Exploiting Scene Context for Image Captioning

Rakshith Shetty
Dept. of Computer Science
Aalto University
Espoo, Finland
rakshith.shetty@aalto.fi

Hamed R.-Tavakoli
Dept. of Computer Science
Aalto University
Espoo, Finland
hamed.r-tavakoli@aalto.fi

Jorma Laaksonen
Dept. of Computer Science
Aalto University
Espoo, Finland
jorma.laaksonen@aalto.fi

ABSTRACT

This paper presents a framework for image captioning by exploiting the scene context. To date, most of the captioning models have been relying on the combination of Convolutional Neural Networks (CNN) and the Long-Short Term Memory (LSTM) model, trained in an end-to-end fashion. Recently, there has been extensive research towards improving the language model and the CNN architecture, utilizing attention mechanisms, and improving the learning techniques in such systems. A less studied area is the contribution of the scene context in the captioning. In this work, we study the role of the scene context, consisting of the scene type and objects. To this end, we augment the CNN features with scene context features, including scene detectors, objects and their localization, and their combinations. We use the scene context features as an initialization feature at the zeroth time step in a LSTM model with deep residual connections. In subsequent time steps, the model, however, uses the original CNN features. The proposed language model, contrary to more conventional ones, thus has access to visual features through the whole process of sentence generation. We demonstrate that the scene context features affect the language formation and improve the captioning results in the proposed framework. We also report results from the Microsoft COCO benchmark, where our model achieves the state-of-the-art performance on the test set.

1. INTRODUCTION

Automatic image captioning is an interesting problem that aims to integrate computer vision and natural language modeling. Recently, it has been in the center of attention and is experiencing a rapid growth with the advent of large annotated datasets and the availability of the computational power required to handle such large amounts of visual and textual data. The recent approaches to image captioning often rely on deep Convolutional Neural Networks (CNN) and Long-Short Term Memory (LSTM) models as key ingredients of their pipeline. The CNN compresses an image into a feature vector to be inputted to an LSTM network that acts as a generative language model. There exist, however, various modifications in the pipeline, such as using different CNN architectures and attention mechanisms for regional feature extraction [43, 44].

A common consensus of belief is that the CNN features encode hierarchical properties that are reflecting the scene context, including objects, the scene type, etc. Even if this is true, it is hard for the language models to learn to utilize this hidden implicit information. Hence, there could be benefits from using explicit contextual features in conjunction with the CNN feature vectors. To prevent very high-dimensional feature vectors adversely influencing the training of the LSTM module, we present an architecture that permits exploiting both CNN features and contextual features, consisting of the scene type, object localizations and segments.

Our language model is based on a deep LSTM network with residual connections where, motivated by the influence of the scene context on human sentence formation, the contextual features are used to initialize the hidden state of the caption generating LSTM network. This initialization then guides the recurrent caption generation process and leads to better captions. We show that the captions generated with such a strategy are better than those generated with pure CNN features, signifying the role of the scene context.

2. RELATED WORK

Image caption generation (or, more generally, image description generation) techniques include a wide range of methods and models. They are categorized into three groups [2], including retrieval-based [13, 29, 17, 20], sentence generation [25, 10] and mixture models, which combine the two paradigms [40, 12, 9].

A recurrent neural network (RNN) language model and CNN features are used in [19], where a technique for aligning the part of caption to image regions is also proposed. Vinyals et al. [40] use CNN image features with an LSTM network as the language model. They propose to use the CNN features only at the zeroth time step of the language model. We, however, let the language model access the CNN features throughout the generation process and find it improving the performance.

In [23], instead of a recurrent network, a feed-forward network is employed to predict a word given previous words and CNN features. In a more novel approach, [22] proposes a two-stage encoder–decoder model. First, both the image feature vector and the word sequence are embedded into a

iV&L-MM'16, October 16 2016, Amsterdam, Netherlands
© 2016 ACM. ISBN 978-1-4503-4519-4/16/10...$15.00
DOI: http://dx.doi.org/10.1145/2983563.2983571

