Oracle Performance for Visual Captioning

Li Yao[1]
li.yao@umontreal.ca

Nicolas Ballas[1]
nicolas.ballas@umontreal.ca

Kyunghyun Cho[3]
kyunghyun.cho@nyu.edu

John R. Smith[2]
jsmith@us.ibm.com

Yoshua Bengio[1]
yoshua.bengio@umontreal.ca

[1] Université de Montréal

[2] IBM T.J. Watson Research

[3] New York University

Abstract

The task of associating images and videos with a natural language description has attracted a great amount of attention recently. The state-of-the-art results on some of the standard datasets have been pushed into the regime where it has become more and more difficult to make significant improvements. Instead of proposing new models, this work investigates performances that an oracle can obtain. In order to disentangle the contribution from visual model from the language model, our oracle assumes that high-quality visual concept extractor is available and focuses only on the language part. We demonstrate the construction of such oracles on MS-COCO, YouTube2Text and LSMDC (a combination of M-VAD and MPII-MD). Surprisingly, despite the simplicity of the model and the training procedure, we show that current state-of-the-art models fall short when being compared with the learned oracle. Furthermore, it suggests the inability of current models in capturing important visual concepts in captioning tasks.

1 Introduction

With standard datasets publicly available, such as COCO and Flickr [15, 22, 38] in image captioning, and YouTube2Text, MVAD and MPI-MD [12, 27, 30] in video captioning, the field has been progressing in an astonishing speed. For instance, the state-of-the-art results on COCO image captioning has been improved rapidly from 0.17 to 0.31 in BLEU [3, 9, 10, 17, 18, 23, 25, 34, 36]. Similarly, the benchmark on YouTube2Text has been repeatedly pushed from 0.31 to 0.50 in BLEU score [1, 26, 28, 32, 33, 35, 37, 39]. While obtaining encouraging results, captioning approaches involve large networks, usually leveraging convolution network for the visual part and recurrent network for the language side. It therefore results model with a certain complexity where the contribution of the different component is not clear.

Instead of proposing better models, the main objective of this work is to develop a method that offers a deeper insight of the strength and the weakness of popular visual captioning

models. In particular, we propose a trainable oracle that disentangles the contribution of the visual model from the language model. To obtain such oracle, we follow the assumption that the image and video captioning task may be solved with two steps [11, 28] . Consider the model $P(\mathbf{w}|\mathbf{v})$ where $\mathbf{v}$ refers to usually high dimensional visual inputs, such as representations of an image or a video, and $\mathbf{w}$ refers to a caption, usually a sentence of natural language description. In order to work well, $P(\mathbf{w}|\mathbf{v})$ needs to form higher level visual concept, either explicitly or implicitly, based on $\mathbf{v}$ in the first step, denoted as $P(\mathbf{a}|\mathbf{v})$, followed by a language model that transforms visual concept into a legitimate sentence, denoted by $P(\mathbf{w}|\mathbf{a})$. $\mathbf{a}$ referes to *atoms* that are visually perceivable from $\mathbf{v}$.

The above assumption suggests an alternative way to build an oracle. In particular, we assume the first step is *close to perfect* in the sense that visual concept (or hints) is observed with almost 100% accuracy. And then we train the best language model conditioned on hints to produce captions.

Using the proposed oracle, we compare the current state-of-the-art models against it, which helps to quantify their capacity of visual modeling, a major weakness, apart from the strong language modeling. In addition, when being applied on different datasets, the oracle offers insight on the intrinsic difficulty and blessing of them, a general guideline when designing new algorithms and developing new models. Finally, we also relax the assumption to investigate the case where visual concept may not be realistically predicted with 100% accuracy and demonstrate a quantity-accuracy trade-off in solving visual captioning tasks.

2   Related work

Visual captioning   The problem of image captioning has attracted a great amount of attention lately. Early work focused on constructing linguistic templates or syntactic trees based on a set of concept from visual inputs [20, 21, 24]. Another popular approach is based on caption retrieval in the embedding space such as Devlin et al. [9], Kiros et al. [18]. Most recently, the use of language models conditioned on visual inputs have been widely studied in the work of Fang et al. [11] where a maximum entropy language model is used and in Donahue et al. [10], Karpathy and Fei-Fei [17], Mao et al. [23], Vinyals et al. [34], Xu et al. [36] where recurrent neural network based models are built to generate natural language descriptions. The work of Devlin et al. [8] advocates to combine both types of language models. Furthermore, CIDEr [31] was proposed as an alternative evaluation metric for image captioning and is shown to be more advantageous compared with BLEU and METEOR. To further improve the performance, Bengio et al. [3] suggests a simple sampling algorithm during training, which was one of the winning recipes for MSR-COCO Captioning challenge [1], and Jia et al. [16] suggests the use of extra semantic information to guide the language generation process.

