Comparing LDA and CCA as a Guide to Improve Automatic Image Description Systems

Wout Vekemans
KULeuven
w.vekemans@gmail.com

Thijs Dieltjens
KULeuven
thijsdieltjens@gmail.com

June 3, 2016

Abstract

We present a model that generates novel, fluent and grammatically correct image descriptions. Most of the current systems based on recurrent neural networks show the same flaws. The used words are too general and are not related to the picture. The generated sentences are often way shorter than the descriptions a human would provide. The most recent literature tries to solve this first problem by adding additional information into the recurrent neural network. This paper compares two sources of semantic information: Latent Dirichlet Analysis (LDA) and Canonical Correlation Analysis (CCA). We compare their effect on both recurrent neural networks (RNN) and long short-term memory networks (LSTM). Both techniques improve the results. Experiments show that both of these models are able to deal with noisy training data. Adding normalization to the search algorithm used to generate the sentences solves the second problem. A first form of normalization is able to improve results and creates longer sentences. A second normalization function leads to more creative sentences but makes more mistakes. The results of this paper are comparable to the literature, but do not outperform current attention based systems.

1. Introduction

Automatic image description is a complex problem. It combines elements from the domains of both Computer Vision and Natural Language Processing. A captioning system must be able to detect what is in the image, label these objects and combine them in a fluent, grammatically correct sentence. Figure 1 shows an example of a correctly described picture.

Figure 1: *Correctly described image*

The problem is drawing more and more attention from the industry. Internet giant Facebook [7], for example, is implementing it in their application to aid the visually impaired.

The currently best performing systems contain a convolutional neural network (CNN) that creates an image representation. A recurrent neural network based language model uses this representation as input to generate a description. An analysis of existing systems shows that the same mistakes occur in many systems. The generated captions are too short and they do not relate to the image. Jia et al. [12] suggest that this is caused by two opposing forces. The generated caption has to comply with the language model but also has to describe the given image.

We deal with this problem in two ways. The first addition focuses on the lack of relation between images and their generated description. Adding semantic information seems a viable way to guide the generation in the right direction. This information is extracted from the images in two ways. In the first way a model extracts topic distributions with Latent Dirichlet Allocation [2]. A second approach uses Canonical Correlation Analysis to project the unseen images into a multimodal image-sentence space. Both methods produce an additional vector which acts as a guide in the language model. Our contribution also compares the resistance to noise

in the training data of the two methods.

A second proposed addition normalizes the beam search algorithm that generates sentences based on predicted word probability distributions. Based on Jia et al. [12] we implement their best scoring Gaussian normalization function. They claim this results in longer sentences and achieves better scores on BLEU and Meteor. We introduce a new normalization function based on word frequency in the training set. This leads to more creative and longer sentences.

The rest of this paper is as follows. First we provide an overview of the most recent works that solve the image description problem in section 2. After that, section 3 elaborates on the extraction of semantic information from the images and how this information is added to existing systems. Section 4 provides a detailed overview of the conducted experiments and their results.

2. Related Work

This section provides an overview of previously proposed image description systems. A first type of older systems follows a transfer-based approach. This approach searches for visually similar images and transforms their captions to a sentence describing the image [4, 11, 18, 23]. This approach generally produces results that are less natural and creative, and often requires manually constructed rules.

A second type of systems extracts visual features such as object and scene detections and uses them to fill in predefined templates [8, 21, 29]. These models produce new sentences but do not allow for rich enough captions. Because of the use of manually constructed rigid templates the generated descriptions also feel less natural.

