*21.8*	*19.0*	*44.6*	*0.452*	*0.134*
With attention mechanism								
Soft-Atten [9]	66.7	43.4	28.8	19.1	18.5	–	–	–
Hard-Atten [9]	66.9	43.9	29.6	19.9	18.5	–	–	–
With extra information / extra information+attention mechanism								
g-LSTM [38]	64.6	44.6	30.5	20.6	17.9	–	–	–
ACVT [13]	73.-	55.-	40.-	28.-	–	–	–	–
Reg-Atten [12]†	64.9	46.2	32.4	22.4	19.4	45.1	0.472	–
Sem-Atten [15]†	64.7	46.0	32.4	23.0	18.9	–	–	–

Table 4
Performance of phi-LSTM and other state-of-the-art methods on MS-COCO dataset.

Models	B-1	B-2	B-3	B-4	MT	RG	CD	SP
mRNN [3]	67.-	48.-	35.-	25.-	–	–	–	–
DeepVS [5]	62.5	45.0	32.1	23.0	19.5	–	0.660	–
LRCNN [7]	66.9	48.9	34.9	24.9	–	–	–	–
NIC [4]†⋆	66.6	46.1	32.9	24.6	–	–	–	–
PbIC [37]	**70.-**	46.-	30.-	20.-	–	–	–	–
CoSMos [39]	65.-	**49.-**	32.-	20.-	20.-	–	–	–
Baseline, $b = 3$	65.2	47.5	34.3	25.2	22.6	49.3	0.779	0.154
Baseline, $b = 20$	61.7	43.7	31.4	23.1	22.4	47.7	0.724	0.150
phi-LSTM (w.o.r)	66.0	48.2	34.7	25.0	23.0	49.4	0.812	0.165
phi-LSTM (w.r)	66.6	48.9	**35.5**	**25.8**	**23.1**	**49.7**	**0.821**	**0.165**
phi-LSTM (RN)	*69.5*	*52.3*	*38.5*	*28.2*	*24.3*	*51.7*	*0.905*	*0.175*
With attention mechanism								
Soft-Atten [9]	70.7	49.2	34.4	24.3	23.9	–	–	–
Hard-Atten [9]	71.8	50.4	35.7	25.0	23.0	–	–	–
Review [11]	–	–	29.-	23.7	–	0.88	–	–
Skel-Key [51]	74.2	57.7	44.0	33.6	26.8	55.2	1.073	0.196
With extra information / extra information+attention mechanism								
g-LSTM [38]	67.0	49.1	35.8	26.4	22.7	–	0.813	–
ACVT [13]	74.-	56.-	42.-	31.-	26.-	–	0.94-	–
Reg-Atten [12]†	72.4	55.5	41.8	31.3	24.8	53.2	0.955	–
Sem-Atten [15]†	70.9	53.7	40.2	30.4	24.3	–	–	–

Fig. 8. Examples of NPs generated from image. Red fonts indicate that the NP's score S_p is lower than threshold T. Complete caption generated from the NP candidates are shown at the bottom of each image.