Similarly, video captioning has made substantial progress recently. Early models such as Barbu et al. [2], Kojima et al. [19], Rohrbach et al. [28] tend to focus on constrained domains with limited appearance of activities and objects in videos. They also rely heavily on hand-crafted video features, followed by a template-based or shallow statistical machine translation approaches to produce captions. Borrowing success from image captioning, recent models such as Donahue et al. [10], Rohrbach et al. [26], Venugopalan et al. [32, 33], Xu et al. [35], Yao et al. [37], Yu et al. [39] and most recently Ballas et al. [1] have adopted a

---
[1]http://mscoco.org

more general encoder-decoder approach with end-to-end parameter tuning. Videos are input into a specific variant of encoding neural networks to form a higher level visual summary, followed by a caption decoder by recurrent neural networks. Training such type of models are possible with the availability of three relatively large scale datasets, one collected from YouTube by Guadarrama et al. [12], the other two constructed based on Descriptive Video Service (DVS) on movies by Torabi et al. [30] and Rohrbach et al. [27]. The latter two have recently been combined as the official dataset for Large Scale Movie Description Challenge (LSMDC) [2].

Capturing higher-level visual concept   The idea of using intermediate visual concept to guide the caption generation has been discussed in Qi Wu et al. [25] in the context of image captioning and in Rohrbach et al. [26] for video captioning. Both work trained classifiers on a predefined set of visual concepts, extracted from captions using heuristics from linguistics and natural language processing. Our work resembles both of them in the sense that we also extract similar constituents from captions. The purpose of this study, however, is different. By assuming perfect classifiers on those visual atoms, we are able to establish the performance upper bounds for a particular dataset. Note that a simple bound is suggested by Rohrbach et al. [26] where METEOR is measured on all the training captions against a particular test caption. The largest score is picked as the upper bound. As a comparison, our approach constructs a series of oracles that are trained to generate captions given different number of visual hints. Therefore, such bounds are clear indication of models' ability of capturing concept within images and videos when performing caption generation, instead of the one suggested by Rohrbach et al. [26] that performs caption retrieval.

3 Oracle Model

The construction of the oracle is inspired by the observation that $P(\mathbf{w}|\mathbf{v}) = \sum_{\mathbf{a}} P_\theta(\mathbf{w}|\mathbf{a})P(\mathbf{a}|\mathbf{v})$ where $\mathbf{w} = \{\mathbf{w}_1, \ldots, \mathbf{w}_t\}$ denotes a caption containing a sequence of words having a length t. $\mathbf{v}$ denotes the visual inputs such as an image or a video. $\mathbf{a}$ denotes visual concepts which we call "atoms". We have explicitly factorized the captioning model $P(\mathbf{w}|\mathbf{v})$ into two parts, $P(\mathbf{w}|\mathbf{a})$, which we call the conditional language model given atoms, and $P(\mathbf{a}|\mathbf{v})$, which we call conditional atom model given visual inputs. To establish the oracle, we assume that the atom model is given, which amounts to treat $P(\mathbf{a}|\mathbf{v})$ as a Dirac delta function that assigns all the probability mass to the observed atom $\mathbf{a}$. In other words, $P(\mathbf{w}|\mathbf{v}) = P_\theta(\mathbf{w}|\mathbf{a})$.

Therefore, with the fully observed $\mathbf{a}$, the task of image and video captioning reduces to the task of language modeling conditioned on atoms. This is arguably a much easier task compared with the direct modeling of $P(\mathbf{w}|\mathbf{v})$, therefore a well-trained model could be treated as a performance oracle of it. Information contained in $\mathbf{a}$ directly influences the difficulty of modeling $P_\theta(\mathbf{w}|\mathbf{a})$. For instance, if no atoms are available, $P_\theta(\mathbf{w}|\mathbf{a})$ reduces to unconditional language modeling, which could be considered as a lower bound of $P(\mathbf{w}|\mathbf{v})$. By increasing the amount of information $\mathbf{a}$ carries, the modeling of $P_\theta(\mathbf{w}|\mathbf{a})$ becomes more and more straightforward.