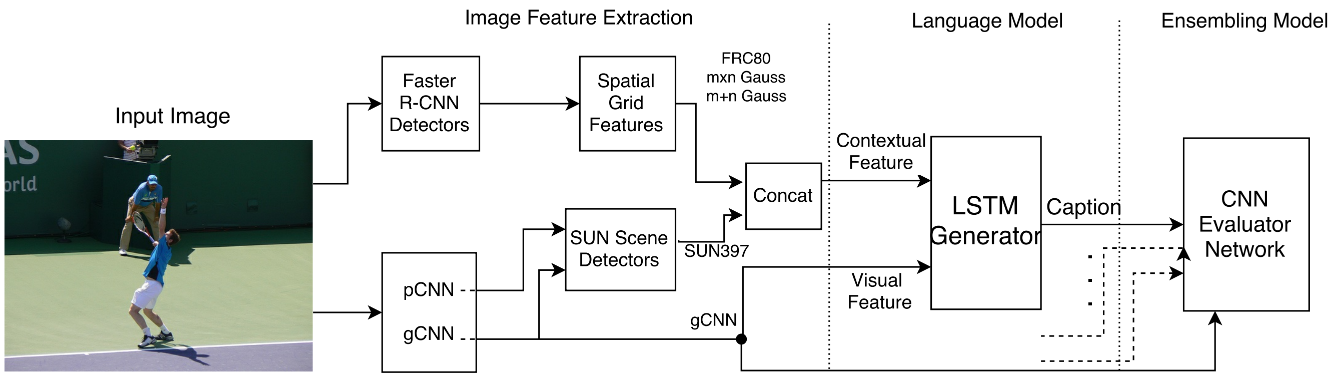


Figure 1: Block diagram showing our image captioning pipeline from an input image to the generated caption.

common multimodal space, where the image–caption pairs have similar representations. Then, given such an encoding, a decoder network generates the candidate caption. These models are trained end-to-end.

Contrary to the end-to-end paradigm, [12] takes a modular approach. It initially trains a set of object or concept detectors using multiple instance learning. Next, a maximum entropy language model utilizes the detector responses for captioning. Our proposed framework is trained end-to-end but shares some commonality with the aforementioned models such as using the CNN features. However, we exploit detector responses, akin to [12], as a part of the contextual features to boost the performance of the language model.

The image captioning models are evaluated on corpora of data consisting of image–description pairs, where the descriptions are provided by human annotators. One of the earliest well-recognized data sets is UIUC PASCAL sentences [32]. It consists of 1,000 images selected from the PASCAL–VOC data set [11] and five human-written sentences for each image. Recently, the same image set is used in PASCAL–50S [39], where 50 human-written sentences are provided. Flickr8k [17] and Flickr30k [45] are two large data sets, consisting of 8,000 images and 30,000 images, respectively. They have five human-written captions for each image. Currently, the most popular and largest dataset for image captioning, is the Microsoft Common Objects in Context (COCO) [27] with over 200,000 images and at least five human-written captions per image. There exists also an associated MS-COCO evaluation server, where researchers can upload their captions on the blind test dataset and compare the performance of their system to the state-of-the-art methods on a public leaderboard. Due to its size and availability of standardized benchmark, we choose to conduct all our experiments on the MS-COCO dataset.

There exists several evaluation metrics for caption assessment. Most of the the metrics are borrowed from machine translation. The three popular ones are the BLEU [30], ROUGE-L [26] and METEOR [6] scores. Another metric specifically designed for image captioning tasks called CIDEr was proposed in [39]. According to the COCO Image Captioning Challenge 2015 evaluations, the automatic evaluation metrics and the human assessments of caption quality do not yet match well, though METEOR and CIDEr were found to have the highest correlations with human judgments [4]. We, thus, put more weight on the performance of our models based on METEOR and CIDEr in the analysis.

In this paper, we will explore two less-studied aspects of image caption generation: 1) explicit scene context features for LSTM language models, and 2) caption picking techniques from an ensemble of caption generator models. To this end, we setup a baseline model and compare it with the proposed framework that augments the CNN features with contextual features. After that, we propose and evaluate a caption picking network. We report the benchmark results on the MS-COCO dataset and show that the proposed framework outperforms the state-of-the-art results.

3. METHOD

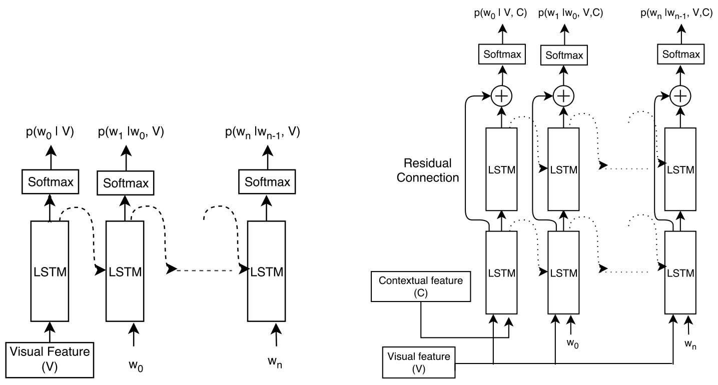
The overall pipeline of the proposed framework is illustrated in Figure 1. It consists of three stages: image encoding and feature extraction, language model and ensemble evaluator. We will elaborate on the details of our model in the rest of this section.