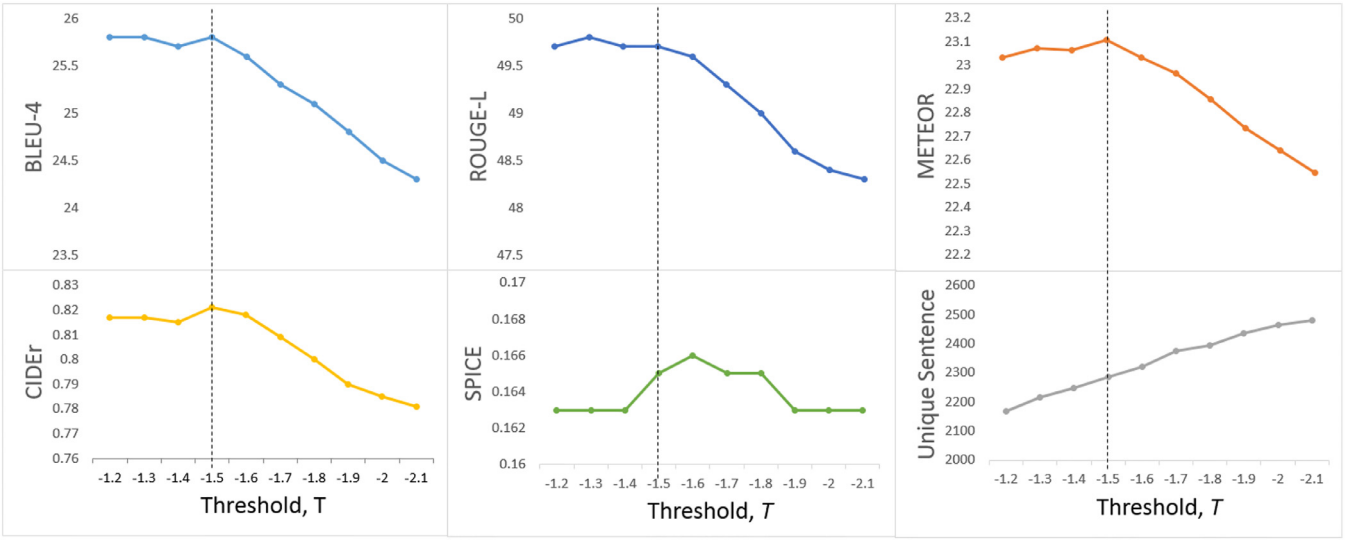


Fig. 9. Effect of threshold T on five different metrics and number of unique captions generated in MS-COCO dataset.

Table 5
Performance of phi-LSTM and baseline model, both using ResNet-152 as image encoder, evaluated on MS-COCO online test server.

Models	B-1		B-2		B-3		B-4		MT		RG		CD	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Baseline	66.6	85.4	49.0	74.3	35.6	62.1	26.1	50.8	23.7	31.9	50.3	64.4	0.818	0.826
phi-LSTM	**69.3**	**87.8**	**51.9**	**77.8**	**37.9**	**65.8**	**27.6**	**53.8**	**24.1**	**32.4**	**51.2**	**65.6**	**0.875**	**0.896**

metrics is bias towards short sentence [38],[5] but we have added length normalization in our beam search algorithm (Eqs. (15) and (16)) to generate longer caption. As a result, we are able to increase the average length of generated caption by approximately three words, e.g. from 6.8 words (as reported in Table 2 of [22]) to 9.72 words for Flickr8k dataset. Longer caption is desired for a better comparison with other models. Tables 2–4 also show the effectiveness of the NP refinement algorithm, as there is approximately 1 BLEU score improvement in all the three datasets when the refinement strategy is employed. Although we have reported the results obtained with ResNet-152 as image encoder for future reference, the VGG-16 results are still a fairer comparison with most of the works in the table.

Since the objective of our work is to investigate the capability of a phrase-based image captioning model, compared to a similar model trained on flat sequences, we do not implement attention mechanism or provide extra information to our model, as it is beyond the scope of this paper. Nevertheless, we argue that our model is comparable to the soft-attention model [9], which requires extra computation of relative importance of each location in feature maps at every time step.

7. Analysis of phi-LSTM model in comparison to its sequence model counterpart

7.1. SPICE metric evaluation

From the evaluation of SPICE metrics shown in Table 6, we observe that there are improvements in all the sub-metrics when image caption is decoded in the phrase-based hierarchical manner. Most of the improvements gained are at the object level (object, attribute, size and color). This is because we have essentially broken down the generation process of subsequences from global sequence with our proposed model. Therefore, the phrase decoder does not need to shift the time-scale of generative process repeatedly, and can focuses on a particular aspect of image when generating the NPs similar to the attention mechanism [9]. The difference is that the model with attention mechanism decodes a

[5] This happens when the brevity penalty (BP) of BLEU is set to 1 (i.e. without BP), which is the default setting of publicly available code for evaluation in https://github.com/karpathy/neuraltalk and https://github.com/tylin/coco-caption. Since the BP value is seldom reported, we assume that this is the setting others used.