Most recently however the task of describing images is often seen as a "translation" from the image to the target language. Methods from machine translation are therefore transferred to this problem. The global structure of the system often follows an encoder-decoder framework. The best performing works use a convolutional neural network to represent image features. Most methods use the values of the top layer of a pre-trained convolutional neural network as the image representation [5, 20, 15, 27]. Other methods use models such as R-CNN [9] to generate representations for the most important regions of the image [14, 22]. The image representation then serves as an input for a language model. Generally two types of language models are proposed. The first type are entropy based language models that also try to couple parts of the image to certain words [19, 22]. Most models however use a fully statistical language model based on neural networks [17]. Mao et al. [20] and Karpathy et al. [15] improve results by using recurrent neural networks which are more fit for learning sequences. Vinyals et al. [27] and Donahue et al. [5] propose LSTM which is an extension of the standard RNN. All the language models produce a probability distribution over the words in the vocabulary. A beam search algorithm is able to produce a final sentence.

Some of the recent papers try to add additional information to the language model. Jia et al. [12] notice that two opposing forces affect the sentence generation. On the one hand the sentence needs to describe the image. On the other hand the sentence needs to fit the language model. Because of these forces, semantic drift occurs after the generation of a few words. To counter this effect they propose a guided LSTM (gLSTM) where a semantic guide directs the sentence generation towards the content of the image. Their best performing model uses the image projection learned with canonical correlation analysis as a guide. Similarly, Jin et al. [14] add a scene vector based on Latent Dirichlet Allocation [2] to their LSTM. The models that currently achieve the best results include visual attention in the LSTM model. This way the model learns where to look in an image [14, 28]. One major drawback of attention models is the need for sampling during both training and testing. This makes these models more complicated.

Jia et al. [12] also point out that the beam search algorithm favors shorter sentences. Therefore they propose a normalization function which punishes shorter descriptions during sentence generation.

3. Methodology

Our implementation is based on an existing system, implemented by Karpathy, that is freely available on his GitHub page[1]. It implements two systems, an RNN-based model described by Karpathy [15] and an LSTM system proposed by Vinyals [27]. Both of these implementations process the images using VGGNet [26]. The output of the last fully connected layer forms an image representation. This image vector is then fed into the RNN-based language

[1] https://github.com/karpathy/neuraltalk

model which predicts a probability distribution over all words at each time step. The final descriptions are generated using a beam search algorithm.

An analysis of the captions generated by these systems shows that the sentences become less related to the image as they get longer. Moreover the generated sentences often already occur in the training set and are far from unique. To deal with this, this work proposes two extensions. First, following Jia et al. [12], it suggests adding semantic information to the existing systems. We suggest to extract this with LDA or CCA. A second addition normalizes the beam search process. Since beam search favors shorter sentences, this normalization function needs to create longer descriptions. We implement a Gaussian function as proposed in [12] and define a new normalization function that focuses on generating more creative sentences.

This section elaborates on how the semantic information can be extracted from the images. It then shows how to add it to the neural networks. It also provides insight in the used normalization functions.

3.1. LDA-Based Information

The extraction of topics from the images seems a promising source of additional information [14]. One of the most widely used topic models is LDA [2], a generative probabilistic model for discrete data. It is mostly used to model topic distributions in corpora consisting of text documents, but we implement it to predict topic distributions for unseen images.

To extract topics, we first train an LDA model on the sentences of the training set. Based on this model, the topic distributions of the validation sentences can be inferred. To predict the topic distribution of an unseen image at test time we use a simple neural network using one hidden layer with 256 neurons and a softmax function. The network is trained with pairs of images and topic distributions from the training set and tuned on the validation set. Figure 2 shows the process of this prediction. Appendix A contains a detailed overview of the selection of the ideal amount of topics and some results of the network prediction.

3.2. CCA-Based Information

Since Jia et al. [12] propose CCA as a method to extract semantic information from images, we follow their approach to compare the performance of CCA with our own LDA additions. CCA focuses in finding correlations between the image and sentence representations. The projection is based on the image vectors computed with VGGNet and a term frequency-inverse document frequency (tf-idf) weighted vector representation for the descriptions. We use the CCA projection of the image with 256 dimensions as experiments show this performs best.

All sentences are considered on their own, to make sure all different descriptions are maximally correlated to the corresponding image. The LDA approach described above considers the five sentences as a whole, because the goal is to learn a topic distribution based on the image. All five descriptions should thus be considered as sampled from the same topic distribution.

