EMPIRICAL PERFORMANCE UPPER BOUNDS FOR IMAGE AND VIDEO CAPTIONING

Li Yao
Université de Montréal
li.yao@umontreal.ca

Nicolas Ballas
Université de Montréal
nicolas.ballas@umontreal.ca

Kyunghyun Cho
New York University
kyunghyun.cho@nyu.edu

John R. Smith
IBM T.J. Watson Research
jsmith@us.ibm.com

Yoshua Bengio
Université de Montréal
yoshua.bengio@umontreal.ca

ABSTRACT

The task of associating images and videos with a natural language description has attracted a great amount of attention recently. Rapid progress has been made in terms of both developing novel algorithms and releasing new datasets. Indeed, 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 the possibility of empirically establishing performance upper bounds on various visual captioning datasets without extra data labelling effort or human evaluation. In particular, it is assumed that visual captioning is decomposed into two steps: from visual inputs to visual concepts, and from visual concepts to natural language descriptions. One would be able to obtain an upper bound when assuming the first step is perfect and only requiring training a conditional language model for the second step. We demonstrate the construction of such bounds on MS-COCO, YouTube2Text and LSMDC (a combination of M-VAD and MPII-MD). Surprisingly, despite of the imperfect process we used for visual concept extraction in the first step and the simplicity of the language model for the second step, we show that current state-of-the-art models fall short when being compared with the learned upper bounds. Furthermore, with such a bound, we quantify several important factors concerning image and video captioning: the number of visual concepts captured by different models, the trade-off between the amount of visual elements captured and their accuracy, and the intrinsic difficulty and blessing of different datasets.

1 Introduction

1.1 Motivation

Recent advance in machine learning, particularly deep learning, has shown great promise in developing powerful and flexible probabilistic models that are capable of handling real-world tasks. For instance, deep neural networks have been widely used to tackle challenging perception-related problems such as object recognition (Krizhevsky et al., 2012), speech recognition Dahl et al. (2012) and machine translation (Cho et al., 2014; Sutskever et al., 2014), Go playing (Silver et al., 2016) with great success. This has further inspired a new generation of research that adapts deep learning models to the task of image and video captioning where a natural language description is automatically produced by a model when it is presented with only visual inputs. Therefore, image and video captioning goes beyond the simple classification where outputs are independent classes. Both visual inputs and output captions are rich in their statistical structure, making it necessary to model not only different modalities individually but also the relationship between them.

With standard datasets publicly available, such as COCO (Lin et al., 2014) and Flickr (Hodosh et al., 2013; Young et al.) in image captioning, and YouTube2Text (Guadarrama et al., 2013), MVAD (Torabi et al., 2015), and MPI-MD (Rohrbach et al., 2015b) 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 with a series of work by Kiros et al. (2014); Devlin et al. (2015b); Donahue et al. (2015); Vinyals et al. (2014); Xu et al. (2015b); Mao et al. (2015); Karpathy & Fei-Fei (2014); Bengio et al. (2015); Qi Wu et al. (2015). Similarly, the benchmark on YouTube2Text has been repeatedly pushed from 0.31 to 0.50 in BLEU score, with the work of Rohrbach et al. (2013); Venugopalan et al. (2015b); Yao et al. (2015); Venugopalan et al. (2015a); Xu et al. (2015a); Rohrbach et al. (2015a); Yu et al. (2015); Ballas et al. (2016). Furthermore, the evaluation procedure has also been established and agreed upon. Automatic evaluation metrics such as BLEU (Papineni et al., 2002), METEOR (Denkowski & Lavie, 2014), CIDEr (Vedantam et al., 2015) are used to measure the quality of generated captions, while testset perplexity or likelihood may also be used when available. Inevitably, as it has happened only so often, the benchmark on image and video captioning has or will most likely reach to a point where further improvement may become less and less significant.