Table 6
Performance of phi-LSTM (w.r.) and baseline model evaluated with SPICE measurements on MS-COCO dataset, with beam size = 20.

Models	SPICE	Precision	Recall	Object	Relation	Attribute	Size	Color	Cardinality
Baseline	0.150	0.386	0.095	0.284	0.033	0.064	0.023	0.070	0.000
phi-LSTM (w.r.)	**0.165**	**0.449**	**0.104**	**0.310**	**0.038**	**0.076**	**0.036**	**0.100**	**0.002**

Table 7
Measure of caption uniqueness and novelty. A higher 'seen' percentage indicates that the generated captions are less novel. The number of unique words of all captions is shown under 'Words', where 'Within vocab.' considers only words that are in the training corpus.

Models	Sentence			Words	
	Unique	Seen	Avg. length	Actual	Within vocab.
Flickr8k					
References	99.84%	1.20%	10.87	3147	1919
Baseline ($b = 3$)	58.70%	10.80%	11.06	–	196
Baseline ($b = 20$)	54.40%	12.20%	11.54	–	201
phi-LSTM (w.r.)	67.70%	7.40%	9.72	–	212
Flickr30k					
References	99.96%	0.30%	12.39	4204	3561
Baseline ($b = 3$)	65.70%	10.70%	12.40	–	348
Baseline ($b = 20$)	58.90%	9.40%	12.81	–	328
phi-LSTM (w.r.)	77.20%	9.30%	11.07	–	375
MS-COCO					
References	99.22%	5.56%	10.44	7241	5949
Baseline ($b = 3$)	38.06%	63.54%	10.12	–	517
Baseline ($b = 20$)	24.54%	77.32%	10.60	–	457
phi-LSTM (w.r.)	46.42%	48.54%	9.81	–	548

sequence that spreads out over multiple time-scales while ours fixes the time-scale of subsequence decoder at the object level. Nonetheless, it has a global sequence of mixed time-scales, as non-object phrases are decoded in multiple time steps at the sentence level. There are only small improvement in terms of relations and cardinality, because the CNN encoder we used does not hold any information regarding the relative position of the objects. Therefore, object relations are mostly inferred from the local statistic of training data (e.g. image with human and bicycle always has ärides' relation inferred because this relation occurs most in training data). On the other hand, cardinality which measure the correctness in terms of object counting has the lowest score because the CNN encoder applied is never trained for object counting.

7.2. Evaluation on uniqueness and novelty of caption

It has been pointed out that multimodal RNN-based approach tends to reconstruct previously seen caption [35]. Hence, we compare our model with baseline in terms of the uniqueness (ratio of non-repeated generated caption in the test set) and novelty (ratio of generated sentence that is different from the training captions), similar to [35]. We compute and tabulate i) the percentage of unique captions generated, ii) the percentage of generated captions that are seen in the training data, iii) the average length of the captions, and iv) the number of unique words generated in Table 7. To obtain an upper bound of performance under these measures, we evaluate the five human annotated captions of the same set of test images as reference.

From Table 7, we observe that our model can generate more unique and novel (not seen in training data) captions, when compared with the baseline in all three datasets. Although the average length of our captions is shorter than the baseline, it is only about one word less when compared with the human annotated captions. In our experiment, the vocabulary size of Flickr8k, Flickr30k and MS-COCO datasets are 2538, 7413 and 9996 words, respectively. Therefore, there are a total of 1228, 643 and 1292 out-of-vocabulary words in the test set of the three datasets respectively, which would penalize all the automatic metrics we used. Assume that all within-vocabulary words in the reference captions are the upper bound of test image relevant words a well-trained image captioning model can infer, we observe that both our model and baseline can only generate captions that made up of around 10% of all possible words. Nevertheless, the number of unique words generated using our model is still higher than the baseline which has a longer average caption length.

