Generating Diverse and Informative Natural Language Fashion Feedback

Gil Sadeh
Amazon Lab126
gilsadeh@amazon.com

Lior Fritz
Amazon Lab126
liorf@amazon.com

Gabi Shalev
Amazon Lab126
shalevg@amazon.com

Eduard Oks
Amazon Lab126
oksed@amazon.com

Abstract

Recent advances in multi-modal vision and language tasks enable a new set of applications. In this paper, we consider the task of generating natural language fashion feedback on outfit images. We collect a unique dataset, which contains outfit images and corresponding positive and constructive fashion feedback. We treat each feedback type separately, and train deep generative encoder-decoder models with visual attention, similar to the standard image captioning pipeline. Following this approach, the generated sentences tend to be too general and non-informative. We propose an alternative decoding technique based on the Maximum Mutual Information objective function, which leads to more diverse and detailed responses. We evaluate our model with common language metrics, and also show human evaluation results. This technology is applied within the "Alexa, how do I look?" feature, publicly available in Echo Look devices.

1 Introduction

Tasks combining image and language understanding, such as image captioning and visual question answering, continue to inspire research that combines computer vision and natural language processing (NLP). These are challenging tasks that open a rich set of interactive applications.

The task of image captioning, i.e. generating natural language image descriptions, has been studied thoroughly in recent years, especially following the release of the MS-COCO captioning challenge and dataset [1, 2]. The most successful approach is based on training deep, end-to-end, encoder-decoder models, which yield impressive results [3, 4]. These models are comprised of a convolutional neural network (CNN) image encoder, and a recurrent neural network (RNN) decoder, and are trained to optimize the maximum likelihood (ML) objective function.

An additional extension of captioning models is the visual attention mechanism, which allows focusing on different image regions in each word prediction step, and leads to improved performance and generalization capabilities [5, 6, 7, 8]. Recently, [8] achieved state-of-the-art performance by combining bottom-up and top-down attention mechanisms, which enable the attention to be calculated at the level of objects and other salient image regions.

Image captioning algorithms are usually evaluated with several NLP metrics, such as BLEU [9] or CIDEr [10], which are based on precision and recall of matching n-grams. Recent works used the REINFORCE algorithm [11] to directly optimize these non-differentiable metrics and managed to reach state-of-the-art performance [12, 7, 13].

Despite the substantial progress in recent years, sentences produced by existing image captioning methods are still often overly rigid and lacking in variability. In [14, 15] an alternative GAN-based training method was proposed to generate more natural and diverse image descriptions.

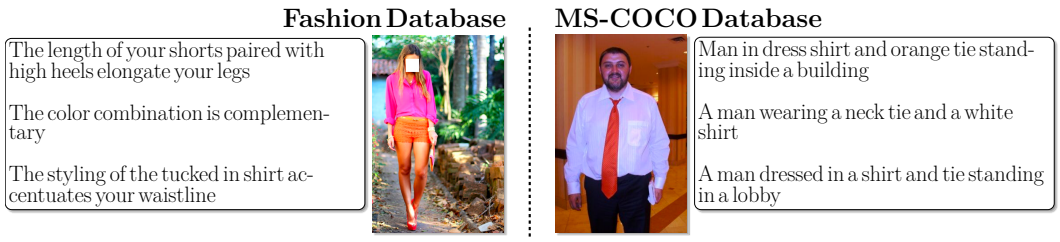

Figure 1: Comparison between our data and MS-COCO data. In standard image captioning datasets, there are some valid and similar sentences per image. In our data, there are many valid and diverse sentences per image.

In this paper, we address the task of generating fashion feedback about outfits. We collect a unique dataset for this task, containing outfit images and corresponding fashion feedback about what is good in the outfit and how to improve it. We treat each feedback type separately, and train generative models following the standard image captioning pipeline as described in Section 3.

Fashion feedback often relates to complex, subtle and abstract concepts (like fit, trend and color combination). Moreover, fashion recommendation feedback is often not directly visually grounded in the image. Thus, it is not trivial to expect the standard image captioning pipeline to generalize well to this task.