Instead of proposing novel models to improve any benchmark, this work takes a different view of the problem and investigates the possibility of upper bounding the performance of visual captioning tasks. Admittedly, the ideal upper bound is the human performance. Unfortunately, such an upper bound hardly provides enough guidance for problem solving or understanding. The main objective of this work is therefore to develop a method that offers not only just the upper bound itself but also deeper insight of the potential pros and cons of various models and algorithms. We therefore propose an empirically trainable and model-based upper bound.

1.2 Fundamental assumptions

To obtain an upper bound, we follow the assumption implicitly used in Rohrbach et al. (2013) and Fang et al. (2015) (see Section 2.3) that the image and video captioning task may be solved with two steps. 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* (see Section 3.1) that are visually perceivable from $\mathbf{v}$. Such a two-step process is elaborated in Section 3.

The above assumption suggests an alternative way to build a performance upper bound. In particular, we assume the first step is *perfect* in the sense that visual concept (or hints) is observed with 100% accuracy (this assumption is however relaxed in Section 6.4.3 for analysis). And then we train the best language model conditioned on hints to produce captions. This procedure constructs a mapping from abstract concept to language, and as one increases or decreases the number of concept observed (see Section 6.4.1), the performance of the conditional language model would vary accordingly.

When sufficiently trained, it is therefore possible to establish a series of performance upper bounds each of which is associated with different number of observed visual concept.

1.3 GAINING INSIGHTS FROM THE UPPER BOUNDS

Once upper bounds are obtained for a dataset in question, one could compare the performance of state-of-the-art models against them. Specifically, those models typically report their performance based on standard evaluation metrics. And the proposed upper bounds connect the number of captured visual concept with the optimal performance achievable on the same type of metrics.

Therefore, using the bounds, one could easily map the performance of different models of $P(\mathbf{w}|\mathbf{v})$ directly into the number of concept a detector would need to capture in order to reach the same performance. This comparison offers insights into separating contribution from visual and language modeling. Without the bound, one could only make overall judgement of joint effectiveness of both.

In addition, the bound offers insight into the intrinsic difficulties of various datasets. Comparing the performance upper bound across different datasets provides a quantitative and objective way to show how hard the task on those datasets could be. In fact, we have observed in this way that the widely used standard benchmark datasets have vastly different properties in nature. This provides a general guideline when developing and competing novel algorithms.

1.4 CONTRIBUTIONS

In summary, this work takes on the problem of image and video captioning and contributes in several aspects:

1. We propose a model-based method to empirically compute an upper bound on the performance of image and video captioning models. The proposed bound is adaptive and trainable on a particular dataset.
2. Using the proposed bound, we suggest to compare the current state-of-the-art model against it. This comparison helps to ground models' capacity of visual modeling, apart from language modeling.
3. When being applied on different datasets, the bound offers insight on the intrinsic difficulty and blessing of them. This could serve as a general guideline when designing new algorithms and developing new models.
4. In the effort of improving performance towards the upper bounds, we investigate the case where visual concept may not be realistically predicted with 100% accuracy and demonstrate a trade-off between its quantity and accuracy.

2 RELATED WORK

2.1 IMAGE 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 such as Kuznetsova et al. (2012); Mitchell et al. (2012); Kulkarni et al. (2013). Another popular approach is based on caption retrieval in the embedding space such as Kiros et al. (2014); Devlin et al. (2015b). Most recently, the use of language models conditioned on visual inputs have been widely studied in the work of Fang et al. (2015) where a maximum entropy language model is used and Donahue et al. (2015); Vinyals et al. (2014); Xu et al. (2015b); Mao et al. (2015); Karpathy & Fei-Fei (2014) where recurrent neural network based models are built to generate natural language descriptions. The work of Devlin et al. (2015a) advocates to combine both types of language models. Furthermore, CIDEr (Vedantam et al., 2015) 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. (2015) suggests a simple sampling algorithm during training, which was one of the winning recipes for MSR-COCO Captioning challenge [1], and Jia et al. (2015) suggests the use of extra semantic information to guide the language generation process.