3.1 Oracle Parameterization

Given a set of atoms $\mathbf{a}^{(k)}$ that summarize the visual concept appearing in the visual inputs $\mathbf{v}$, this section describes the detailed parameterization of the model $P_\theta(\mathbf{w}|\mathbf{a}^{(k)})$ with θ denoting the overall parameters. In particular, we adopt the commonly used encoder-decoder framework [7] to model this conditional based on the following simple factorization $P_\theta(\mathbf{w}|\mathbf{a}^{(k)}) = \prod_{t=1}^{T} P_\theta(\mathbf{w}_t|\mathbf{w}_{<t}, \mathbf{a}^{(k)})$.

Recurrent neural networks (RNNs) are natural choices when outputs are identified as sequences. We borrow the recent success from a variant of RNNs called Long-short term memory networks (LSTMs) first introduced in Hochreiter and Schmidhuber [14], formulated as the following

$$
\begin{bmatrix}
p(\mathbf{w}_t \mid \mathbf{w}_{<t}, \mathbf{a}^{(k)})) \\
\mathbf{h}_t \\
\mathbf{c}_t
\end{bmatrix}
= \psi(\mathbf{h}_{t-1}, \mathbf{c}_{t-1}, \mathbf{w}_{t-1}, \mathbf{a}^{(k)}),
\tag{1}
$$

where $\mathbf{h}_t$ and $\mathbf{c}_t$ represent the RNN state and memory of LSTMs at timestep t respectively. Combined with the atom representation, Equ. (1) is implemented as following

$$
\begin{aligned}
\mathbf{f}_t &= \sigma(\mathbf{W}_f \mathbf{E}_w[\mathbf{w}_{t-1}] + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{A}_f \Phi(\mathbf{a}^{(k)}) + \mathbf{b}_f), \\
\mathbf{i}_t &= \sigma(\mathbf{W}_i \mathbf{E}_w[\mathbf{w}_{t-1}] + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{A}_i \Phi(\mathbf{a}^{(k)}) + \mathbf{b}_i), \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o \mathbf{E}_w[\mathbf{w}_{t-1}] + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{A}_o \Phi(\mathbf{a}^{(k)}) + \mathbf{b}_o), \\
\tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_c \mathbf{E}_w[\mathbf{w}_{t-1}] + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{A}_c \Phi(\mathbf{a}^{(k)}) + \mathbf{b}_c), \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \\
\mathbf{h}_t &= \mathbf{o}_t \odot \mathbf{c}_t,
\end{aligned}
$$

where $\mathbf{E}_w$ denotes the word embedding matrix, as apposed to the atom embedding matrix $\mathbf{E}_a$, $\mathbf{W}$, $\mathbf{U}$, $\mathbf{A}$ and $\mathbf{b}$ are parameters of the LSTM. With the LSTM's state $\mathbf{h}_t$, the probability of the next word in the sequence is $\mathbf{p}_t = \text{softmax}(\mathbf{U}_p \tanh(\mathbf{W}_p \mathbf{h}_t + \mathbf{b}_p) + \mathbf{d})$ with parameters $\mathbf{U}_p$, $\mathbf{W}_p$, $\mathbf{b}_p$ and $\mathbf{d}$. The overall training criterion of the oracle is

$$
\theta = \arg\max_{\theta} \mathcal{U}_k(\theta) = \log \prod_{n=1}^{N} P_\theta(\mathbf{w}^{(n)}|\mathbf{a}^{(n,k)}) = \sum_{n=1}^{N} \sum_{t=1}^{T} \log P_\theta(\mathbf{w}_t^{(n)}|\mathbf{w}_{<t}^{(n)}, \mathbf{a}^{(n,k)}),
\tag{2}
$$

given N training pairs $(\mathbf{w}^{(n)}, \mathbf{a}^{(n,k)})$. θ represents parameters in the LSTM.

3.2 Atoms Construction

Each configuration of $\mathbf{a}$ may be associated with a different distribution $P_\theta(\mathbf{w}|\mathbf{a})$, therefore a different oracle model. We define configuration as an orderless collection of unique atoms. That is, $\mathbf{a}^{(k)} = \{a_1, \ldots, a_k\}$ where k is the size of the bag and all items in the bag are different from each other. Considering the particular problem of image and video captioning, atoms are defined as words in captions that are most related to actions, entities, and attributes of entities (in Figure 1). The reason of using these three particular choices of language components as atoms is not an arbitrary decision. It is reasonable to consider these three types among the most visually perceivable ones when human describes visual content in natural language. We further verify this by conducting a human evaluation procedure to

identify "visual" atoms from this set and show that a dominant majority of them indeed match human visual perception, detailed in Section 5.1. Being able to capture these important concepts is considered as crucial in getting superior performance. Therefore, comparing the performance of existing models against this oracle reveals their ability of capturing atoms from visual inputs when $P(\mathbf{a}|\mathbf{v})$ is unknown.