3.1 Image Feature Extraction

Good image representation is a key factor for successful image captioning. A representation should be compact and complete to encode the relevant information. Thinking about how humans construct a sentence, we can hypothesize the features should reflect the scene context, i.e., the information about objects and their attributes such as color, absolute and relative positions along with the scene type.

CNN Features. The activations of fully-connected layers of a pre-trained CNN have been shown to be useful features for image captioning [8, 34]. We use the GoogLeNet [37] architecture to extract CNN-based features. In the proposed framework, we utilize the CNN features for two different purposes. Features from a network pre-trained on ImageNet [5] are used as direct input to the language model, whereas features from a network pre-trained on MIT Places [47] are used to train our scene detector.

To extract features from the GoogLeNet pre-trained on ImageNet, we employ a reverse spatial pyramid pooling [14] unit on top of the *5th Inception module*. The pooling has two scales, where the second scale consists of 26 regions obtained from a 3×3 overlapping grid and its horizontal flipping. To extract the feature vector, we compute the response of each of the 26 regions and employ the combinations of center-crop, average- and maximum-pooling. The feature vector,



(a) Baseline (b) Proposed

Figure 2: The language model architectures. The LSTMs are unrolled in time. Specifically, a two-layer LSTM with residual connections is shown.

denoted as "gCNN", has the dimensionality 2048 and is used as a direct input to the language model as shown in Figure 1.

Features from the GoogLeNet pre-trained on MIT Places are extracted similarly, but this time the responses of the *3rd classification branch* are fed to the reverse spatial pyramid pooling unit. These features will be denoted as "pCNN" in the rest of the paper.

Scene Type. We use the scene type cue as a feature to the language model. To determine the scene type, we created a bank of specialized visual scene detectors by training a classifier on pCNN features extracted for the SUN Scene Categorization Benchmark database [41, 42]. For the classification purpose, akin to [24], we employ Radial Basis Function (RBF) Support Vector Machines (SVMs).

We use three variants each of pCNN and gCNN features, obtained with different pooling strategies, and train a separate classifier for each variant. Each classifier determines the degree of association of the image to the 397 scene types of the SUN database. The final scene type score is determined by the average fusion of the six classifiers. Thus, for an input image, we form a 397-dimensional feature vector consisting of these raw scene type scores in the range of $[0, 1]$, denoted as "SUN397". This feature vector is used as an input to the language model.

Object Type and Location. The type of objects and their locations affect sentence formation and influence adjectives used in human sentence constructs. To extract such information, we use an object detector network, more specifically the Faster Region-based Convolutional Neural Network (R-CNN) [33]. This network predicts the object locations in terms of bounding boxes and object detection scores of the 80 object categories of COCO 2014 [27]. To produce object location maps, we use the same object proposals generated from the Faster R-CNN network. We first define an $m \times n$ non-overlapping grid on the image where each of the cells, $F_c(i, j)$, accumulates the integral of Gaussian distributions fit to the object proposals of the class category c as

$$F_c(i,j) = \sum_{b_k \in BB(c)} \iint_{b_k \cap G(i,j)} p(b_k) N(\text{center}(b_k), \text{diag}(b_k)) \ , \quad (1)$$

where $BB(c)$ is the set containing bounding box object proposals for category c, $p(b_k)$ is the confidence assigned by the detector to proposal b_k, $G(i, j)$ is the grid cell at position (i, j) and $N(\mu, \sigma)$ are Gaussians of given mean μ and standard deviation σ. For each object category, this grid is computed and vectorized to produce a feature vector of dimensionality $m \times n$. The final feature vector is obtained by concatenating the category-wise vectors and denoted as "$m{\times}n$Gauss".

One disadvantage of the "$m{\times}n$Gauss" feature vector is that it can become large depending on the resolution of the grid. To address this we also experimented with using smaller number of spatial grids to encode location information. Instead of splitting an image into $m \times n$ non-overlapping grid cells, we split it into m vertical and n horizontal strips. In this case, the vertical and horizontal cells will overlap, but the number of cells is reduced from $m \times n$ to $m + n$. The calculation of these features is analogous to that of the regular grid-based ones, only the definition of the grid cells $G(i, j)$ is different. We abbreviate these features as "$m{+}n$Gauss" in the figures and result tables.