[1] http://mscoco.org

2.2 VIDEO CAPTIONING

Similarly, video captioning has made substantial progress recently. Early models such as Kojima et al. (2002); Barbu et al. (2012); Rohrbach et al. (2013) 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 Venugopalan et al. (2015b); Donahue et al. (2015); Yao et al. (2015); Venugopalan et al. (2015a); Xu et al. (2015a); Rohrbach et al. (2015a); Yu et al. (2015) and most recently Ballas et al. (2016) have adopted a 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. (2013), the other two constructed based on Descriptive Video Service (DVS) on movies by Torabi et al. (2015) and Rohrbach et al. (2015b). The latter two have recently been combined together as the official dataset for Large Scale Movie Description Challenge (LSMDC) [2].

2.3 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. (2015) in the context of image captioning and in Rohrbach et al. (2015a) 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. (2015a) 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 upper bounds 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. (2015a) that performs caption retrieval.

3 OVERALL FORMULATION OF THE EMPIRICAL UPPER BOUNDS

The upper bound is inspired by the following observation that

$$P(\mathbf{w}|\mathbf{v}) = \sum_{\mathbf{a}} P_\theta(\mathbf{w}|\mathbf{a})P(\mathbf{a}|\mathbf{v}) \tag{1}$$

where $\mathbf{w} = \{w_1, \ldots, w_t\}$ 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 upper bound, we assume that the atom model is given as an oracle. This amounts to treat $P(\mathbf{a}|\mathbf{v})$ as a Dirac delta function that assigns all the probability mass to the observed atom $\mathbf{a}$. Therefore, Equ. (1) is simplified as

$$P(\mathbf{w}|\mathbf{v}) = P_\theta(\mathbf{w}|\mathbf{a}) \tag{2}$$

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 upper bound of it. One could vary the amount of atoms given to Equ. (2). 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.

[2] `https://sites.google.com/site/describingmovies/`

3.1 Constructing atoms from captions

In principal, each configuration of $\mathbf{a}$ may be associated with a different distribution $P_\theta(\mathbf{w}|\mathbf{a})$, therefore a different performance upper bound. We define configuration as an orderless collection of unique atoms. That is, $\mathbf{a}^{(k)} = \{a_1, \ldots, a_k\}$ where k 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.

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 6.4.2. Being able to capture these important concepts is considered as crucial in getting superior performance. Therefore, comparing the performance of existing models against this upper bound reveals their ability of capturing atoms from visual inputs when $P(\mathbf{a}|\mathbf{v})$ is unknown.

3.2 Analytical form of the empirical upper bound

Given a bag of atoms $\mathbf{a}^{(k)}$, and captions $\mathbf{w}$, the upper bound is written as

$$\mathcal{U}_k(\theta) = \log P_\theta(\mathbf{w}|\mathbf{a}^{(k)}) = \log \prod_{t=1}^{T} P(\mathbf{w}_t|\mathbf{w}_{<t}, \mathbf{a}^{(k)}) = \sum_{t=1}^{T} \log P(\mathbf{w}_t|\mathbf{w}_{<t}, \mathbf{a}^{(k)}) \tag{3}$$

Note that for different choice of atoms $\mathbf{a}^{(k)}$, one would obtain a different upper bound $\mathcal{U}_k$.

4 Parameterization

4.1 Model $P_\theta(\mathbf{w}|\mathbf{a}^{(k)})$

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 (Cho et al., 2014) to model this conditional based on the following simple factorization

$$P_\theta(\mathbf{w}|\mathbf{a}^{(k)})) = P_\theta(\mathbf{w}_1|\mathbf{a}^{(k)}))P_\theta(\mathbf{w}_2|\mathbf{w}_1, \mathbf{a}^{(k)}))\ldots P_\theta(\mathbf{w}_t|\mathbf{w}_{<t}, \mathbf{a}^{(k)})) \tag{4}$$