Unlike the standard image captioning task, in our case some very general responses coincide with many images (e.g., "Your outfit fits you well"). Therefore, the standard pipeline leads to non-informative responses with low variability. A similar issue is addressed in [16], where a topic modeling based solution is proposed in order to increase the diversity of generated fashion comments. We propose to utilize a decoding method based on the Maximum Mutual Information (MMI) objective, as suggested in [17], which incorporates an auxiliary language model that minimizes the effect of the sentence prior. This method yields considerable improvements in the diversity and specificity of the generated sentences.

This technology has a significant role in the "Alexa, how do I look?" feature, currently available in Echo Look devices. This feature allows customers to stand in front of this camera-enabled device, and ask for fashion feedback on their outfit. The device takes a picture, processes it and returns a vocal response. We have found that improving the diversity and specificity of the responses leads to a better customer experience.

2 Dataset

We have targeted two general types of fashion feedback, which describe what is good in the outfit, and tips for improvement. We refer to these feedback types as `GOOD` and `TIP`, respectively. Figure 4 shows examples of generated sentences. With the help of a Fashion Specialists team, we have collected approximately 170K sentences, for each feedback type, on approximately 60K outfit images. A held-out set of 500 images, collected densely with approximately 15 captions per image, is used as an evaluation set. The annotation team was instructed to write sentences that are detailed and concise.

2.1 Data Challenges

One of the main challenges in our task stems from its subjectivity. The collected sentences may often reflect the annotators personal styling taste and preferences. Moreover, the ambiguous nature of the task may lead to completely different outfit feedback, where one sentence may mention a certain fashion aspect or garment, and the others will refer to totally different aspects. This causes severe evaluation difficulties. Unlike traditional machine translation and image captioning tasks, our set of possible correct responses is much larger with a reduced tendency of using similar n-grams. Thus, most of the standard evaluation metrics, which rely on precision and recall of matching n-grams, do not reflect the performance of our models. And so, we rely heavily on qualitative tests and human evaluations.

Another challenge we face is the fact that the visual differences between images are very subtle. In the MS-COCO dataset [1, 2], image descriptions refer mostly to dominant objects and the relationships

between them. In our case, all images include a human with more or less the same set of garments, so that the visual differences are much more fine-grained. Moreover, our text often refers to abstract fashion concepts. This requires both object recognition capabilities and fashion understanding. Figure 1 demonstrates the differences between our dataset and the MS-COCO captioning dataset.

Finally, a challenging aspect in our task is avoiding model degradation to generating very general responses, such as "Your outfit fits you well". These responses could be valid for a large set of images, yet we aim at generating diverse and specific responses in order to provide informative feedback. A solution to this problem is proposed in Section 4.1.

3 Training Method

We train two separate models for each of the feedback types, `GOOD` and `TIP`. We adopt the standard image captioning training procedure for this task, which is typically done with an encoder-decoder mechanism. The encoder, which commonly consists of a convolutional neural network (CNN), transforms the image into visual features, and, in turn, a recurrent neural network (RNN) decodes these features in a generative process into a sequence of words (i.e. sentence).

The standard training procedure is termed *Teacher Forcing* [18]. The RNN receives the sub-sequence of previous ground-truth words as an input, and is trained to predict the next ground-truth word. Consider a database of images and corresponding sentences $\{I_n, \mathbf{s}_n\}$ where I_n is an image, and $\mathbf{s}_n = (w_{n,0}, \ldots, w_{n,T_n})$ is a sequence of T_n words (i.e. sentence). The model parameters, θ, are trained to optimize the maximum likelihood (ML) objective function, $\sum_n \log p(\mathbf{s}_n \mid I_n; \theta)$, where

$$\log p(\mathbf{s} \mid I; \theta) = \sum_{t=1}^{T} \log p\left(w_t \mid w_0, \ldots, w_{t-1}, I; \theta\right) . \tag{1}$$

The first and final words, w_0 and w_T, are special *begin of sentence* and *end of sentence* tokens.

We use a ResNet-18 [19] CNN image encoder, pretrained on the ImageNet [20]. Using a human detector (based on an SSD [21], and trained on a separate dataset) we extract a bounding box of a human, resize it to a fixed size of 224×448 and insert it as the image to our CNN. We extract the output of the last layer of spatial features, each of size 512, from the ResNet model. The CNN weights are fixed during the initial training steps, and after some epochs we propagate the gradients through them. Since the `TIP` feedback is often less visually grounded, we initialize the `TIP` model ResNet weights with the trained weights from the `GOOD` model.