A set of atoms $\mathbf{a}^{(k)}$ is treated as "a bag of words". As with the use of word embedding matrix in neural language modeling [4], the ith atom $\mathbf{a}_i^{(k)}$ is used to index the atom embedding matrix $\mathbf{E}_a[\mathbf{a}_i^{(k)}]$ to obtain a vector representation of it. Then the representation of the entire set of atoms is $\Phi(\mathbf{a}^{(k)}) = \sum_{i=1}^{k} \mathbf{E}_a[\mathbf{a}_i^{(k)}]$.

4 Contributing factors of the oracle

The formulation of Section 3 is generic, only relying on the assumption the two-step visual captioning process, independent of the parameterization in Section 3.1. In practice, however, one needs to take into account several contributing factors to the oracle.

Firstly, atoms, or visual concepts, may be defined as 1-gram words, 2-gram phrases and so on. Arguably a mixture of N-gram representations has the potential to capture more complicated correlations among visual concepts. For simplicity, this work uses only 1-gram representations, detailed in Section 5.1. Secondly, the procedure used to extract atoms needs to be reliable, extracting mainly *visual* concepts, leaving out *non-visual* concepts. To ensure this, the procedure used in this work is verified with human evaluation, detailed in 5.1. Thirdly, the modeling capacity of the conditional language $P_\theta(\mathbf{w}|\mathbf{a}^{(k)})$ has a direct influence on the obtained oracle. Section 3.1 has shown one example of many possible parameterizations. Lastly, the oracle may be sensitive to the training procedure and its hyper-parameters (see Section 5.2).

While it is therefore important to keep in mind that proposed oracle *conditions* on the above factors, quite surprisingly, however, with the simplest procedure and parameterization we show in the experimental section that oracle serves their purpose reasonably well.

5 Experiments

We demonstrate the procedure of learning the oracle on three standard visual captioning datasets. MS COCO [22] is the most commonly used benchmark dataset in image captioning. It consists of 82,783 training and 40,504 validation images. each image accompanied by 5 captions, all in one sentence. We follow the split used in Xu et al. [36] where a subset of 5,000 images are used as validation, and another subset of 5,000 images are used for testing. YouTube2Text is the most commonly used benchmark dataset in video captioning. It consists of 1,970 video clips, each accompanied with multiple captions. Overall, there are 80,000 video and caption pairs. Following Yao et al. [37], it is split into 1,200 clips for training, 100 for validation and 670 for testing. Another two video captioning datasets have been recently introduced in Torabi et al. [30] and Rohrbach et al. [27]. Compared with YouTube2Text, they are both much larger in the number of video clips, most of which are associated with one or two captions. Recently they are merge together for Large Scale Movie Description Challenge (LSMDC). [3] We therefore name this particular dataset LSMDC. The

---

[3] https://goo.gl/2hJ4lw

official splits contain 91,908 clips for training, 6,542 for validation and 10,053 for testing.

5.1 Atom extraction

visual input	caption	entity	action	attribute
	A blue train sitting along side of a green forest.	train, side, forest	sit	blue, green
	Old train left out on the ground has graffiti all over it	train, it, ground	leave, graffiti, have	old
	Two abandoned blue and white train cars next to trees.	car, train, tree	abandon	blue, white, next
	A man with a skateboard under him, not touching, are in the air.	air, skateboard, him, man	touch	NA
	Two guys are skateboarding and performing jumps and tricks.	jump, trick, guy	perform, skateboard	NA
	Two men doing tricks on skateboards at the skate park	trick, park, men, skateboard	do	skate

Figure 1: Given ground truth captions, three categories of visual atoms (entity, action and attribute) are automatically extracted using NLP Parser. "NA" denotes the empty atom set.