3.3. Adding Information to RNN

The LDA topic distribution can guide the RNN generation process. The topic distribution L, predicted by the neural network described in section 3.1 is integrated using formulas (1)-(3). The elements in red are our contributions to the original formulas proposed by Karpathy et al. [15]. $W_{hi}, W_{hx}, W_{hh}, W_l, b_h$ and b_v are parameters to be learned by the network. x_t and y_t are the in- and output at time t. h_t are the values of the hidden layer at time t. $CNN_{\theta_c}(I)$ is the output of VGGNet given image I. f is an activation function.

$$b_v = W_{hi}[CNN_{\theta_c}(I)] \quad (1)$$

$$h_t = f(W_{hx}x_t + W_{hh}h_{t-1} + b_h + b_v + W_l L) \quad (2)$$

$$y_t = Softmax(W_{oh}h_t + b_o) \quad (3)$$

3.4. Adding Information to LSTM

The addition of semantic information to the LSTM network of Vinyals et al. [27] follows the gLSTM approach of Jia et al. [12]. They propose four different guides of which the CCA projections produce the best results. We experiment with both LDA and CCA based vectors containing semantic information.

3.5. Normalizing Beam Search

Since the beam search implementation focuses on maximizing the log probability of the generated sentence, the algorithm favors shorter sentences. To

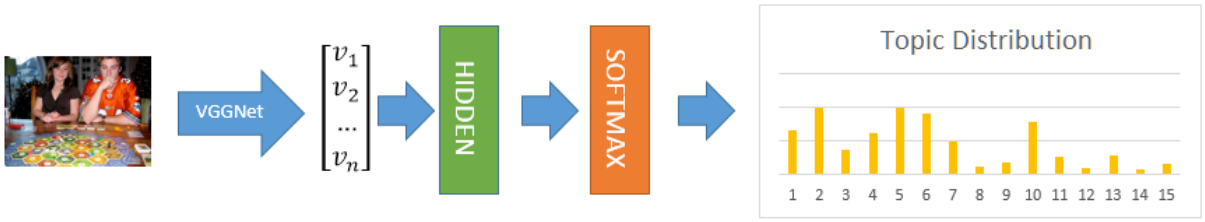

Figure 2: *Overview of LDA topic distribution prediction for an unseen image*

cope with this, Jia et al. [12] propose a normalization function. They propose a Gaussian function based on the sentence length distribution of the training set. We add the use of idf weights as another possible normalization. Formula (4) shows how this function $\Omega(\ell)$ is added to the beam search probability. ℓ is the length of the current word sequence. x_i is the ith word in the sequence, I is the image and ζ represents the parameters of the language model.

$$p = \frac{1}{\Omega(\ell)} \sum_{i=1}^{\ell} \log p(x_i | I, x_{1:i}, \zeta) \tag{4}$$

The Gauss normalization implements the normalization as the value of the Gaussian distribution based on the sentence lengths of the training corpus. This leads to $\Omega(\ell) \sim \mathcal{N}(\mu, \sigma)$ with μ and σ the average and standard deviation of the sentence length. This forces the length of the generated sentences to be similar to the length of the training sentences.

Our own contribution focuses on the quality of the generated sentences. These sentences seem to prefer vague terms, leading to higher BLEU scores and sentences that are often too general. To solve this, we implement the Ω function based on the idf weights of the individual words in the training set. Summing these weights over all words in a sentence gives an estimate of how much "information" the sentence contains. Doing so, sentences that use only frequent words are punished. This normalization also gives higher probabilities to longer sentences. Formula (5) contains the exact calculation of this score. N is the total number of training images, and n_i is the number of training sentences the ith word of the generated sentence occurs in.

$$\Omega(\ell) = \sum_{i=1}^{\ell} \log(\frac{N}{n_i}) \tag{5}$$

3.6. Introducing Noise

To evaluate the ability of the gLSTM to deal with noisy training data, we add noise to the training set in the following way. The main purpose of this is to compare both semantic guides on how resistant they are to random noise. The noise is introduced as follows. Each word in the training sentences is replaced with another, random word with a probability of 0.15.