Recurrent neural networks (RNNs) are natural choices when outputs are identified as sequences. The overall model can be written as

$$\begin{bmatrix} p(\mathbf{w}_t \mid \mathbf{w}_{<t}, \mathbf{a}^{(k)})) \\ \mathbf{h}_t \end{bmatrix} = \psi(\mathbf{h}_{t-1}, \mathbf{w}_{t-1}, \mathbf{a}^{(k)})) \tag{5}$$

where $\mathbf{h}_t$ represents the RNN state at timestep t. Vanilla RNNs, however, encountered serious issues when used to model long term dependencies. We hence borrow the recent success from a variant of RNNs called Long-short term memory networks (LSTMs) first introduced in Hochreiter & Schmidhuber (1997), formulated as the following

$$\begin{bmatrix} p(\mathbf{w}_t \mid \mathbf{w}_{<t}, \mathbf{a}^{(k)})) \\ \mathbf{h}_t \\ \mathbf{c}_t \end{bmatrix} = \psi(\mathbf{h}_{t-1}, \mathbf{c}_{t-1}, \mathbf{w}_{t-1}, \mathbf{a}^{(k)})). \tag{6}$$

where compared with Equ. (5), an extra $\mathbf{c}_t$ is used to denote the memory state of LSTMs at timestep t.

4.2 Model atoms as a bag of words

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 (Bengio et al., 2003), 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 $\Phi(\mathbf{a}^{(k)}) = \sum_{i=1}^{k} \mathbf{E}_a[\mathbf{a}_i^{(k)}]$.

4.3 LSTM PARAMETERIZATIONS

Combined with the bag of words representation of atoms, an LSTM parameterizes $\psi(\mathbf{h}_{t-1}, \mathbf{c}_{t-1}, \mathbf{w}_{t-1}, \mathbf{a}^{(k)})$ in Equ. (6) as following

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

where $\mathbf{E}_w$ denotes the word embedding matrix, as apposed to the atom embedding matrix $\mathbf{E}_a$ in Section 4.2. $\mathbf{W}$, $\mathbf{U}$, $\mathbf{A}$ 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

$$
\mathbf{p}_t = \mathrm{softmax}(\mathbf{U}_p \tanh(\mathbf{W}_p \mathbf{h}_t + \mathbf{b}_p) + \mathbf{d}), \tag{7}
$$

with parameters $\mathbf{U}_p$, $\mathbf{W}_p$, $\mathbf{b}_p$ and $\mathbf{d}$.

4.4 OVERALL TRAINING CRITERION

Based on Equ (3), the overall training criterion is as the following

$$
\theta = \arg\max_{\theta} \mathcal{U}_k(\theta) = \log \prod_{n=1}^{N} P_\theta(\mathbf{w}^{(n)} | \mathbf{a}^{(n,k)}) = \sum_{n=1}^{N} \sum_{t=1}^{T} \log P_\theta(\mathbf{w}_t^{(n)} | \mathbf{w}_{<t}^{(n)}, \mathbf{a}^{(n,k)}) \tag{8}
$$

given N training pairs $(\mathbf{w}^{(n)}, \mathbf{a}^{(n,k)})$. θ represents parameters in the LSTM, discussed in Section 4.3. Intuitively, training amounts to maximizing the proposed upper bound given ground truth captions and atoms extracted from them. Section 6.2 provides details of atom extraction.

5 IMPLICATION AND CONSTRAINT OF THE EMPIRICAL UPPER BOUNDS

The formulation of Section 3 is generic, only relying on the assumption the two-step visual captioning process, independent of the parameterization in Section 4. In practice, however, one needs to take into account several contributing factors to the trained upper bounds.

- 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 6.2.
- 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 6.4.2.
- The modeling capacity of the conditional language $P_\theta(\mathbf{w}|\mathbf{a}^{(k)})$ has a direct influence on the obtained upper bounds. Section 4 has shown one example of many possible parameterizations.
- The upper bounds may be sensitive to the training procedure and its hyper-parameters (see Section 6.3).

It is therefore important to keep in mind that the empirically obtained upper bound is *conditioned* on the above factors. Quite surprisingly, however, with the simplest procedure and parameterization adopted in this work, we show in the experimental section that the obtained upper bounds serve their purpose reasonably well.

6 Experiments

We demonstrate the procedure of computing the proposed upper bound on some standard datasets. For image captioning, we choose COCO, the largest one available to date. For video captioning, we choose YouTube2Text and LSMDC, both widely used for evaluating model performance.