Visual concepts in images or videos are summarized as atoms that are provided to the caption language model. They are split into three categories: actions, entities, and attributes. To identify these three classes, we utilize Stanford natural language parser [4] to automatically extract them. After a caption is parsed, we apply simple heuristics based on the tags produced by the parser, ignoring the phrase and sentence level tags [5]: Use words tagged with {"NN", "NNP", "NNPS" ,"NNS", "PRP"} as entity atoms. Use words tagged with {"VB", "VBD", "VBG", "VBN", "VBP", "VBZ"} as action atoms. Use words tagged with {"JJ", "JJR", "JJS"} as attribute atoms. After atoms are identified, they are lemmatized with NLTK lemmatizer [6] to unify them to their original dictionary format [7]. Figure 1 illustrates some results. We extracted atoms for COCO, YouTube2Text and LSMDC. This gives 14,207 entities, 4,736 actions and 8,671 attributes for COCO, 6,922 entities, 2,561 actions, 2,637 attributes for YouTube2Text, and 12,895 entities, 4,258 actions, 8550 attributes for LSMDC. Note that although the total number of atoms of each categories may be large, atom frequency varies. In addition, the language parser does not guarantee the perfect tags. Therefore, when atoms are being used in training the oracle, we sort them according to their frequency and make sure to use more frequent ones first to also give priority to atoms with larger coverage, detailed in Section 5.2 below.

We conducted a simple human evaluation [8] to confirm that extracted atoms are indeed predominantly visual. As it might be impractical to evaluate all the extracted atoms for all three datasets, we focus on top 150 frequent atoms. This evaluation intends to match the last column of Table 2 where current state-of-the-art models have the equivalent capacity of capturing perfectly less than 100 atoms from each of three categories. Subjects are asked to cast their vote independently. The final decision of an atom being visual or not is made by majority vote. Table 1 shows the ratio of atoms flagged as visual by such procedure.

5.2 Training

After the atoms are extracted, they are sorted according to the frequency they appear in the dataset, with the most frequent one leading the sorted list. Taking first k items from this list

---

[4] http://goo.gl/lSvPr
[5] complete list of tags: https://goo.gl/fU8zDd
[6] http://www.nltk.org/
[7] available at https://goo.gl/t7vtFj
[8] details available at https://goo.gl/t7vtFj

Table 1: Human evaluation of proportion of atoms that are voted as visual. It is clear that extracted atoms from three categories contain dominant amount of visual elements, hence verifying the procedure described in Section 3.2. Another observation is that entities and actions tend to be more visual than attributes according to human perception.

	entities	actions	attributes
COCO	92%	85%	81%
YouTube2Text	95%	91%	72%
LSMDC	83%	87%	78%

gives the top k most frequent ones, forming a bag of atoms denoted by $\mathbf{a}^{(k)}$ where k is the size of the bag. Conditioned on the atom bag, the oracle is maximized as in Equ (2).

To form captions, we used a vocabulary of size 20k, 13k and 25k for COCO, YouTube2Text and LSMDC respectively. For all three datasets, models were trained on training set with different configuration of (1) atom embedding size, (2) word embedding size and (3) LSTM state and memory size. To avoid overfitting we also experimented weight decay and Dropout [13] to regularize the models with different size. In particular, we experimented with random hyper-parameter search by Bergstra and Bengio [5] with range $[128, 1024]$ on (1), (2) and (3). Similarly we performed random search on the weight decay coefficient with a range of $[10^{-6}, 10^{-2}]$, and whether or not to use dropout. Optimization was performed by SGD, minibatch size 128, and with Adadelta [40] to automatically adjust the per-parameter learning rate. Model selection was done on the standard validation set, with an early stopping patience of 2,000 (early stop if no improvement made after 2,000 minibatch updates). We report the results on the test splits.

5.3 Interpretation

All three metrics – BLEU, METEOR and CIDER are computed with Microsoft COCO Evaluation Server [6]. Figure 2 summarizes the learned oracle with an increasing number of k.

comparing oracle performance with existing models We compare the current state-of-the-art models' performance against the established oracles in Figure 2. Table 2 shows the comparison on three different datasets. With Figure 2, one could easily associate a particular performance with the equivalent number of atoms perfectly captured across all 3 atom categories, as illustrated in Table 2, the oracle included in bold. It is somehow surprising that state-of-the-art models have performance that is equivalent to capturing only a small amount of "ENT" and "ALL". This experiment highlights the shortcoming of the state-of-art visual models. By improving them, we could close the performance gap that we currently have with the oracles.

quantify the diminishing return As the number of atoms k in $\mathbf{a}^{(k)}$ increases, one would expect the oracle to be improved accordingly. It is however not yet clear the speed of such improvement. In other words, the gain in performance may not be proportional to the number of atoms given when generating captions, due to atom frequencies and language modeling. Figure 2 quantifies this effect. The oracle on all three datasets shows a significant gain at the beginning and diminishes quickly as more and more atoms are used.