4. Experiments

4.1. Dataset

For evaluation and training we use *Flickr30k* [30]. Flickr30k is a widely adopted dataset containing images together with five reference captions. We use the publicly available splits [16] that divide the set into 28,000 images for training, 1,000 for testing and 1,000 for validating.

4.2. Evaluation Metrics

To evaluate the models we follow the current literature and document the two most popular automatic evaluation metrics: BLEU [24] and Meteor [3]. BLEU is a form of precision of word n-grams between generated and reference sentences. Unlike the original paper we follow Jin et al. [14] and drop the brevity penalty to be able to compare. BLEU has some obvious drawbacks such as the requirement of exact word matches. Elliott et al. [6] show that the lower BLEU scores correlate weakly with human evaluation. Meteor and the higher BLEU scores achieve moderate correlation. Meteor tries to solve some of the drawbacks of BLEU by including recall, synonyms, stemming and phrase table matches.

In addition to the automatic evaluation measures we also look at some other statistics of the generated systems to evaluate the creativity and quality of the

generated sentences. Interesting metrics include the sentence length, vocabulary size, word usage and the number of unique sentences.

4.3. Implementation Details

Our implementation extends the code publicly available by Karpathy[2]. Since we do not evaluate BLEU scores with a brevity penalty and Karpathy [15] does not provide a Meteor score, we define a set of default settings and create a reference model with his RNN implementation. To be able to compare with the results of Vinyals et al. [27] we also create a reference model with the provided LSTM implementation.

We use the last fully-connected layer of the 16-layer VGGNet [26] convolutional neural network pre-trained on ImageNet [25] as an image representation. The default training settings use *RMSprop* [10] with batches of 100 images, a decay rate of 0.999, epsilon smoothing factor 1e-8 and an initial learning rate of 1e-4. The network has 256 hidden units and an image and word encoding of size 256. To reduce the effect of exploding gradients and overfitting we use gradient clipping with threshold 5 and a drop-out percentage of 50 respectively. Only words that occur more than 5 times in the training set are added to the vocabulary. For testing we use a beam-length of 50. Due to time constraints we do not experiment with other settings.

Just like the original paper by Jia et al. [12], we implement the gLSTM on top of the LSTM code by Karpathy. For both LDA and CCA we first stem the words with the publicly available Natural Language ToolKit (NLTK)[1]. We use the *canoncorr* function available in `MATLAB` to compute the CCA projection matrices. We experiment with different numbers of correlation components but find that 256 provides the best results. We learn an LDA model with the *python* package *lda*[3]. Experiments with different numbers of topics show that the gLSTM benefits the most from 120 topics.

4.4. Results

Adding information to RNN Karpathy [15] states that results improve by adding the image only at the first generation step. Our reference add the image each time step, but we found the results to be comparable to the original. We did consider adding the image only once with LDA. This does not improve the results however. Table 1 compares the scores of the reference to those of a system with LDA. It also shows that the reference (*ref-RNN*, feeding the image each time step and not using LDA) approximates the results of the original paper by Karpathy after application of the brevity penalty to the BLEU scores. It is clear that the addition of LDA improves the results on all considered metrics.

	B1	B2	B3	B4	M
Karpathy* [15]	57.3	36.9	24	15.7	
ref-RNN*	55.2	36.6	23.9	15.1	14.3
ref-RNN	64.9	43.1	28.1	17.8	14.3
RNN + LDA	**65.4**	**44**	**29.1**	**19**	**14.4**

Table 1: *Comparison of results of adding LDA to RNN with reference and original paper [15] on Flickr30k. * indicates results with brevity penalty. Bn is the BLEU-n score. M is the Meteor score.*

	B1	B2	B3	B4	M
Vinyals [27]	**66.3**	42.3	27.7	18.3	
ref-LSTM	62.1	41.4	27.1	17.6	15.1
gLSTM+LDA	64.4	**43.2**	28.1	17.8	**16**
gLSTM+CCA	63.7	43.4	**29.2**	**19.3**	15.8