We use a RNN decoder module which is based on the top-down attention architecture as described in [8]. The module consists of two long short-term memory (LSTM) [22] layers. The first layer, termed the top-down attention LSTM, handles the attention mechanism. The second layer, termed the language LSTM, produces the prediction of the next word. Figure 2 depicts the general structure of the decoder architecture.

The feed-forward procedure works as follows. Given an image I of a detected human, the encoder encodes the image into 7×14 features of size 512. These features are fed through a fully connected layer with ReLU activation. The parameters of this layer are trained jointly with the RNN decoder, unlike the parameters of the CNN, which are only trained after some epochs. Let $\boldsymbol{v}$ be the output of these layers, which is of size 7×14 in the spatial axes, and 512 in the features axis. At each step, t, the input to the first LSTM layer is given by $\boldsymbol{x}_t^1 = [\boldsymbol{h}_{t-1}^2, \bar{\boldsymbol{v}}, W_e \boldsymbol{e}_t]$, where $\boldsymbol{h}_{t-1}^2$ is the output of the second LSTM layer at the previous step, $\bar{\boldsymbol{v}}$ is the averaged image features (over the spatial axes), $\boldsymbol{e}_t$ is the one-hot encoding of the word w_t and W_e is an embedding matrix. The output of the first LSTM layer, $\boldsymbol{h}_t^1$, along with the spatial image encoding, $\boldsymbol{v}$, is used to calculate the attention score, for each i in the spatial axes,

$$\alpha_{i,t} = \boldsymbol{u}_\alpha^T \tanh\left(W_{v\alpha} \boldsymbol{v}_i + W_{h\alpha} \boldsymbol{h}_t^1\right), \tag{2}$$

where $\boldsymbol{u}_\alpha$, $W_{v\alpha}$ and $W_{h\alpha}$ are learned parameters. The attention weights $\hat{\boldsymbol{\alpha}}_t$ are normalized with a softmax function,

$$\hat{\boldsymbol{\alpha}}_t = \mathrm{softmax}\left(\boldsymbol{\alpha}_t\right) . \tag{3}$$

Finally, the attended image encoding is calculated with a convex combination of all input features,

$$\hat{\boldsymbol{v}}_t = \sum_i \hat{\alpha}_{i,t} \boldsymbol{v}_i \,. \tag{4}$$

The input to the language LSTM layer consists of the attended image feature, concatenated with the output of the attention LSTM, $\boldsymbol{x}_t^2 = [\hat{\boldsymbol{v}}_t, \boldsymbol{h}_t^1]$. The output of the second LSTM layer, $\boldsymbol{h}_t^2$, is used to calculate the conditional distribution over possible output words,

$$p\left(w_t | w_0, \ldots, w_{t-1}, I; \theta\right) = \operatorname{softmax}\left(W_p \boldsymbol{h}_t^2 + \boldsymbol{b}_p\right) \,, \tag{5}$$

where W_p and $\boldsymbol{b}_p$ are learned weights and biases, respectively.

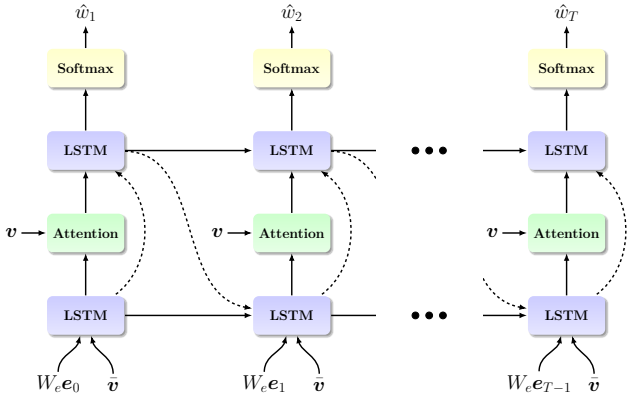

Figure 2: The top-down architecture [8] we use as our decoder. The first LSTM layer receives the averaged image features (over the spatial axes), $\bar{\boldsymbol{v}}$, and the embedded ground-truth words. It also receives the hidden state of the second LSTM layer as an input. W_e embeds the one-hot word vectors, $\{\boldsymbol{e}_t\}$, into vectors of size 512. $\boldsymbol{v}$ is the spatial image features. The second LSTM layer receives both the attention mechanism output and the first layer output as inputs. The output of the second LSTM layer is followed by a softmax layer, and $\{\hat{w}_t\}$ are the predicted words.