We deduce that the uniqueness and novelty of caption generated with our model is gained from the changes of probability distribution of words during inference with the AS decoder. In the baseline model, there are a lot of times where the model obtains high probability score from predicting the determiner word such as '*a*' and '*the*', especially during the first word prediction. Due to the cumulative nature of probability score when using the beam search algorithm, such prediction tends to remain as high-rank beam candidate, compared to prediction of other words. Therefore, the baseline model tends to generate caption starts with word '*a*'. On the other hand, our model decodes the NPs separately, and thus the AS seldom contains determiner words. This boosted the probability of other words to be inferred and remain in high-rank among all beam candidates during the inference with AS decoder. As a result, our model get less influence from the determiner words and this improves the uniqueness and novelty of our generated captions.

7.3. Model limitations observed with qualitative analysis

To gain further insights on how the number of occurrence of each word in the training corpus affects the word prediction when generating caption, we record the top five, *least seen* words that are inferred by both models in Table 8. Then, we examine manually the captions that contain those words, and highlight the words that are used correctly in describing their respective image. The image-caption pair of some correctly inferred least seen words are shown in Fig. 10 as examples. From Table 8, we can see that our phrase-based model is generally able to infer correctly more words which are less seen, compared to the baseline. The only exception is in Flickr8k dataset, where the baseline manage to infer correctly the word '*snowboarding*' which is seen for only 44 times. The corresponding caption is shown in the first image in Fig. 10, and we found that our model has inferred '*snowboarder*' for that image, which naturally makes the generation of action '*snowboarding*' redundant.

Furthermore, we record the top five, *most seen* words which are absent in the generated captions of our model and the baseline in Table 9. From the table, we observe that the words in which our model is able to infer while the baseline cannot are: '*an*' (Flickr8k & Flickr30k), '*green*' (Flickr8k), '*one*' (Flickr30k), '*there*' and '*three*' (MS-COCO). In the case of word '*an*', it is because the test set of Flickr dataset contains more attributes starting with vowels compared to objects, such as '*an orange shirt*' and '*an outdoor market*' as shown in Fig. 11. Such sequence usually has a low score until the object word is predicted (i.e. the sequence score of '*an outdoor*' is much lower than '*a market*' before the third word '*market*' is predicted). Due to the re-rank and drop out procedure of beam search

phi-LSTMv2: A snowboarder is jumping in the air.

Baseline, b=3: A man in a red jacket is snowboarding on a snowy hill.

Goundtruth: A lone snowboarder jumping in the air doing a trick.

A red car is driving through mud.

A red car is driving through a forest.

A black mitsubishi is driving through a muddy puddle and spraying water.

A man is doing a trick on a ramp.

A man is doing a trick on a bicycle.

A skateboarder in the air at a big outdoor ramp.

A black and white dog jumps to catch a frisbee.
A black and white dog is running through the grass.
Two dogs play catch with a ball in a field.

phi-LSTMv2: A football player in a red and white uniform is being tackled by another player in a white uniform.

Baseline, b=3: A man in a red and white uniform is holding a football.

Goundtruth: Young defensive football players executing a tackle on a member of the opposite team during a football game.

A man in a white shirt is standing in front of a chalkboard.

A man in a white shirt is sitting at a table in a restaurant.

A man wearing a baseball hat and jacket reads while sitting in an italian restaurant.

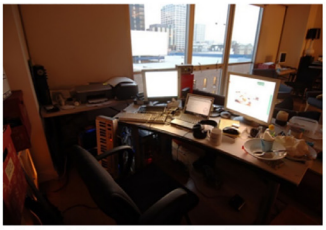
A man and a woman are looking at produce at a market.

A man in a white shirt and a woman in a white shirt are standing in front of a cart full of fruits and vegetables.
An employee is handing a woman a bag while she is browsing through fish on ice at a street market.

A man is jumping into the air.

A man in a black shirt is jumping on a sand dune.