Row 2 of Figure 2 also highlights the difference among actions, entities and attributes in generating captions. For all three datasets tested, entities played much more important roles,
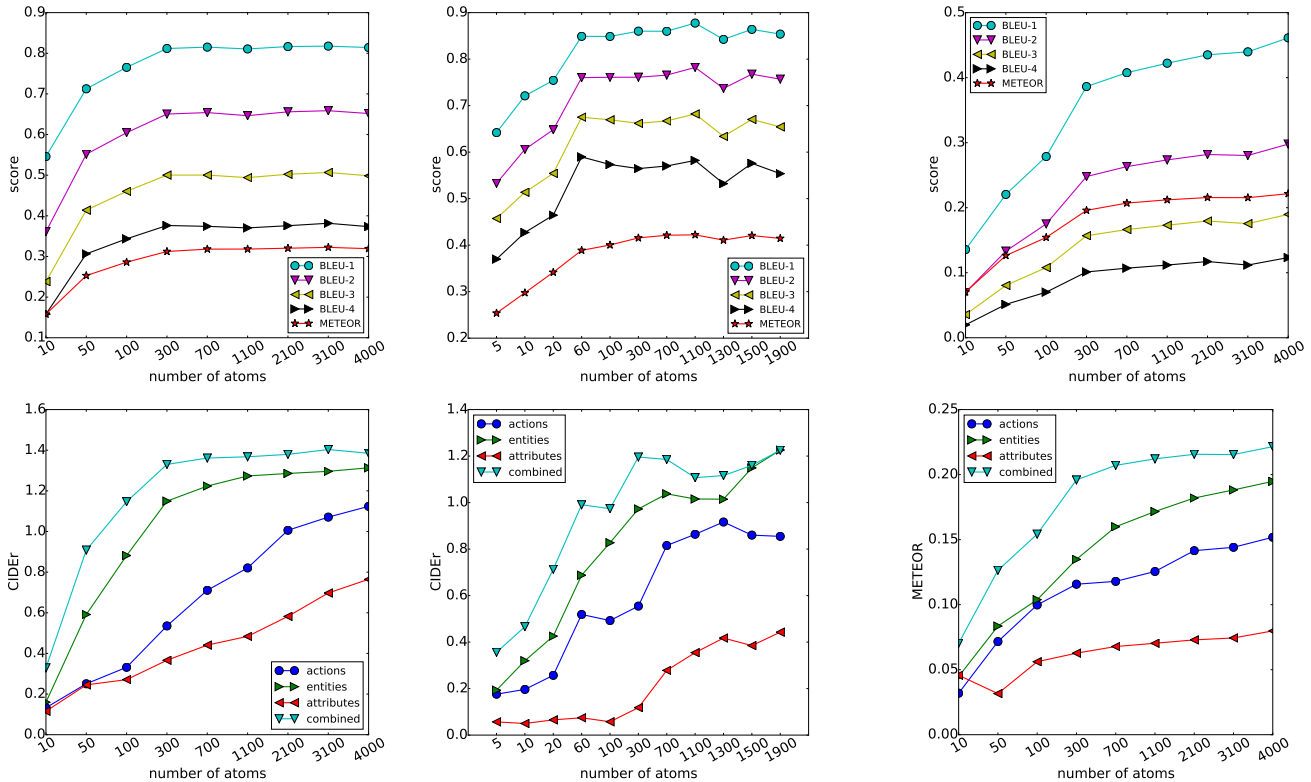
Figure 2: Learned oracle on COCO (left), YouTube2Text (middle) and LSMDC (right). The number of atoms $\mathbf{a}^{(k)}$ is varied on x-axis and oracles are computed on y-axis on testsets. The first row shows the oracles on BLEU and METEOR with $3k$ atoms, k from each of the three categories. The second row shows the oracles when k atoms are selected individually for each category. CIDEr is used for COCO and YouTube2Text as each test example is associated with multiple ground truth captions, argued in [31]. For LSMDC, METEOR is used, as argued by Rohrbach et al. [26].

even more so than action atoms in general. This is particularly true on LSMDC where the gain of modeling attributes is much less than the other two categories.