4 Sequence Generation

At inference, ground truth labels are not given. The model predicts a sentence, word-by-word, given the image and its previous predictions. The standard inference objective is to find the sentence that maximizes the log likelihood of the sentence given the image,

$$\hat{\mathbf{s}} = \operatorname*{argmax}_{\mathbf{s}} \log p(\mathbf{s}|I) \,. \tag{6}$$

Most image captioning works employ the beam search decoding technique in order to approximate this objective. In beam search [23], at each step, a set of k (beam width) most probable candidate sentences are considered based on the log likelihood of the sub-sequences and the next word candidates.

Empirically, we have found that increasing the beam width tends to degrade the generated sentences in terms of diversity and specificity. Since general and vague sentences can be paired with many images, and may also occur more often in our training set, the generative model would tend to favor these sentences. This happens due to the definition of the decoding objective function, which is aimed at finding the sentence that is most probable given the image.

4.1 Decoding with Maximum Mutual Information

In order to account for the issues posed by the standard decoding objective, we suggest to change it into the Maximum Mutual Information (MMI) objective function, as done in [17], where the task of conversation response generation is considered. It was found that the MMI decoding objective leads to more specific and diverse responses. The MMI decoding objective is defined by

$$\hat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmax}} \left\{ \log \frac{p(\mathbf{s}, I)}{p(\mathbf{s})p(I)} \right\} = \underset{\mathbf{s}}{\operatorname{argmax}} \left\{ \log p(\mathbf{s} \mid I) - \log p(\mathbf{s}) \right\} = \underset{\mathbf{s}}{\operatorname{argmax}} \log p(I \mid \mathbf{s}) \,. \quad (7)$$

This objective increases the specificity and diversity of the generated sentences, at the cost of allowing grammatically incorrect sentences. Hence, a compromised solution is considered. As in [17], we use a weighted average of the two objectives

$$\hat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmax}} \left\{ (1-\beta) \log p(\mathbf{s} \mid I) + \beta \log p(I \mid \mathbf{s}) \right\} = \underset{\mathbf{s}}{\operatorname{argmax}} \left\{ \log p(\mathbf{s} \mid I) - \beta \log p(\mathbf{s}) \right\} \,, \quad (8)$$

where $\beta \in [0, 1]$ is a hyperparameter, $\log p(\mathbf{s} \mid I)$ is given by our trained model and $\log p(\mathbf{s})$ is given by an auxiliary language model, as explained in Section 4.2.

4.2 Auxiliary Language Model

In order to apply the MMI criterion, an auxiliary language model (LM) is required. We consider a simple RNN LM which is comprised of one LSTM layer. The model is also trained with *Teacher Forcing*. The model parameters, ϕ, are trained to optimize the ML objective function, $\sum_n \log p(\mathbf{s}_n; \phi)$, where

$$\log p(\mathbf{s}; \phi) = \sum_{t=1}^{T} \log p \left(w_t \mid w_0, \ldots, w_{t-1}; \phi \right) \,. \quad (9)$$

This model can approximate a sentence likelihood, $p(\mathbf{s})$.

4.3 Language Mistakes

Captioning models are known to produce language mistakes in some cases [24]. As discussed in 4.1, the introduction of the MMI objective further increases the rate of language mistakes generated by our model. We have found that these mistakes mostly occur within *noun phrases* in the sentences, while the other grammatical aspects of the sentences remain valid. A noun phrase is a noun with its preceding words which help define it. We have defined the following set of rules, which detect most of these common language mistakes. For `GOOD` and `TIP`, a word cannot repeat in a noun phrase ("Add a black striped black jacket"). For `GOOD`, a noun cannot repeat ("Your leggings complement your black leggings"). For `TIP`, we consider these cases as valid ("Swap your black leggings for white leggings"), however complete noun phrases repetition is forbidden ("Swap your black leggings for black leggings").

We perform *noun phrase chunking* of the sentence with the Spacy toolkit [1]. Since the model generates a number of sentences during the beam search procedure, we can filter sentences which do not follow our set of rules.