A man in a long sleeved gray shirt and jeans leaps from a sandy hillside.



phi-LSTMv2: A desk with a laptop and a printer on it.

Baseline, b=3: A desk with a laptop and a computer on it.

Goundtruth: A desk holds a computer and leftover food containers.

A very tall clock tower towering over the middle of a city.
A large building with a clock on the top of it.
A large clock tower on top of a church next to a road.

A group of people riding elephants on a river.
A group of people riding on the backs of elephants.
People rafting on a river while others ride on top of elephants.

A woman taking a selfie in a bathroom mirror.
A woman taking a picture of herself in the mirror.
A woman is taking a selfie in her bathroom mirror.

Fig. 10. Examples of caption generated in Flickr8k (1st row), Flickr30k (2nd row) and MS-COCO (3rd row) datasets. The least seen words that are used correctly in the description are in green. More qualitative results are provided in the supplementary materials.

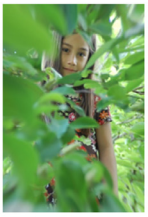
phi-LSTMv2: A group of three brown cows standing next to each other.

Baseline, b=3: A herd of sheep standing on top of a dirt field.

Goundtruth: Horse figurines next to walk on a table.

A tennis player in an orange shirt is holding a tennis ball.
A man in a blue shirt is playing tennis.
An african american woman dressed in orange is hitting a tennis ball with a racquet.

A group of people are standing at an outdoor market.
A group of people are sitting on a bench in front of a large building.
The woman in the red pant suit and white hat is selling her produce.



A young girl in a green shirt is sitting on a tree.
A man in a blue shirt is sitting on a rock in the
A girl in a flowered dress peaking through some leaves.

Fig. 11. Examples of the caption generated from different images. The most seen words that are inferred by our model, but not the baseline, are in green.

Table 8
Top-5 least seen words that are inferred in the generated captions. Those highlighted words means they have been inferred correctly in describing the image content.

Flickr8k				Flickr30k				MS-COCO			
phi-LSTM		Baseline		phi-LSTM		Baseline		phi-LSTM		Baseline	
Words	Seen	Words	Seen	Words	Seen	Words	Seen	Words	Seen	Words	Seen
bubble	34	stage	39	**tackled**	48	tablecloth	40	clearly	70	headboard	117
kayak	54	log	42	cows	49	tackled	48	**unripe**	94	drivers	183
driving	55	**snowboarding**	44	**chalkboard**	52	**dune**	82	**printer**	123	**racquets**	184
tent	57	hind	44	**tackle**	86	**formations**	82	hangar	134	**backs**	219
book	61	**kayak**	54	handstand	91	fruits	88	**towering**	176	**herself**	237

Table 9
Top-5 most seen words that are not inferred in the generated captions.

Flickr8k				Flickr30k				MS-COCO			
phi-LSTM		Baseline		Oursphi-LSTM		Baseline		Oursphi-LSTM		Baseline	
Words	Seen	Words	Seen	Words	Seen	Words	Seen	Words	Seen	Words	Seen
while	1443	an	1807	up	4762	an	14,590	by	16,378	by	16,378
child	1120	while	1443	as	4598	one	5890	several	9082	there	12,109
three	1052	child	1120	outside	4273	as	4598	sits	8847	three	10,612
one	876	three	1052	from	3721	outside	4273	area	8377	several	9082
her	861	green	931	their	3702	their	3702	one	8335	sits	8847

at every time step, sequence with lower score at earlier time step tends to drop out easily, especially those with longer previous sequence. Thus, generating caption in a phrase-based manner avoid such problem, because the sequence score of short phrase gets less influence from previous words during beam search. As for the word '*there*', it can be inferred by our model because we split the decoding of AS and NPs separately, which naturally makes the prediction of word '*a*' the job of phrase decoder. Without the word '*a*' as competitor, the word '*there*' is more likely to be predicted as first word in a caption. The same applies for word '*three*' with word '*two*' as competitor. These are the reasons that our model is capable of generating more unique captions compared to the baseline.