6.1 Datasets

MS COCO (Lin et al., 2014) is the most commonly used benchmark dataset in image captioning. It consists of 82,783 training and 40,504 validation images, much larger than Flickr8k and Flickr30k. Each image is accompanied by 5 captions, all in one sentence. We follow the split used in Xu et al. (2015b) 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. (2015), it is split into 1,200 clips for training, 100 for validation and 670 for testing.

Another two datasets have been recently introduced in Torabi et al. (2015) and Rohrbach et al. (2015b). Compared with YouTube2Text, they are both much larger in the number of video clips. As they are created based with semi-automatically transcribed descriptive video description service (DVS), most clips are associated with one or two captions. There has been a recent effort to merge two datasets together for Large Scale Movie Description Challenge (LSMDC). [3] We therefore call this particular dataset LSMDC. The official splits contain 91,908 clips for training, 6,542 for validation and 10,053 for testing.

6.2 Atom extraction

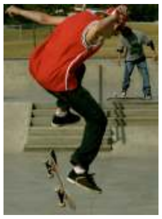
visual input	caption	entity	action	attribute
	A blue train sitting along side of a green forest.	train, side, forest	sit	blue, green
	Old train left out on the ground has graffiti all over it	train, it, ground	leave, graffiti, have	old
	Two abandoned blue and white train cars next to trees.	car, train, 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.

As previously mentioned in Section 3.1, visual concepts in images and videos are summarized as atoms that are provided to the caption language model. For the purpose of automation we have argued in Section 3.1, to some extent, that atoms could be represented by words related to three categories: actions, entities, and attributes. To identify these three classes, we utilize the publicly available Stanford natural language parser [4] to automatically extract them. After a caption is parsed,

[3] `https://sites.google.com/site/describingmovies/`

[4] `http://nlp.stanford.edu/software/lex-parser.shtml`

we apply simple heuristics based on the tags produced by the parser, ignoring the phrase and sentence level tags [5]:

- words tagged with {"NN", "NNP", "NNPS" ,"NNS", "PRP"} as entity atoms
- words tagged with {"VB", "VBD", "VBG", "VBN", "VBP", "VBZ"} as action atoms.
- 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]. The result of such a procedure is illustrated in Figure 1.

Following the above procedure, 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 upper bounds, 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 6.3 below.

6.3 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 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 upper bounds are maximized based on the training criterion discussed in Section 4.4.

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 (Hinton et al., 2012) to regularize the models with different size. In particular, we experimented with random hyper-parameter search by Bergstra & Bengio (2012) 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 (Zeiler, 2012) 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.

6.4 Interpretation

All three metrics – BLEU, METEOR and CIDER are computed with Microsoft COCO Evaluation Server (Chen et al., 2015). Figure 2 summarizes the learned upper bounds with an increasing number of k. We make several observations based on the figure.

6.4.1 Bounding performance of existing models

As an upper bound, one could compare the current state-of-the-art models' performance against the established bounds in Figure 2. Table 1 shows the comparison with the established upper bounds on three different datasets. Without such a bound, it would have been difficult to draw any conclusion about the models' capacity of capturing visual concept. With Figure 2, however, one could easily associate a particular performance with the equivalent number of atoms perfectly captured, across all 3 atom categories, as illustrated in Table 1, including the upper bounds 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".

[5] For a complete list of tags and their meaning, refer to https://web.archive.org/web/20130517134339/http://bulba.sdsu.edu/jeanette/thesis/PennTags.html