To reduce the feature vector size even further, an alternative is to discard the location information completely and just encode the object detection scores. We obtain such an 80-dimensional feature vector using the detection score of the Faster R-CNN responses by setting $m = n = 1$, and refer to it as "FRC80".

3.2 Language Model

The next stage in the pipeline is a conditional language model which takes as input the image features and generates a caption. The language model in the proposed framework is based on the Long-Short Term Memory network [16]. It computes the probability of a sentence S given an image I, denoted by $P(S|I)$:

$$P(S|I) = P(w_0, w_1, \cdots, w_n|I) \quad (2)$$

$$= p(w_0|I) \prod_{t=1}^{n} p(w_t|w_{t-1}, I) \ . \quad (3)$$

where w_i corresponds to the i-th word in a sentence. Since we are augmenting CNN features with contextual ones, we need two seperate feature input channels to the language model. This is contrary to conventional LSTM language models, such as our baseline model based on [40], which uses a single feature input channel. We thus modify the LSTM architecture, according to Figure 2b, in order to suit this purpose. In the following, we first explain the baseline language model. Later, we outline the modifications proposed for feature handling, followed by an extension for a deep model with residual connections.

Baseline Language Model. The baseline language model, depicted in Figure 2a, is consistent with the architecture of the language model of [40]. The model consists of an LSTM network with a softmax layer at its output, which produces the probability distribution over the model's vocabulary:

$$p(w_t|w_{t-1}, \cdots, w_0, V) = \text{softmax}(y(t)) \ , \qquad (4)$$

where V are the gCNN features, w_i corresponds to a word generated at the i-th time step and $y(i)$ is the output of the LSTM network at the i-th time step. The visual features are fed to the LSTM network at time step zero and share the input weight matrix with the word vectors, which are fed in subsequent time steps.

In the training, the LSTM learns to assign the highest probability to the next ground truth word, given the current inputs and the hidden state by minimizing

$$\mathcal{L}(w_{1\cdots L}|V) = -\sum_{t=1}^{n} \log(p(w_t|w_{t-1}, V)) \ . \qquad (5)$$

A caption is generated based on the visual features and previously generated words until reaching a stop symbol. The caption generation process can be seen as inferring the best next word given a partial sentence, which can be further enhanced using a beam search strategy. The beam search maintains a list of top b partial sentences instead of one partial sentence at a time. It then extends each partial sentence with the b most probable words at each time step, resulting in b^2 extensions, which are pruned back to the b most probable partial sentences.

Proposed Language Model. The language model we use should be capable of utilizing both the contextual and the visual features. Since the baseline model has only one input channel, shared between word vectors and image features, it can utilize multiple features only if they are fused into a single input vector. In our study, we found that performing simple feature fusion, like concatenating the two feature vectors, and using it as the input in the baseline language model leads to inferior performance, presumably due to large dimensions of the resulting feature vector.

Thus, to address this issue, we introduce another data input channel and make the visual features available throughout the whole inference process. This is similar to [35, 36], where we use such a language model for video captioning. The proposed framework thus computes the distribution

$$p(w_t|w_{t-1}, \cdots, w_0, V, C) = \text{softmax}(y(t)) \ , \qquad (6)$$

where C represents the contextual features. The training is done by minimizing the cost function

$$\mathcal{L}(w_{1\cdots L}|V, C) = -\sum_{t=1}^{n} \log(p(w_t|w_{t-1}, V, C)) \ . \qquad (7)$$

Contrary to the baseline model, the gCNN visual features are accessible throughout the whole sentence generation process. In the proposed language model, the contextual features share the same weight matrix with the words and are presented at the time step zero, while the gCNN features will have their own weight matrix. Thus this model has the ability to learn different functions from word embeddings and the CNN features, since they no longer share the weight matrix. By simultaneous learning from two different sources, the proposed architecture results in more powerful models.

Deeper Model with Residual Connections. We also extend our language model by using a deeper version consisting of multiple layers of stacked LSTMs and residual connections. The first layer receives visual and contextual features and the consequent layer receives its inputs from the corresponding bottom LSTM. We further add a series of residual connections [15] between the stacked LSTM layers. In our experiments, we noticed that the residual connections improve the convergence speed and produce significantly lower cost function values and model perplexity.