Table 2: *Comparison of results of adding LDA and CCA to LSTM with reference and original paper on Flickr30k. Bn is the BLEU-n score. M is the Meteor score.*

Adding Information to LSTM Table 2 shows the effect of adding a semantic guide to the LSTM. Both guides considered improve the results on the most important metrics compared to the reference (*ref-LSTM*) and the original paper. LDA scores best on Meteor while CCA is the best performer on the high BLEU metrics. Compared to RNN, LSTM produces slightly longer sentences, uses more unique words and generates more unique sentences.

Normalizing Beam Search We experiment with two normalization strategies. First we look at the effects of Gauss normalization. The goal of this strategy is to generate longer sentences with higher quality. Table 3 shows that Gauss normalization always improves the Meteor score. For most of the models except for gLSTM with CCA it also improves BLEU-3 and BLEU-4. Jia et al. [13] point out that the lower BLEU scores have the bad property to favor shorter sentences. Therefore Gauss normalization does not improve BLEU-1 and BLEU-2.

[2] `https://github.com/karpathy/neuraltalk`
[3] `https://pypi.python.org/pypi/lda`

	B1	B2	B3	B4	M
ref-RNN	**64.9**	**43.1**	28.1	17.8	14.3
RNN$^+$	62.4	42	**28.2**	**18.6**	**16.6**
RNN+LDA	**65.4**	**44**	**29.1**	19	14.4
RNN+LDA$^+$	62.7	42.6	28.8	**19.5**	**16.6**
ref-LSTM	**62.1**	**41.4**	27.1	17.6	15.1
LSTM$^+$	61.2	41.1	**27.3**	**18.2**	**16.9**
gLSTM+LDA	**64.4**	**43.2**	28.1	17.8	16
gLSTM+LDA$^+$	62.7	42.5	**28.8**	**19.4**	**17.4**
gLSTM+CCA	**63.7**	**43.4**	**29.2**	**19.3**	15.8
gLSTM+CCA$^+$	62.1	41.9	28.2	18.7	**17.2**

Table 3: *Effect of adding Gauss normalization evaluated on Flickr30k. $^+$ indicates the use of Gauss normalization. Bn is the BLEU-n score. M is the Meteor score.*

gLSTM `A dog runs through the grass`
gLSTM+Gauss `A brown and white dog is running through the grass`

Figure 3: *Example of improvement made by Gauss normalization.*

Gauss normalization also achieves its second goal. The length of the sentences grows from an average of 7.5 to 10.3 words. The number of generated sentences that are not in the training set grows from 75% to 90%. The number of truly unique sentences (not occurring in the training set and only occurring once in the generated sentences) does not show this growth. Figure 3 shows an example of how Gauss normalization leads to a better description.

As a second normalization strategy we propose idf normalization. We evaluate this normalization on the gLSTM model with LDA as guide. The goal is to generate longer sentences that also contain more information. Because this normalization favors more unseen words, we expect the BLEU score to drop significantly. A manual evaluation of the generated sentences shows that most images indeed lead to longer, more humanlike sentences that use less generic words. On the other hand a lot of the generated descriptions contain unwanted word repetitions and sentences that no longer correspond to the image.

	B1	B2	B3	B4	M
gLSTM	64.4	43.2	28.1	17.8	16
gLSTM+idf	40.7	23.2	13.4	7.6	12.8

Table 4: *Effects of adding idf-normalization evaluated on Flickr30k. Bn is the BLEU-n score. M is the Meteor score.*

Table 4 shows the automatic evaluation measures of idf normalization. The BLEU scores deteriorate dramatically, the Meteor score also decreases. Table 5 displays other interesting statistics that show that the idf normalization increases the creativity of the language model and creates longer sentences. Idf normalization generates sentences with an average length of 10.5. Figure 4 shows an example of an improved caption and a caption where the system fails.