Although visual atoms dominant the three atom categories shown in Section 5.1, as they increase in number, more and more non-visual atoms may be included, such as "living", "try", "find" and "free" which are relatively difficult to be associated with a particular part of visual inputs. Excluding non-visual atoms in the conditional language model can further tighten the oracle bound as less hints are provided to it. The major difficulty lies in the labor of hand-separating visual atoms from non-visual ones as to the our best knowledge this is difficult to automate with heuristics.

atom accuracy versus atom quantity   We have assumed that the atoms are given, or in other words, the prediction accuracy of atoms is 100%. In reality, one would hardly expect to have a perfect atom classifier. There is naturally a trade-off between number of atoms one would like to capture and the prediction accuracy of it. Figure 3 quantifies this trade-off on COCO and LSMDC. It also indicates the upper limit of performance given different level of atom prediction accuracy. In particular, we have replaced $\mathbf{a}^{(k)}$ by $\hat{\mathbf{a}}_r^{(k)}$ where r portion of $\mathbf{a}^{(k)}$ are randomly selected and replaced by other randomly picked atoms not appearing in $\mathbf{a}^{(k)}$. The case of $r = 0$ corresponds to those shown in Figure 2. And the larger the ratio r, the worse the assumed atom prediction is. The value of r is shown in the legend of Figure 3. According to the figure, in order to improve the caption generation score, one would have two options, either by keeping the number of atoms fixed while improving the atom prediction accuracy

Table 2: Measure semantic capacity of current state-of-the-art models. Using Figure 2, one could easily map the reported metric to the number of visual atoms captured. This establishes an equivalence between a model, the proposed oracle and a model's semantic capacity. ("ENT" for entities. "ACT" for actions. "ATT" for attributes. "ALL" for all three categories combined. "B1" for BLEU-1, "B4" for BLEU-4. "M" for METEOR. "C" for CIDEr. Note that the CIDEr is between 0 and 10 according to Vedantam et al. [31]. The learned oracle is denoted in **bold**.

	B1	B4	M	C	ENT	ACT	ATT	ALL
COCO	0.74	0.31	0.26	0.94	$\sim$200	$\sim$2100	>4000	$\sim$ 50
[25]	**0.80**	**0.35**	**0.30**	**1.4**				
YouTube2Text	0.815	0.499	0.326	0.658	$\sim$60	$\sim$500	>1900	$\sim$ 20
[39]	**0.88**	**0.58**	**0.40**	**1.2**				
LSMDC	N/A	N/A	0.07	N/A	$\sim$40	$\sim$50	$\sim$4000	$\sim$10
[32]	**0.45**	**0.12**	**0.22**	**N/A**				

or by keeping the accuracy while increasing the number of included atoms. As state-of-art visual model already model around 1000 atoms, we hyphotesize that we could gain more in improving the atoms accuracy rather than increase the number of atom detected by those models.
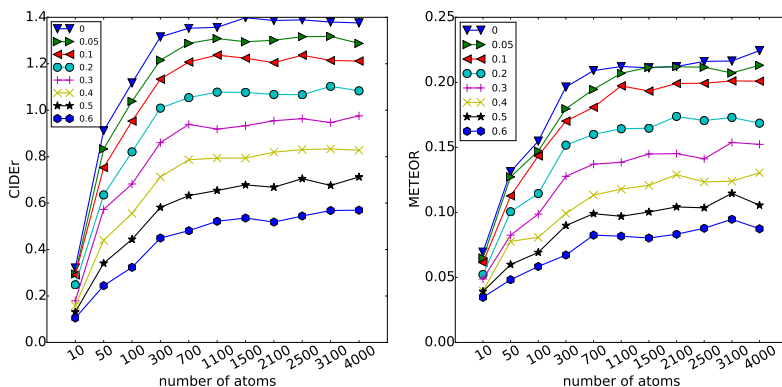

Figure 3: Learned oracles with different atom precision (r in red) and atom quantity (x-axis) on COCO (left) and LSMDC (right). The number of atoms $\hat{\mathbf{a}}_r^{(k)}$ is varied on x-axis and oracles are computed on y-axis on testsets. CIDEr is used for COCO and METEOR for LSMDC. It shows one could increase the score by either improving $P(\mathbf{a}^{(k)}|\mathbf{v})$ with a fixed k or increase k. It also shows the maximal error bearable for different score.

intrinsic difficulties of particular datasets Figure 2 also reveals the intrinsic properties of each dataset. In general, the bounds on YouTube2Text are much higher than COCO, with LSMDC the lowest. For instance, from the first column of the figure, taking 10 atoms respectively, BLUE-4 is around 0.15 for COCO, 0.30 for YouTube2Text and less than 0.05 for LSMDC. With little visual information to condition upon, a strong language model is required, which makes a dramatic difference across three datasets. Therefore the oracle, when compared across different datasets, offer an objective measure of difficulties of using them in the captioning task.