3.3 Ensemble of Language Models

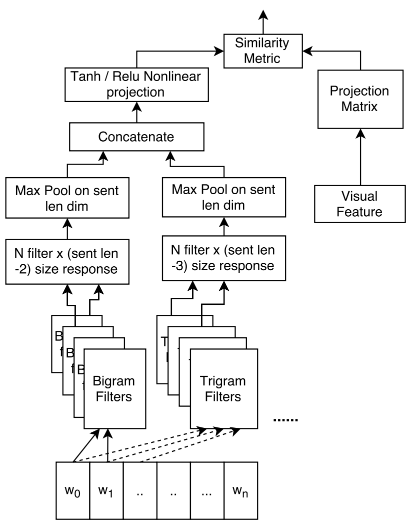


Figure 3: Caption evaluator: computing the similarity between visual features and a caption in a common space.

The key idea behind an ensemble of language models is boosting the captioning results by picking the best candidate caption from a set of candidates. For this purpose, we learn a set of different language models using the combination of various image features and LSTM architectures. Given an image, the set of language models generate a pool of captions, of which one is the best candidate. In order to pick the best candidate, we use a convolutional neural network. We train the network to match the image, represented by its gCNN features, with the best available caption for it. We choose this ensembling technique as our older unpublished experiments showed that simpler language model ensembling methods, such as averaging the multiple generators or choosing the caption that has the highest likelihood, only produced inferior results. A complete discussion of various ensembling techniques is beyond the scope of this work.

Caption Picking. The caption picking is performed by comparing the similarity of the gCNN features and encoded

Table 1: The performance of contextual features in comparison with the baseline model M1. The effect of various language model parameters, including depth and residual connections, “r”, are reported.

Model	Contextual Feature	parameters		validation	performance metrics			
		depth	r	perplexity	BLEU-4	METEOR	ROUGE-L	CIDEr
M1	—	1	—	10.82	0.259	0.222	0.490	0.750
M2	FRC80	1	—	10.15	0.316	0.249	0.534	0.952
M3	SUN397+FRC80	1	—	10.05	0.315	0.250	0.532	0.954
M4	4×4Gauss	1	—	10.15	0.308	0.246	0.527	0.921
M5	3+3Gauss	1	—	10.08	0.308	0.247	0.527	0.928
M6	SUN397+FRC80	2	no	10.14	0.318	0.252	0.535	0.967
M7	SUN397+FRC80	3	no	10.34	0.316	0.253	0.533	0.964
M8	SUN397+FRC80	2	yes	9.92	**0.320**	0.253	**0.536**	0.966
M9	SUN397+FRC80	3	yes	**9.69**	0.316	**0.254**	0.532	0.962
M10	3+3Gauss+SUN397+FRC80	3	yes	9.72	0.319	0.252	0.535	**0.970**

captions in a neural architecture, depicted in Figure 3.

To encode the captions, we employ a convolutional neural network operating on the words of a sentence. This resembles a sentence sentiment predictor network [21]. A caption is treated as a sequence of word vectors which are learned during training. Alternatively, one could also use off-the-shelf word embeddings like word2vec [28] or GloVe [31]. The convolutional neural network accepts such vector representations and produces an encoding of the sentence.

Before comparing the gCNN image features to this sentence encoding, the CNN features are projected into a common space with the sentence encoding by using a projection matrix. Then, the cosine similarity between the two encoded vectors is computed. The higher the similarity score is, the better the caption is.

To train the evaluator, we take a ground truth caption of an image as the positive sample, S^+, and k negative samples, $S_i^-, i = 1, \dots, k$, from the captions of the rest of the training images. We then use $-\log P(S^+|S^-, I)$ as the cost function for maximizing the score of the positive samples, where

$$P(S^+|S^-, I) = \frac{\exp(c(S^+, I))}{\exp(c(S^+, I)) + \sum_{i=1}^{k} \exp(c(S_i^-, I))} \quad (8)$$

and $c(S, I)$ represents the cosine similarity between the sentence candidate S and the image I. Note that the normalization terms force the score assigned to the negative captions to be minimized when we maximize $-\log P(S^+|S^-, I)$.

4. EXPERIMENTS AND RESULTS

We use the Microsoft COCO 2014 data set [27] in our experiments. It consists of more than 164,000 images, where each train and validation set image has five reference captions. There are 80 object class categories annotated. For the training of the language model, we pre-process the captions by tokenizing and removing the punctuations. The reference captions use 23,528 unique words and, after removing words occurring fewer than five times, we end up with a vocabulary of 8,790 words.

To evaluate the performance of the captioning framework, we report four standard evaluation metrics, BLEU-4, METEOR, ROUGE-L and CIDEr. The performance of the proposed framework is measured on both the validation and test sets using the above metrics. Additionally, we report our models' perplexities for the ground truth captions on the validation set.