[6] http://www.nltk.org/

[7] The final set of atoms is available at git@github.com:yaoli/atoms.git

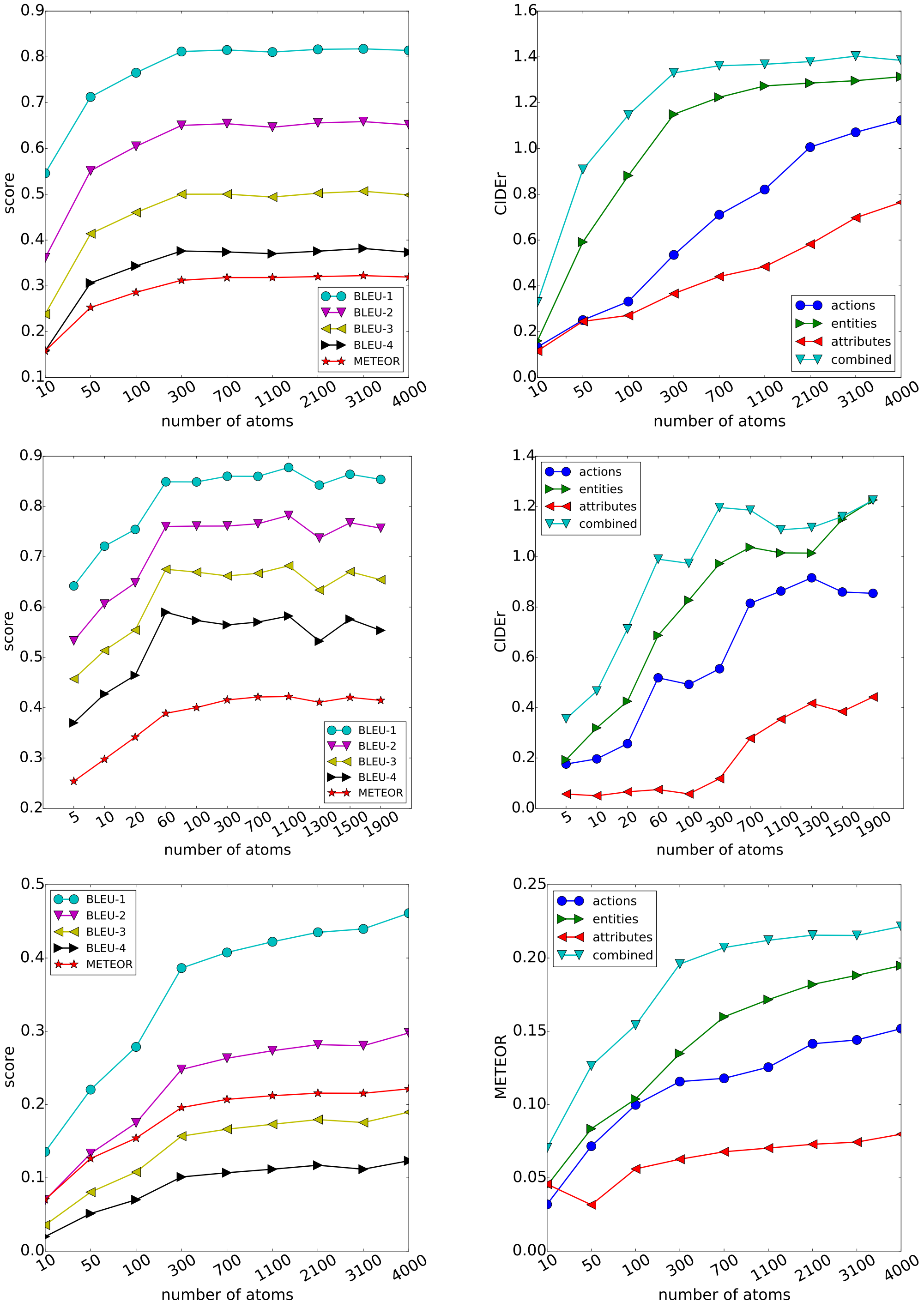


Figure 2: Learned upper bounds on COCO (top), YouTube2Text (middle) and LSMDC (bottom). The number of atoms $\mathbf{a}^{(k)}$ is varied on x-axis and upper bounds are computed on y-axis on testsets. The first column shows the bounds on BLEU and METEOR with $3k$ atoms, k from each of the three categories. The second column shows the bounds when k atoms are selected individually for each category. CIDEr is used for COCO and YouTube2Text on column 2 as each test example is associated with multiple ground truth captions, the case where CIDEr is deemed suitable (Vedantam et al., 2015). For the second column of LSMDC, the more suitable METEOR is used, as argued by Rohrbach et al. (2015a).