Noise Resistance As a last experiment we evaluate gLSTM on the noisy training set (section 3.6) with both LDA and CCA as guide. Table 6 shows the results. Both LDA and CCA only show a small drop in the evaluation results. They are still able to generate gramatically correct sentences. A manual inspection of the most important words in each topic of LDA shows that they still are logically connected. A possible explanation of this ability to deal with noise is the way both of the models train. They both use data from 5 training sentences. The probability that all these 5 sentences show the same errors is small. Further experiments with more word changes in the sentences and other types of noise seem necessary to understand the full extent of their robustness.

Comparison with Literature Table 7 compares our best results with the results reported in the most recent literature. We compare our systems with Vinyals [27] since we extend the model that they propose. We also look at Jia et al. [12] since they proposed the gLSTM model we implemented. We also include the results of two attention based systems [14, 28].

The current systems still have a few recurring problems. The models often connect colors with a wrong object. Numbers of objects form a second

	Unique words	Avg Sentence Length	Unique1	Unique2
gLSTM	296	8.33	775	490
gLSTM+idf	**721**	**10.5**	**991**	**927**

Table 5: *Effects of adding idf-normalization to gLSTM with LDA on the sentence statistics.* Unique1 *is the number of sentences not occurring in the training set.* Unique2 *is the amount of sentences that are only generated once and do not occur in the training set.*

gLSTM `A man and woman are talking to each other`

gLSTM+idf `Two men dressed in formal attire share a conversation`

gLSTM `A man in a black jacket is looking at the camera`

gLSTM+idf `African american african american male wearing a blue jacket is looking at the camera`

Figure 4: *Examples of better and worse results with idf normalization.*

	B1	B2	B3	B4	M
LDA	64.4	43.2	28.1	17.8	16
LDA+noise	63.8	42.6	27.9	18.2	15.8
CCA	63.7	43.4	29.2	19.3	15.8
CCA+noise	63.4	42.7	28.6	18.8	15.5

Table 6: *Effect of noise on the automatic evaluation criteria for LDA and CCA as guide for gLSTM. Bn is the BLEU-n score. M is the Meteor score.*

	B1	B2	B3	B4	M
RNN+LDA	**65.4**	**44**	29.1	19	14.4
RNN+LDA$^+$	62.7	42.6	28.8	**19.5**	16.6
gLSTM+LDA$^+$	62.7	42.5	28.8	19.4	**17.4**
gLSTM+CCA	63.7	43.4	**29.2**	19.3	15.8
Vinyals [27]	66.3	42.3	27.7	18.3	
Jia [12]	64.6	44.6	30.5	20.6	17.9
Xu [28]	66.9	43.9	29.6	19.9	18.5
Jin [14]	**67**	**47.5**	**33**	**24.3**	**19.4**

Table 7: *Comparison of our best results with the current state of the art.* $^+$ *indicates the use of Gauss normalization. Bn is the BLEU-n score. M is the Meteor score.*

difficulty. LDA is not able to help with this problem since it contains a topic of numbers. It does help a bit with colors since it combines similar colors into topics.

During the experiments, we left a lot of training parameters untouched, since training a network took roughly a week on the provided hardware. Tuning these parameters may lead to improved results. This is probably why the results of Jia et al. [12] using CCA are better than our own implementation. However, our results are comparable with theirs.

The attention models perform best. It seems promising to extend our models with attention vectors. Adding attention to a model makes the training more complex and slower. This is why our RNN+LDA model is still good competition for the attention based models. The results are quite close, and our network is much faster to train.

5. Conclusion

This contribution tries to improve existing image description systems. Extensions are made by adding semantic information using LDA and CCA, and with a normalization factor in the beam search algorithm.

Adding LDA topic distributions as additional information to the RNN implementation described by Karpathy [15] improves the results on all metrics. It should be noted that RNN trains faster than the LSTM implementation but achieves comparable results. For now we used 120 topics, but further research on the upper limit of this number may lead to a bigger improvement.