On the other hand, the baseline model has a better chance to predict particle word '*up*' and conjunction '*from*' with influence from longer previous words. Other words which cannot be inferred by both models usually have alternative words that have higher score. For example, '*boy/girl*' and '*next to*' are better alternative to '*child*' and '*by*'. Moreover, both models are incapable of inferring conjunction '*while*' and '*as*', which are mostly used to describe multiple actions performed by the same or different individuals in an image.

8. Conclusion

This paper presented a phrase-based LSTM (phi-LSTM) model to generate image caption in a hierarchical manner, where NPs that describe the salient objects in an image are first generated, before a complete caption is formed from the NPs. Each generated NP is encoded as a compositional vector, which acts as the input of one time step at the sentence level. Such design allows NPs to be decoded in a consistent time-scale, while reducing the variation of time-scale resolution at the sentence level. Empirical results show that image caption generated in such manner is more precise in terms of object and attribute, compared to a pure sequential model using words as atomic unit. Moreover, the hierarchical decoding process allows more novel captions with diverse word content to be generated. Our future work will focus on designing of a phrase-based bi-directional model for image captioning.

Acknowledgments

This work is supported in part by the Postgraduate Research Grant (PPP) PG003-2016A, and in part by the Frontier Research Grant FG002-17AFR, from University of Malaya. Also, we gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

[1] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, B. Plank, Automatic description generation from images: A survey of models, datasets, and evaluation measures, J. Artif. Intell. Res. 55 (2016) 409–442.
[2] S. Bai, S. An, A survey on automatic image caption generation, Neurocomputing 311 (2018) 291–304.
[3] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, A. Yuille, Deep captioning with multimodal recurrent neural networks (m-RNN), in: Proceedings of the ICLR, 2015.
[4] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: a neural image caption generator, in: Proceedings of the CVPR, 2015, pp. 3156–3164.
[5] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: Proceedings of the CVPR, 2015, pp. 3128–3137.
[6] R. Kiros, R. Salakhutdinov, R. Zemel, Unifying visual-semantic embeddings with multimodal neural language models, arXiv:1411.2539(2014).
[7] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in: Proceedings of the CVPR, 2015, pp. 2625–2634.
[8] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[9] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: neural image caption generation with visual attention, in: Proceedings of the ICML, 2015, pp. 2048–2057.
[10] L. Li, S. Tang, L. Deng, Y. Zhang, Q. Tian, Image caption with global-local attention., in: Proceedings of the AAAI, 2017, pp. 4133–4139.
[11] Z. Yang, Y. Yuan, Y. Wu, W.W. Cohen, R.R. Salakhutdinov, Review networks for caption generation, in: Proceedings of the NIPS, 2016, pp. 2361–2369.
[12] K. Fu, J. Jin, R. Cui, F. Sha, C. Zhang, Aligning where to see and what to tell: image captioning with region-based attention and scene-specific contexts, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2321–2334.
[13] Q. Wu, C. Shen, P. Wang, A. Dick, A. van den Hengel, Image captioning and visual question answering based on attributes and external knowledge, IEEE Trans. Pattern Anal. Mach. Intell. 40 (6) (2018) 1367–1381.
[14] T. Yao, Y. Pan, Y. Li, Z. Qiu, T. Mei, Boosting image captioning with attributes, in: Proceedings of the IEEE ICCV, 2017, pp. 22–29.
[15] Q. You, H. Jin, Z. Wang, C. Fang, J. Luo, Image captioning with semantic attention, in: Proceedings of the CVPR, 2016, pp. 4651–4659.
[16] M. Hermans, B. Schrauwen, Training and analysing deep recurrent neural networks, in: Proceedings of the NIPS, 2013, pp. 190–198.
[17] V. Yngve, A model and an hypothesis for language structure, Proc. Am. Philos. Soc. 104 (5) (1960) 444–466.