Table 1: 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 upper bound 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, in theory, according to Vedantam et al. (2015). The learned upper bound obtained in this work with respect to different metric is denoted in **bold**.

	B1	B4	M	C	ENT	ACT	ATT	ALL
COCO (Qi Wu et al., 2015)	0.74 **0.80**	0.31 **0.35**	0.26 **0.30**	0.94 **1.4**	$\sim$200	$\sim$2100	$>$4000	$\sim$ 50
YouTube2Text (Yu et al., 2015)	0.815 **0.88**	0.499 **0.58**	0.326 **0.40**	0.658 **1.2**	$\sim$60	$\sim$500	$>$1900	$\sim$ 20
LSMDC (Venugopalan et al., 2015a)	N/A **0.45**	N/A **0.12**	0.07 **0.22**	N/A **N/A**	$\sim$40	$\sim$50	$\sim$4000	$\sim$10

6.4.2 IDENTIFYING VISUAL ATOMS BY HUMAN EVALUATION

As discussed in Section 3.1, 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 1 where current state-of-the-art models have the equivalent capacity of capturing perfectly less than 100 atoms from each of three categories. Three people are asked to cast their vote independently on the top 150 extracted atoms for each category for all three datasets. The final decision of an atom being visual or not is made by majority vote. Table 2 shows the ratio of atoms that are flagged visual by this procedure.

Table 2: 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 3.1. 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%

6.4.3 ATOM ACCURACY VERSUS ATOM QUANTITY

So far we have assumed that the atoms are given, or in other words, the prediction accuracy of atoms is 100%. This is the case where one would like to upper bound the model performance. 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 $\mathbf{a}^{(k)}$ in Equ. (3) 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 or by keeping the accuracy while increasing the number of included atoms.

6.4.4 THE DIMINISHING RETURN

As the number of atoms k in $\mathbf{a}^{(k)}$ increases, one would expect the upper bounds to be improved accordingly. It is however not yet clear the speed of such improvement. In other words, the gain in

[8] The corresponding results are available at `git@github.com:yaoli/atoms.git`

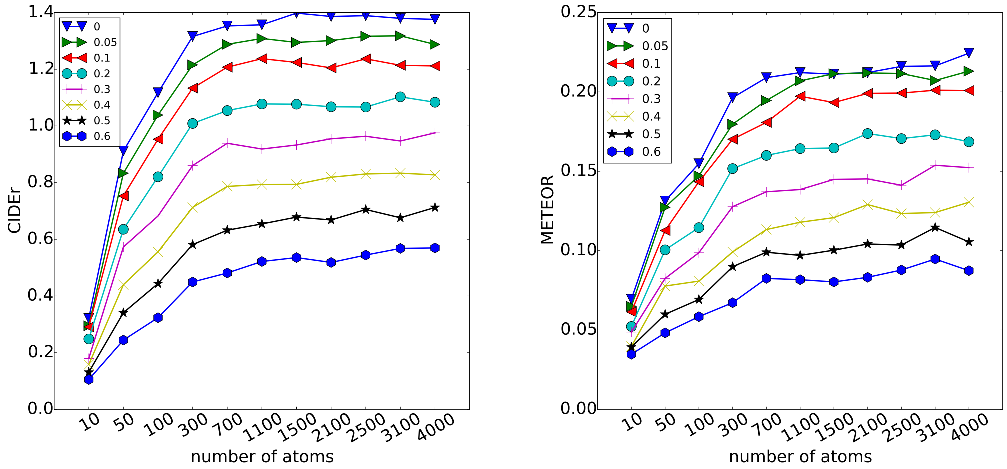


Figure 3: Learned upper bounds with various atom precision on COCO (left) and LSMDC (right). The number of atoms $\hat{\mathbf{a}}_r^{(k)}$ is varied on x-axis and upper bounds are computed on y-axis on testsets. CIDEr is used for COCO and METEOR for LSMDC. Compared with Figure 2, the upper bounds are obtained with various atom accuracy (r in legend) and atom quantity (x-axis). There is a trade-off between atom accuracy and atom quantity. 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.

performance may not be proportional to the number of atoms given when generating captions. Based on the Figure 2, the upper bounds on all three datasets show a significant gain at the beginning and such gain diminishes quickly as more and more atoms are used.