Extending the LSTM model proposed by Vinyals [27] with both considered semantic guides leads to better results. LDA yields better Meteor scores, while CCA increases the BLEU scores. Compared to the original gLSTM paper [12] our system performs slightly worse. Most of the parameters of our network were not fine tuned during training. Further research of the effect of these parameters may lead to improvement.

Using Gauss normalization improves BLEU-3, BLEU-4 and Meteor scores. It also leads to longer sentences. The number of generated sentences that are not in the training set increases drastically. A gLSTM model with LDA as guide and Gauss normalization achieves the best results on the scores that correlate best with human evaluation. Idf normalization does not lead to better BLEU and Meteor scores, since it focuses on words that are not used frequently. It does however lead to more creative sentences. Apart from a lot of improved sentences it sometimes forces the model to diverge from the image content or add word repetitions. Investigating possible ways to tune the idf normalization function may lead to sentences that are both creative and correct descriptions of the image.

A comparison of the two semantic guides on their noise resistance shows that they are both very resistant. Both the BLEU and Meteor scores decrease only slightly This is probably caused by the training with 5 reference sentences. Other experiments with more changed words and other types of noise are necessary to understand the full extent of this robustness.

This work produces results comparable to the literature. Attention based models still outperform our models so adding attention can perhaps further improve our results.

References

[1] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python*. " O'Reilly Media, Inc.", 2009.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] M. Denkowski and A. Lavie. Meteor 1 . 3 : Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. 2007.

[4] J. Devlin, S. Gupta, R. Girshick, M. Mitchell, and C. L. Zitnick. Exploring Nearest Neighbor Approaches for Image Captioning. *arXiv preprint*, 2015.

[5] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, U. T. Austin, U. Lowell, and U. C. Berkeley. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2634, 2015.

[6] D. Elliott and F. Keller. Comparing Automatic Evaluation Measures for Image Description. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2:452–457, 2014.

[7] Facebook. Using Artificial Intelligence to Help Blind People See – Facebook Newsroom. `http://newsroom.fb.com/news/2016/04/using-artificial-intelligence-to-help-blind-people-see-facebook/`, April 2016.

[8] 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. *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 15–29, 2010.

[9] R. Girshick, J. Donahue, T. Darrell, U. C. Berkeley, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2–9, 2014.

[10] G. Hinton. Neural Networks for Machine Learning: Overview of Mini-batch Gradient Descent, lecture notes. http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf, 2014.

[11] M. Hodosh, P. Young, and J. Hockenmaier. Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

[12] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars. Guiding Long-Short Term Memory for Image Caption Generation. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2407 – 2415, 2015.

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, and U. C. B. Eecs. Caffe : Convolutional Architecture for Fast Feature Embedding. *Proceedings of the Association for Computing Machinery (ACM) Conference on Multimedia*, 2014.

[14] J. Jin, K. Fu, R. Cui, F. Sha, and C. Zhang. Aligning Where to See and What to Tell: Image Caption with Region-based Attention and Scene Factorization. *arXiv preprint arXiv:1506.06272*, 2015.

[15] A. Karpathy and L. Fei-Fei. Deep Visual-Semantic Alignments for Generating Image Descriptions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137, 2015.

[16] A. Karpathy, A. Joulin, and L. Fei-Fei. Deep Fragment Embeddings for Bidirectional Image Sentence Mapping. *Advances in Neural Information Processing Systems (NIPS)*, pages 1889–1897, 2014.

[17] R. Kiros, R. Zemel, and R. Salakhutdinov. Multimodal Neural Language Models. *Proceedings of the International Conference on Machine Learning (ICML)*, pages 595–603, 2014.

[18] P. Kuznetsova, V. Ordonez, and A. Berg. Collective generation of natural image descriptions. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 359–368, 2012.

[19] R. Lebret, P. O. Pinheiro, and R. Collobert. Phrase-based Image Captioning. *arXiv preprint arXiv:1502.03671*, 2015.

[20] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain Images with Multimodal Recurrent Neural Networks. *Proceedings of the Advances in Neural Information Processing Systems (NIPS) Deep Learning Workshop*, 2014.