[18] I.V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. Courville, Y. Bengio, A hierarchical latent variable encoder-decoder model for generating dialogues, in: Proceedings of the AAAI, 2017.
[19] C. Rashtchian, P. Young, M. Hodosh, J. Hockenmaier, Collecting image annotations using Amazon's mechanical turk, in: NAACL: Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, 2010, pp. 139–147.
[20] P. Young, A. Lai, M. Hodosh, J. Hockenmaier, From image descriptions to visual denotations: new similarity metrics for semantic inference over event descriptions, Trans. ACL 2 (2014) 67–78.
[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: common objects in context, in: Proceedings of the ECCV, 2014, pp. 740–755.
[22] Y.H. Tan, C.S. Chan, phi-LSTM: A phrase-based hierarchical LSTM model for image captioning, in: Proceedings of the ACCV, 2016, pp. 101–117.
[23] S. Banerjee, A. Lavie, Meteor: An automatic metric for MT evaluation with improved correlation with human judgments, in: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, 29, 2005, pp. 65–72.
[24] C.-Y. Lin, Rouge: a package for automatic evaluation of summaries, in: Proceedings of the ACL Workshop Text Summarization Branches Out, 8, Barcelona, Spain, 2004.
[25] R. Vedantam, C. Lawrence Zitnick, D. Parikh, Cider: consensus-based image description evaluation, in: Proceedings of the CVPR, 2015, pp. 4566–4575.
[26] P. Anderson, B. Fernando, M. Johnson, S. Gould, Spice: semantic propositional image caption evaluation, in: Proceedings of the ECCV, 2016, pp. 382–398.
[27] A. Farhadi, M. Hejrati, M. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, D. Forsyth, Every picture tells a story: generating sentences from images, in: Proceedings of the ECCV, 2010, pp. 15–29.
[28] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. Berg, T. Berg, Babytalk: understanding and generating simple image descriptions, IEEE Trans. Pattern Anal. Mach. Intell. 35 (12) (2013) 2891–2903.
[29] S. Li, G. Kulkarni, T.L. Berg, A.C. Berg, Y. Choi, Composing simple image descriptions using web-scale n-grams, in: Proceedings of the CoNLL, ACL, 2011, pp. 220–228.
[30] Y. Yang, C.L. Teo, H. Daumé III, Y. Aloimonos, Corpus-guided sentence generation of natural images, in: Proceedings of the EMNLP, ACL, 2011, pp. 444–454.
[31] M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. Berg, K. Yamaguchi, T. Berg, K. Stratos, H. Daumé III, Midge: generating image descriptions from computer vision detections, in: Proceedings of the 13th Conference of the European Chapter of the ACL, 2012, pp. 747–756.
[32] P. Kuznetsova, V. Ordonez, A.C. Berg, T.L. Berg, Y. Choi, Collective generation of natural image descriptions, in: Proceedings of the ACL, 2012, pp. 359–368.
[33] P. Kuznetsova, V. Ordonez, T. Berg, Y. Choi, Treetalk: composition and compression of trees for image descriptions, Trans. ACL 2 (10) (2014) 351–362.
[34] R. Socher, A. Karpathy, Q.V. Le, C.D. Manning, A. Ng, Grounded compositional semantics for finding and describing images with sentences, Trans. ACL 2 (1) (2014) 207–218.
[35] J. Devlin, H. Cheng, H. Fang, S. Gupta, L. Deng, X. He, G. Zweig, M. Mitchell, Language models for image captioning: the quirks and what works, CoRR (2015). arXiv: 1505.01809.
[36] R. Kiros, R. Salakhutdinov, R. Zemel, Multimodal neural language models, in: Proceedings of the ICML, 2014, pp. 595–603.
[37] R. Lebret, P.O. Pinheiro, R. Collobert, Phrase-based image captioning, in: Proceedings of the ICML, 2015, pp. 2085–2094.
[38] X. Jia, E. Gavves, B. Fernando, T. Tuytelaars, Guiding the long-short term memory model for image caption generation, in: Proceedings of the ICCV, 2015, pp. 2407–2415.
[39] Y. Ushiku, M. Yamaguchi, Y. Mukuta, T. Harada, Common subspace for model and similarity: phrase learning for caption generation from images, in: Proceedings of the ICCV, 2015, pp. 2668–2676.