6 Discussion

This work formulates oracle performance for visual captioning. The oracle is constructed with the assumption of decomposing visual captioning into two consecutive steps. We have assumed the perfection of the first step where visual atoms are recognized, followed by the

second step where language models conditioned on visual atoms are trained to maximize the probability of given captions. Such an empirical construction requires only automatic atom parsing and the training of conditional language models, without extra labeling or costly human evaluation.

Such an oracle enables us to gain insight on several important factors accounting for both success and failure of the current state-of-the-art models. It further reveals model independent properties on different datasets. We furthermore relax the assumption of prefect atom prediction. This sheds light on a trade-off between atom accuracy and atom coverage, providing guidance to future research in this direction. Importantly, our experimental results suggest that more efforts are required in step one where visual inputs are converted into visual concepts (atoms).

Despite its effectiveness shown in the experiments, the empirical oracle is constructed with the simplest atom extraction procedure and model parameterization in mind, which makes such a construction in a sense a "conservative" oracle.

Acknowledgments

References

[1] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. *ICLR*, 2016.

[2] A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. Dickinson, S. Fidler, A. Michaux, S. Mussman, S. Narayanaswamy, D. Salvi, et al. Video in sentences out. *UAI*, 2012.

[3] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *arXiv preprint arXiv:1506.03099*, 2015.

[4] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003.

[5] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *JMLR*, 2012.

[6] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv 1504.00325*, 2015.

[7] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *EMNLP*, 2014.

[8] Jacob Devlin, Hao Cheng, Hao Fang, Saurabh Gupta, Li Deng, Xiaodong He, Geoffrey Zweig, and Margaret Mitchell. Language models for image captioning: The quirks and what works. *arXiv preprint arXiv:1505.01809*, 2015.

[9] Jacob Devlin, Saurabh Gupta, Ross Girshick, Margaret Mitchell, and C Lawrence Zitnick. Exploring nearest neighbor approaches for image captioning. *arXiv preprint arXiv:1505.04467*, 2015.

[10] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *CVPR*, 2015.

[11] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John Platt, et al. From captions to visual concepts and back. *CVPR*, 2015.

[12] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, 2013.

[13] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[15] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 2013.

[16] Xu Jia, Efstratios Gavves, Basura Fernando, and Tinne Tuytelaars. Guiding long-short term memory for image caption generation. *arXiv preprint arXiv:1509.04942*, 2015.

[17] A Karpathy and L Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2014.

[18] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[19] Atsuhiro Kojima, Takeshi Tamura, and Kunio Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *IJCV*, 2002.

[20] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *PAMI*, 2013.

[21] Polina Kuznetsova, Vicente Ordonez, Alexander C Berg, Tamara L Berg, and Yejin Choi. Collective generation of natural image descriptions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 359–368. Association for Computational Linguistics, 2012.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pages 740–755. Springer, 2014.

[23] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *ICLR*, 2015.

[24] Margaret Mitchell, Xufeng Han, Jesse Dodge, Alyssa Mensch, Amit Goyal, Alex Berg, Kota Yamaguchi, Tamara Berg, Karl Stratos, and Hal Daumé III. Midge: Generating image descriptions from computer vision detections. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 747–756. Association for Computational Linguistics, 2012.

[25] Qi Qi Wu, Chunhua Shen, Anton van den Hengel, Lingqiao Liu, and Anthony Dick. What value high level concepts in vision to language problems? *arXiv 1506.01144*, 2015.

[26] Anna Rohrbach, Marcus Rohrbach, and Bernt Schiele. The long-short story of movie description. 2015.

[27] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. *CVPR*, 2015.

[28] Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. Translating video content to natural language descriptions. In *ICCV*, 2013.

[29] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016. URL http://arxiv.org/abs/1605.02688.

[30] Atousa Torabi, Christopher Pal, Hugo Larochelle, and Aaron Courville. Using descriptive video services to create a large data source for video annotation research. *arXiv: 1503.01070*, 2015.

[31] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. *CVPR*, 2015.

[32] Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence – video to text. In *ICCV*, 2015.

[33] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. *NAACL*, 2015.

[34] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *CVPR*, 2014.

[35] Huijuan Xu, Subhashini Venugopalan, Vasili Ramanishka, Marcus Rohrbach, and Kate Saenko. A multi-scale multiple instance video description network. *arXiv 1505.05914*, 2015.

[36] Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *ICML*, 2015.

[37] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *ICCV*, 2015.

[38] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *ACL14*.

[39] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. *arXiv 1510.07712*, 2015.

[40] Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. Technical report, 2012.