[21] M. Mitchell, J. Dodge, A. Goyal, K. Yamaguchi, K. Stratos, A. Mensch, A. Berg, X. Han, T. Berg, and O. Health. Midge: Generating Image Descriptions From Computer Vision Detections. *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 747–756, 2012.

[22] M. Mitchell, P. Doll, F. Iandola, J. Gao, and C. L. Zitnick. From Captions to Visual Concepts and Back. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1473–1482, 2015.

[23] V. Ordonez, G. Kulkarni, and T. Berg. Im2text: Describing Images Using 1 Million Captioned Photographs. *Advances in Neural Information Processing Systems (NIPS)*, pages 1143–1151, 2011.

[24] K. Papineni, S. Roukos, T. Ward, and W. Zhu. BLEU: a method for automatic evaluation of machine translation. *... of the 40Th Annual Meeting on ...*, (July):311–318, 2002.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2014.

[26] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks For Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[27] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and Tell: A Neural Image Caption Generator. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2015.

[28] 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. *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2048–2057, 2015.

[29] Y. Yang, C. L. Teo, H. Daume, and Y. Aloimonos. Corpus-Guided Sentence Generation of Natural Images. *Proceedings of EMNLP*, pages 444–454, 2011.

[30] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From Image Descriptions to Visual Denotations: New Similarity Metrics for Semantic Inference over Event Descriptions. *Transactions of the Association for Computational Linguistics (TACL)*, 2(April):67–78, 2014.

A. LDA prediction

A.1. Choosing the number of topics

The number of topics is the most important parameter of an LDA model. However, it is a nontrivial task to evaluate the quality of such a model. This makes it hard to find the perfect number of topics. Since Jin et al. [14] propose 80 topics, we experiment with numbers in this range.

To get a grip on the accuracy of the trained LDA model, we manually evaluate the learned topic distributions. The ten most probable words for each topic give a global idea of how coherent the topics are. For each topic a summarizing name is chosen. Table 8 gives two examples of these chosen names.

Experiments show that when training a model with less than 50 topics, it becomes very difficult to find topic names that capture the learned concepts. When going higher than 120, each concept is captured in many topics, which might make it very hard for the language model to distinguish between them. It also leads to longer computation times.

A.2. Topic distribution predictions

As described in section 3.1 our approach uses a simple feedforward neural network to predict the topic distributions for unseen images. This section shows some prediction results. For each shown image, the five most probable topic names are displayed. Figure 5 shows predictions that are correct. Figure 6 shows predictions where the network makes small mistakes, but most of the topics are correct. Figure 7 shows images where the network is mistaken about the contents of the image. Moreover we find that these images correspond to those that are described poorly by each of the investigated systems. This might be caused by a bad image representation.

Topic	Probability
sit outside	0.095
sit/chair	0.070
musicians	0.047
instruments	0.035
sit at table/ restaurant	0.032

Topic	Probability
formal clothing	0.110
men together	0.078
couple/wedding	0.062
older man	0.058
beard/mustache	0.041

Figure 5: *Images with the five most probable topics, correct prediction*

Topic	Probability
sit/chair	0.044
sit outside	0.041
dog	0.036
room/floor	0.031
sleep/laydown	0.027

Topic	Probability
baseball	0.058
toddler	0.053
girl	0.50
playground	0.049
photograph	0.040

Figure 6: *Images with the five most probable topics, almost correct prediction*

Most probable words	Topic Name
wave man surf ocean surfer ride surfboard person wetsuit board	surfing
car truck drive vehicl back van road park driver behing	vehicle

Table 8: *Most probable stemmed words together with chosen topic name*

Topic	Probability
constructor	0.067
body of water	0.059
stairs/rail	0.056
boats	0.040
cleaning	0.026

Topic	Probability
dog + toy	0.077
shirtless/ bird/white	0.067
rock climbing	0.065
jump/trick	0.034
clothes/color	0.025

Figure 7: *Images with the five most probable topics, wrong prediction*