We thoroughly study the contribution of various contextual features and compare the new language model to the baseline by using the validation data. We collate the proposed framework with the state-of-the-art entries of the MS-COCO leaderboard[1] and published results.

Implementation. We implemented the LSTM language model using Theano [1]. The CNN feature extraction and the Faster R-CNN models are based on the Caffe library [18]. In other words, the training of the language model is decoupled from the rest of the architecture. The Faster R-CNN object detector is trained for the 80 object class categories of COCO 2014.

The language model is trained using the stochastic gradient descent with RMSProp [38] and drop out [46]. The hidden size of the LSTM network is chosen to be 512 dimensions. In the test phase, we use beam search to sample captions from the language model with a beam size of $b = 5$.

The caption picking convolutional neural network was created by bi-, tri-, 4-, and 5-gram filters, where 100 filters of each type are used. It also uses a 100-dimensional word vector representation, which is learned from random initialization. We use $k = 50$ negative captions in the training.

4.1 Contextual Features

To understand the benefits from the proposed framework and contextual features, we compare them to the baseline model. Table 1 presents the results of using each contextual feature and their combinations on the validation set. It also includes information on some of the parameters such as the depth and existence of the residual connections “r” in the models. We report a total of ten models, identified as M1,. . .,M10, where M1 is the baseline model.

Comparing the models with contextual features to M1, we learn that the contextual features and the proposed architecture outperform the baseline model significantly. On the depth of 1, the proposed model achieves a CIDEr score of 0.954, which is 26% better than the baseline. Fine-tuning the depth and combining different features, the performance of the proposed framework improves further.

Contrasting the contextual features with each other, we see that the object category feature FRC80 is the most con-

[1]http://mscoco.org/dataset/#captions-leaderboard

Table 2: COCO 2014 test results. The scores are based on Microsoft COCO leaderboard. c# indicates the number of reference captions used in the evaluation. The models are sorted based on the CIDEr score as done on the leaderboard.

Leaderboard Name	**BLEU-4**		**METEOR**		**ROUGE-L**		**CIDEr**	
	c5	c40	c5	c40	c5	c40	c5	c40
AugmentCNNwithDet (M10)	0.315	0.597	0.251	0.340	0.531	**0.683**	**0.956**	**0.968**
—— (Ensemble M11)	0.310	0.596	0.250	0.338	0.529	0.681	0.948	0.961
ATT_VC [44]	**0.316**	0.599	0.250	0.335	**0.535**	0.682	0.943	0.958
—— (M9)	0.309	0.588	0.251	0.342	0.529	0.680	0.943	0.948
OriolVinyals [40]	0.309	0.587	**0.254**	**0.346**	0.530	0.682	0.943	0.946
MSR_Captivator [12]	0.308	**0.601**	0.248	0.339	0.526	0.680	0.931	0.937
Berkeley LRCN [7]	0.306	**0.585**	0.247	0.335	0.528	0.678	0.921	0.934
human [3]	0.217	0.471	0.252	0.335	0.484	0.626	0.854	0.910
Montreal/Toronto [43]	0.277	0.537	0.241	0.322	0.516	0.654	0.865	0.893

tributing feature. Nonetheless, the model achieving the best CIDEr score, M10, employs the combination of the scene type, object category and object location maps as features.

4.2 Ensemble of Models

Table 3: The ensemble effect. The performance metrics and vocabulary size, VOC, are reported.

Model	**VOC**	**performance metrics**	
		METEOR	**CIDEr**
M1 (Baseline)	513	0.222	0.750
M2	887	0.249	0.952
M3	962	0.250	0.954
M7	1093	0.253	0.964
M8	983	0.253	0.966
M9	1197	**0.254**	0.962
M10	1112	0.252	0.970
M11 (Ensemble)	1303	**0.254**	**0.978**

To evaluate the our caption picking mechanism, we build an ensemble using M3, M7, M9, M10, and two additional models obtained by combining SUN397 with the object maps of spatial grids of 3+3Gauss and 4×4Gauss, respectively. We compare the performance of the proposed ensemble model with each of its contributing models, the baseline and two of the best-performing models, M2 and M8.

The results are reported in Table 3, where we also show the size of the vocabulary of the language models. For brevity, we skip the BLEU and ROUGE-L metrics in this table. It is evident that the captions picked from the ensemble of models improve the validation set performance on all the four metrics.