[40] A. Karpathy, A. Joulin, F.F.F. Li, Deep fragment embeddings for bidirectional image sentence mapping, in: Proceedings of the NIPS, 2014, pp. 1889–1897.
[41] H. Fang, S. Gupta, F. Iandola, R.K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J.C. Platt, et al., From captions to visual concepts and back, in: Proceedings of the CVPR, 2015, pp. 1473–1482.
[42] P. Kinghorn, L. Zhang, L. Shao, A region-based image caption generator with refined descriptions, Neurocomputing 272 (2018) 416–424.
[43] Z. Wang, X. Liu, L. Chen, L. Wang, Y. Qiao, X. Xie, C. Fowlkes, Structured triplet learning with pos-tag guided attention for visual question answering, in: Proceedings of the WACV, 2018, pp. 1888–1896.
[44] V. Ordonez, G. Kulkarni, T.L. Berg, Im2text: describing images using 1 million captioned photographs, in: Proceedings of the NIPS, 2011, pp. 1143–1151.
[45] M. Hodosh, P. Young, J. Hockenmaier, Framing image description as a ranking task: Data, models and evaluation metrics, J. Artif. Intell. Res. (2013) 853–899.
[46] A. Gupta, Y. Verma, C. Jawahar, Choosing linguistics over vision to describe images., in: Proceedings of the AAAI, 2012, pp. 606–612.
[47] J. Mun, M. Cho, B. Han, Text-guided attention model for image captioning., in: Proceedings of the AAAI, 2017, pp. 4233–4239.
[48] X. Chen, C. Lawrence Zitnick, Mind's eye: a recurrent visual representation for image caption generation, in: Proceedings of the CVPR, 2015, pp. 2422–2431.
[49] P. Tang, H. Wang, S. Kwong, Deep sequential fusion LSTM network for image description, Neurocomputing 312 (2018) 154–164.
[50] X. Liang, X. Shen, J. Feng, L. Lin, S. Yan, Semantic object parsing with graph LSTM, in: Proceedings of the ECCV, Springer, 2016, pp. 125–143.
[51] Y. Wang, Z. Lin, X. Shen, S. Cohen, G.W. Cottrell, Skeleton key: Image captioning by skeleton-attribute decomposition, in: Proceedings of the CVPR, 2017, pp. 7272–7281.
[52] C.D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S.J. Bethard, D. McClosky, The stanford CoreNLP natural language processing toolkit, in: Proceedings of the ACL System Demonstrations, 2014, pp. 55–60.
[53] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the ACL, 2002, pp. 311–318.
[54] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR (2014). arXiv: 1409.1556.
[55] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the CVPR, 2015, pp. 1–9.
[56] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the CVPR, 2016, pp. 770–778.
[57] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: Lessons learned from the 2015 mscoco image captioning challenge, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2017) 652–663.

Ying Hua Tan received her B.E. degree in Mechatronics Engineering from UCSI University, Malaysia in 2013. She is currently working towards Ph.D. in Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. Her research interests include deep learning in computer vision and natural language processing, with main focus on image and video captioning.

Chee Seng Chan received his Ph.D. degree from University of Portsmouth, U.K., in 2008. He is currently an associate professor with the Faculty of Computer Science & Information Technology, University of Malaya, Malaysia. His research interests are compyter vision and fuzzy set theory, particularly focus on image/video content analysis. He is/was the founding chair of the IEEE CIS Malaysia chapter, the organizing chair of the 3rd Asian Conference on Patter Recognition (2015), and the general chair of the IEEE 21st International Workshop on Multimedia Signal Processing (2019) and IEEE International Conference on Visual Communications and Image Processing (2013). He is a senior member of IEEE.