5 Experimental Results

We train both captioning and language models for each of the question types, `GOOD` and `TIP`. The vocabulary is built for each question type separately, containing only words which appear more than 5 times in the training set. All other words are encoded as a special *unknown* token. The sentences are also pre-processed by removing punctuation marks, except for commas, and changing all letters to lowercase. The resulting vocabulary sizes are 1,745 and 1,807 words for the `GOOD` and `TIP` models, respectively.

We train the captioning models using the ADAM [25] optimizer. We use an effective batch size of 96, which is composed of 32 images with 3 sentences each. We also employ dropout [26] with probability 0.5, both in the final fully connected layer and in the word embedding layer. We train the network for some epochs (10-15) before unfreezing the CNN weights. The best model is picked according to its CIDEr-D score on the evaluation set.

[1] `https://spacy.io/`

5.1 MMI Decoding Analysis

We experiment with MMI decoding using various values of β in order to demonstrate its effect on the generated sentences. We evaluate the specificity and diversity of a given model and decoding technique by measuring 2 metrics, computed on the corpus of generated responses on images from our evaluation set. The *Diversity* metric measures the rate of unique generated sentences. The *Vocabulary Usage* metric captures the rate of words appearing at least once in the generated text corpus. We demonstrate the qualitative and quantitative effects of β in Figure 3.

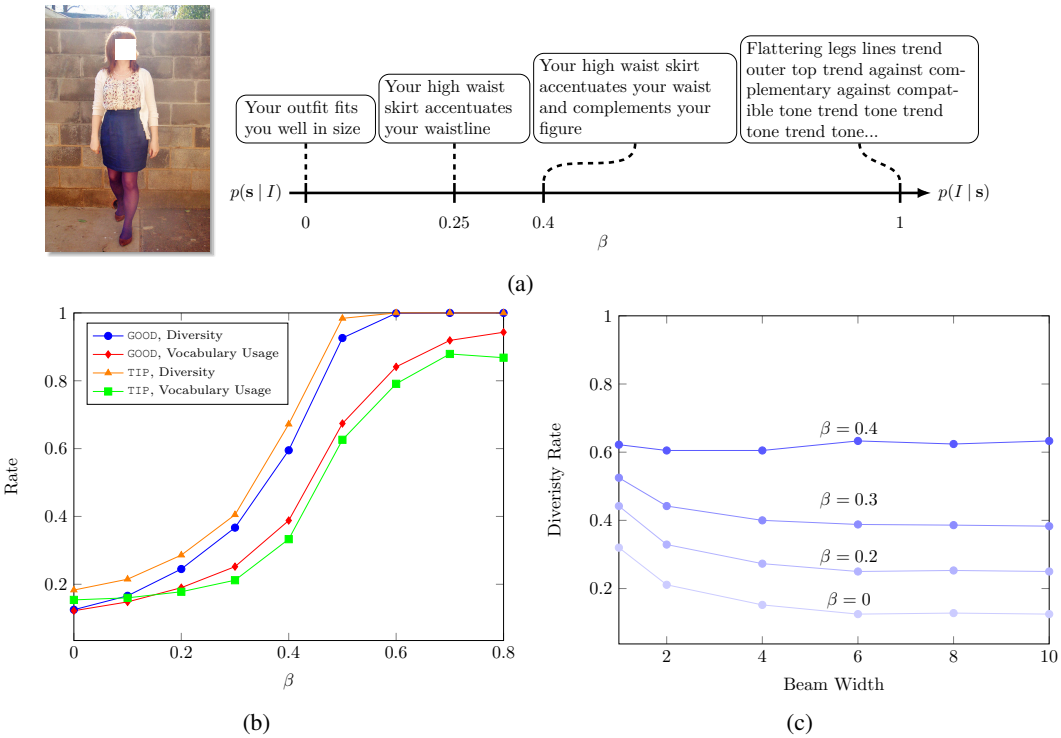

Figure 3: The effect of beam width and β on the diversity and vocabulary usage. (a) demonstrates the qualitative effect of β on generated sentences. When it is too low, the sentence is general and uninformative, while very high values may lead to grammatical mistakes. (b) shows the quantitative effect of β on diversity and vocabulary usage of generated sentences. Beam width is set to 10 in these experiments. (c) demonstrates the effect of beam width on the diversity, with different values of β, on the GOOD model. A larger beam width provides a better approximation of the decoding objective, and produces more stabilized results. Yet, diversity degradation is apparent when the beam width is increased in standard decoding ($\beta = 0$). This effect is diminished with larger β values.