Comparing the size of the vocabulary of the models, we learn that the baseline has the smallest vocabulary of 513 words, while the ensemble's vocabulary has the largest number of words, 1303. The vocabulary size signifies the variety of words used in the captions generated by a model. It is interesting that the ensemble model both has the most diverse vocabulary and outperforms all the other models, which indicates that the evaluator successfully utilizes the diversity in the ensemble. Figure 4 shows some sample captions generated by the ensemble model M11 and the model M10 on images from the validation set.

Table 4: Comparison of our models to the best published results on COCO 2014 validation set.

Model	BLEU-4	METEOR	CIDEr
M11	**0.320**	**0.254**	**0.978**
M10	0.319	0.252	0.970
ATT_VC [44]	0.304	0.243	–
Berkeley LRCN [7]	0.300	0.242	0.896
OriolVinyals [40]	0.277	0.233	0.855
MSR_Captivator [12]	0.257	0.236	–
Montreal/Toronto [43]	0.250	0.230	–

4.3 Benchmark

We compare the performance of the proposed model with several state-of-the-art models reported on the COCO 2014 captioning leaderboard and published results on the validation set. For this purpose, we submitted M9, M10 and the ensemble model M11 captions to the CodaLab portal. We compare our results with ATT_VC [44], MSR_Captivator [12], Berkeley LRCN [7], Montreal/Toronto [43], OriolVinyals [40], and human [3] reported in CodaLab. It is worth noting that the OriolVinyals model shares the same architecture with our baseline model.

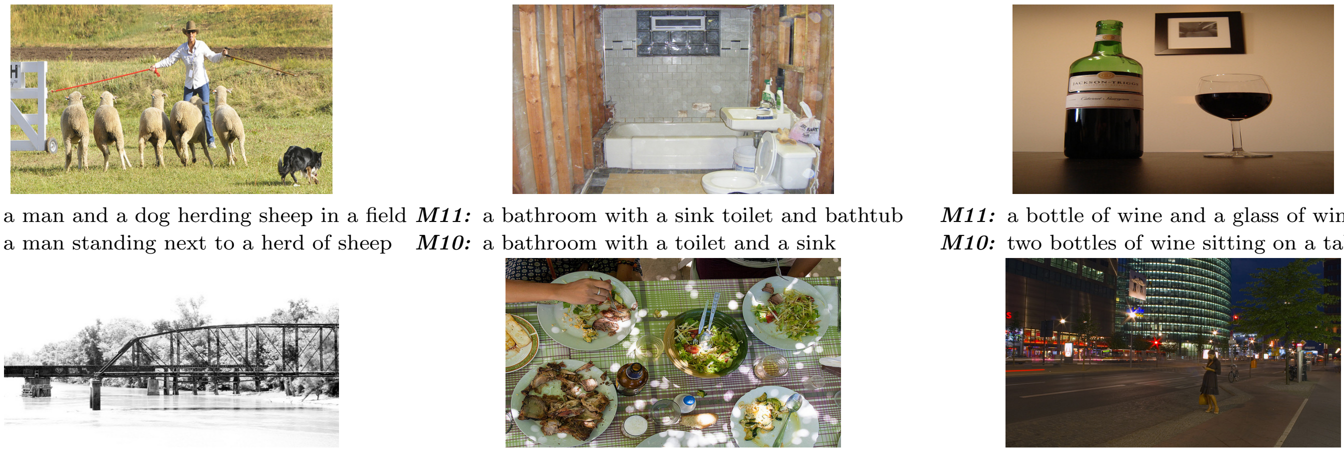
Table 2 reports the results of the benchmark[2]. While the ensemble model was the best model on the validation set, we observe that M10 is a better model on the test set. This could be because some models in the ensemble do not generalize well to the test set. Considering the overall performance of our model M10, we outperform several models and are on the top of the leaderboard as on 12-Apr-2016.

The scores in the CodaLab leaderboard are not necessarily reflecting the original published work results due to changes and updates. Thus, we also present a comparison with the published scores on the validation set in Table 4. We see that our models outperform all published results on the validation set, with the largest improvement seen in the CIDEr score. We omit the ROUGE-L metric from this table as only one other prior publication listed here reports it.