On the other hand, it is possible to further tighten the upper bounds obtained in Figure 2. Although visual atoms dominant the three atom categories shown in Section 6.4.2, as they increase in number, it is possible that 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 in images and videos. Excluding non-visual atoms in the conditional language model can further tighten the upper 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.

6.4.5 RANKING THE IMPORTANCE OF THREE ATOM CATEGORIES

Column 2 of Figure 2 highlights the difference among actions, entities and attributes in generating captions. For all three datasets tested, entities played much more important roles, even more so than action atoms in general. This is particularly true on LSMDC where the gain of modeling attributes is much less than modeling the other two categories.

6.4.6 DIFFICULTIES AND BLESSINGS OF PARTICULAR DATASETS

Figure 2 also reveals the intrinsic properties of each dataset. In general, the bounds on YouTube2Text are much higher than COCO, and bounds on LSMDC are the lowest compare with the rest. 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 bounds, when compared across different datasets, offer an objective measure of difficulties and blessings of using them in the captioning task.

7 Discussion

This work formulates performance upper bounds for image and video captioning. The upper bound is constructed based on 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, all done without extra labeling or costly human evaluation besides the already given image/video and caption pairs.

Such a bound 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, their difficulty and blessing. Furthermore, we 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). With recent success of achieving super-human performance on ImageNet (for instance in He et al. (2015)), it would make sense to follow the two-step hypothesis to first consider building an atom classifier with reasonable accuracy. Once properly obtained, it is relatively easier to turn visual atoms into a natural language description. Such a two-stage model is likely to yield better performance than those trained end-to-end on visual captioning tasks, with a clear disentanglment of contributing factors.

Despite of its effectiveness shown in the experiments, the empirical upper bound is constructed with the simplest atom extraction procedure and model parameterization in mind, which makes such a construction in a sense a “lower bound” of the upper bound.

The bounds are measured on the most commonly used automatic evaluation metrics that have been subjected to discussion regarding their closeness to human performance. This might be the case in COCO image captioning, as observed in the recent MS-COCO captioning competition where several models exibited super-human performance. We argue, however, that this may due to the specificity of the COCO dataset (e.g., large overlap of training and test captions). This does not sufficiently invalidate the usage of automatic metrics in video captioning. Furthermore, to our best knowledge, none of the existing system comes even closer to human performance, as the video captioning datasets are either too small (as in Youtube2Text) or too difficult (as in LSMDC).

It it also worth mentioning that there exist alternative ways to provide visual cues to the language model besides atom extraction used in this work. For instance, the bounding boxes and their identity could be used when available. This is indeed the case of MS-COCO, however not available for YouTube2Text and LSMDC. The effect of incorporating such hints is left for future work.

Acknowledgments

The authors would like to acknowledge the support of the following agencies for research funding and computing support: IBM T.J. Watson Research, NSERC, Calcul Québec, Compute Canada, the Canada Research Chairs and CIFAR. We would also like to thank the developers of Theano (Bergstra et al., 2010; Bastien et al., 2012) , for developing such a powerful tool for scientific computing.

References

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

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.

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

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

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

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

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2010.

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.

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.

George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):30–42, 2012.

Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *EACL Workshop*, 2014.

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*, 2015a.

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

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.

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.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv preprint arXiv:1502.01852*, 2015.

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.

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

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.

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

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

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

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

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.

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*, pp. 359–368. Association for Computational Linguistics, 2012.

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*, pp. 740–755. Springer, 2014.

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

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*, pp. 747–756. Association for Computational Linguistics, 2012.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

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.

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

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

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

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 01 2016.

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

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.

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

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

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

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

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

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*, 2015b.

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

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*.

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

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