Table 1: Evaluation results on standard captioning metrics, diversity (Div.) and vocabulary usage (Vocab.). In all metrics, higher is better. The FC model is the model without attention. FS stands for Fashion Specialists.

		BLEU4	ROUGE-L	METEOR	CIDEr-D	Div.	Vocab.
GOOD	FC model	0.582	0.671	0.331	0.421	0.158	0.061
	Top-Down	0.62	0.706	0.364	0.593	0.43	0.096
	Top-Down + MMI	0.604	0.694	0.365	0.743	0.848	0.127
	FS performance	0.341	0.55	0.29	0.438	0.965	0.281
TIP	FC model	0.46	0.599	0.293	0.244	0.096	0.051
	Top-Down	0.512	0.631	0.317	0.256	0.335	0.11
	Top-Down + MMI	0.518	0.623	0.317	0.38	0.737	0.141
	FS performance	0.295	0.486	0.242	0.184	0.988	0.287

5.2 Automatic Metrics Results

Traditionally, captioning tasks are evaluated with NLP metrics, based mostly on n-gram matching, such as BLEU [9], ROUGE [27], METEOR [28] or CIDEr [10]. We found that CIDEr is the only NLP metric which captures diversity and specificity. This is due to the fact that unlike other metrics, it rewards accurate generation of less frequent n-grams.

As a baseline, we compare our top-down architecture to a standard captioning model without an attention mechanism, in which a single LSTM layer receives the current word embedding at every step and predicts the following word. In this model, the image embedding is projected onto the word embedding subspace, and is inserted as input to the LSTM in the first step. We also compare our MMI decoding technique with the standard decoding objective. In addition, we estimate the performance of Fashion Specialists by selecting a random ground truth sentence for each image, and evaluating it based on the remaining ground truth sentences.

In Table 1 we report the performance of our best model in comparison to the mentioned baselines, on the evaluation set. In our best model, we incorporate the top-down architecture alongside MMI decoding with $\beta = 0.4$. We set $\beta = 0$ after 11 and 16 steps in the GOOD and TIP models, respectively. This decreases the rate of language mistakes, while not harming the diversity significantly, as also reported in [17]. It can be observed from the table, that incorporating attention improves all metrics, and using the MMI objective further improves the CIDEr, diversity and vocabulary usage metrics. It is also evident that our best model surpasses the Fashion Specialists performance in all standard NLP metrics. This shows the discrepancy of these metrics for our task. However, a gap remains in terms of diversity and vocabulary usage. Figure 4 shows examples of generated responses.

5.3 Human Evaluation

We perform an extensive human evaluation study in order to get a better qualitative evaluation of the responses our algorithm produces. In this study, we show Fashion Specialists images with sentences from the evaluation set, where the sentences are randomly selected either from the ground truth sentences or from the generated ones. We perform 2 types of human evaluations. The *Fashion* test is a fashionable variation of a Turing Test, where we ask whether the response was written by a Fashion Specialist on the given image. In addition, we wish to isolate the language aspect. The *Language* test is based on the sentence alone, without relating to the correspondence with the image. The evaluator is asked whether the response was written by a human, and is free of any language mistakes. Each annotation is performed 3 times by different Fashion Specialists. The results are presented in Table 2.

The overall performance of our models does not fall short considerably compared to the performance of the Fashion Specialists. Moreover, the results on ground truth responses in these tests can shed light on some of the problems in our dataset, which is not free of language and fashion mistakes. In the Fashion Turing Test, it can be observed that our generated responses managed to frequently fool the evaluators. Regarding the language aspect, the generated responses achieve comparable results to human performance.

Table 2: Human evaluation results. The numbers represent the rate of responses passing each test.

		Fashion	Language
GOOD	Generated	0.83	0.89
	Ground Truth	0.88	0.86
TIP	Generated	0.84	0.93
	Ground Truth	0.90	0.88

6 Conclusions

In this paper, we explored the task of fashion feedback generation in natural language. We trained end-to-end encoder-decoder models that generate sentences about what is good in the outfit, and how to improve it. We employed a decoding technique based on the MMI objective function in order to obtain diverse and specific generated responses.

These experiments show that this type of models can learn complex, subtle, abstract and non-directly visually grounded concepts. Results demonstrate that the performance of our models is comparable to Fashion Specialists in the measured metrics. The models are deployed within the "Alexa, how do I look?" feature, available in Echo Look devices.