5. DISCUSSION AND CONCLUSION

We proposed an image captioning framework in which the language model is first initialized with contextual image fea-

[2]The leaderboard with more models and scores is at http://mscoco.org/dataset/#captions-leaderboard



M11: a man and a dog herding sheep in a field
M10: a man standing next to a herd of sheep

M11: a bathroom with a sink toilet and bathtub
M10: a bathroom with a toilet and a sink

M11: a bottle of wine and a glass of wine
M10: two bottles of wine sitting on a table

M11: a view of a bridge in the snow
M10: a train crossing a bridge over a river

M11: a table with plates of food on it
M10: a table topped with plates of food and drinks

M11: a person riding a bike down a city street
M10: a city street filled with lots of traffic

Figure 4: Captions generated for some validation images by two of our models. The first row contains samples where the ensemble model, M11, performs better, and the second row cases where M10 is better.

tures and then has access to the CNN features during the whole process of caption generation. The proposed framework outperformed the state-of-the-art on the MS-COCO captioning leaderboard.

From the experiments with various contextual features, we learned that the 80-dimensional object type feature is the most contributing feature. We also reported our experiments with a caption picking mechanism which uses an ensemble of caption generators and a neural network trained to choose the best caption among the candidates. While the ensemble model performed the best on the validation set, it fell slightly short of our best single model on the test set.

Despite the proposed model significantly improves over a baseline model and outperforms the state-of-the-art, it seems still failing to learn the fine-grained relationships between the objects in an image. For example, as depicted in Figure 4, the model cannot yet discriminate between a bicyclist and a person standing next to a bike. The proposed model also cannot count the objects correctly and is prone to repeated words instead of a numeric referral in some of the captions. For example, it sometimes generates "apple and apple" for referring to "two apples".

The vocabulary size of our ensemble model achieves 1/8 of that of the ground-truth captions. This can be interpreted as room for improvement in the sense of language enrichment. We hope to address these issues in the future work.

6. ACKNOWLEDGMENTS

This work was funded by the grant 284642 of the Academy of Finland and calculations performed with resources of the Aalto University School of Science "Science-IT" project.

7. REFERENCES

[1] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS Workshop, 2012.

[2] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, and B. Plank. Automatic description generation from images: A survey. *arXiv*, abs/1601.03896, 2016.

[3] X. Chen, T.-Y. L. Hao Fang, R. Vedantam, S. Gupta, P. DollÃąr, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv*, abs/1504.00325, 2015.

[4] Y. Cui, M. Ruggero Ronchi, and T.-Y. Lin. 1st captioning challenge slides. Large-scale Scene UNderstanding Workshop, 2015.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[6] M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *EACL*, 2014.

[7] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[8] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[9] D. Elliott and A. P. de Vries. Describing images using inferred visual dependency representations. In *ACL*, 2015.

[10] D. Elliott and F. Keller. Image description using visual dependency representations. In *EMNLP*, 2013.

[11] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015.

[12] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. Platt, L. Zitnick, and G. Zweig. From captions to visual concepts and back. In *CVPR*, 2015.

[13] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every

picture tells a story: Generating sentences from images. In *ECCV*, 2010.

[14] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. *arXiv*, abs/1403.1840, 2014.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[17] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *JAIR*, 2013.

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv*, abs/1408.5093, 2014.

[19] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[20] A. Karpathy, A. Joulin, and L. Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *NIPS*, 2014.

[21] Y. Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

[22] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In *ICML*, 2014.

[23] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv*, abs/1411.2539, 2014.

[24] M. Koskela and J. Laaksonen. Convolutional network features for scene recognition. In *ACMMM*, 2014.

[25] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi. Composing simple image descriptions using web-scale n-grams. In *CoNLL*, 2011.

[26] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In S. S. Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 2004.

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[29] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011.

[30] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *ACL*, 2002.

[31] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[32] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. Collecting image annotations using amazon's mechanical turk. In *NAACL HLT*, 2010.

[33] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *arXiv*, abs/1506.01497, 2015.

[34] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *CVPR Workshops*, 2014.

[35] R. Shetty and J. Laaksonen. Video captioning with recurrent networks based on frame-and video-level features and visual content classification. *ICCV Workshop on LSMDC*, abs/1512.02949, 2015.

[36] R. Shetty and J. Laaksonen. Frame- and segment-level features and candidate pool evaluation for video caption generation. In *ACMMM Multimedia Grand Challenge Solutions*, 2016.

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv*, abs/1409.4842, 2014.

[38] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *Coursera*: Neural Networks for Machine Learning, 2012.

[39] R. Vedantam, C. L. Zitnick, and D. Parikh. CIDEr: Consensus-based image description evaluation. In *CVPR*, 2015.

[40] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[41] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva. Sun database: Exploring a large collection of scene categories. *IJCV*, 2014.

[42] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[43] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv*, abs/1502.03044, 2015.

[44] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. *arXiv*, abs/1603.03925, 2016.

[45] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2014.

[46] W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. *arXiv*, abs/1409.2329, 2014.

[47] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014.