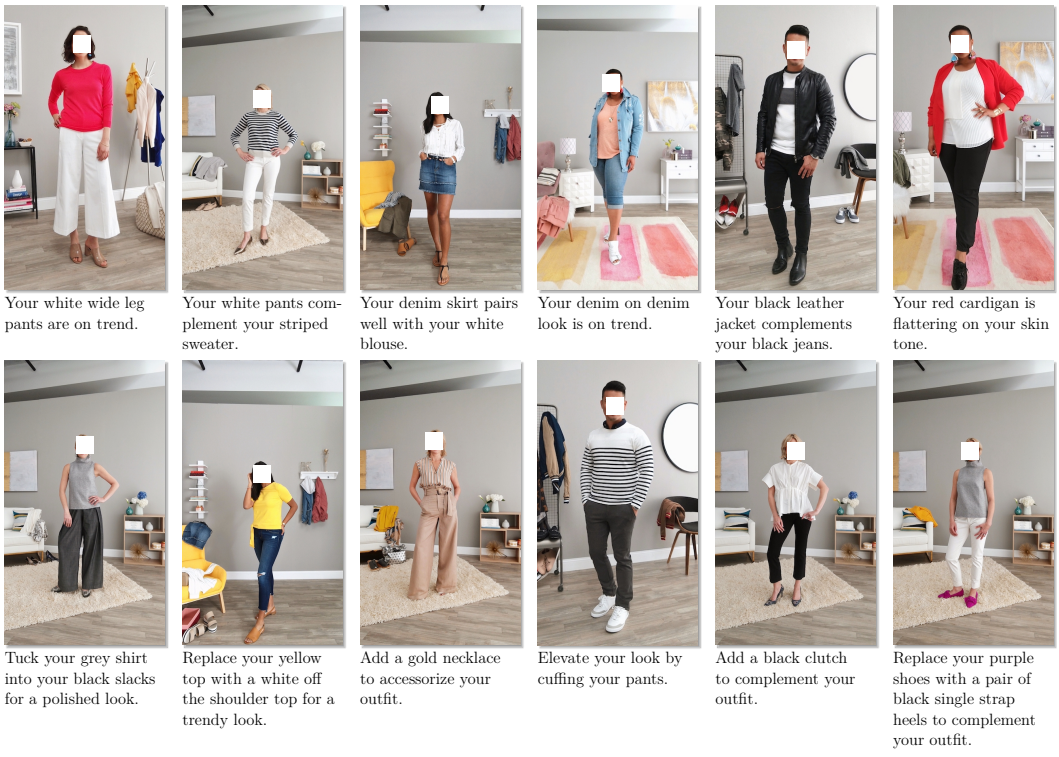

Figure 4: Examples of outfit images [2]and corresponding generated sentences from the GOOD (top row) and TIP (bottom row) models.

[2]Example images belong to Amazon marketing, and are not part of our dataset.

References

[1] 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 *European conference on computer vision*, pages 740–755. Springer, 2014.

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

[3] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[4] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[5] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

[6] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. *arXiv preprint arXiv:1612.01887*, 2016.

[7] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *arXiv preprint arXiv:1612.00563*, 2016.

[8] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998*, 2017.

[9] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[10] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[11] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[12] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

[13] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. *arXiv preprint arXiv:1612.00370*, 2016.

[14] Bo Dai, Dahua Lin, Raquel Urtasun, and Sanja Fidler. Towards diverse and natural image descriptions via a conditional gan. *arXiv preprint arXiv:1703.06029*, 2017.

[15] Rakshith Shetty, Marcus Rohrbach, Lisa Anne Hendricks, Mario Fritz, and Bernt Schiele. Speaking the same language: Matching machine to human captions by adversarial training. *arXiv preprint arXiv:1703.10476*, 2017.

[16] Wen Hua Lin, Kuan-Ting Chen, Hung Yueh Chiang, and Winston Hsu. Netizen-style commenting on fashion photos: Dataset and diversity measures. *arXiv preprint arXiv:1801.10300*, 2018.

[17] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.

[18] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

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

[23] Philipp Koehn. *Statistical machine translation*. Cambridge University Press, 2009.

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

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[27] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004.

[28] Alon Lavie and Abhaya Agarwal. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*, pages 65–72, 2005.