

Generating Unambiguous and Diverse Referring Expressions

Nikolaos Panagiaris, Emma Hart, Dimitra Gkatzia

[n.panagiaris,e.hart,d.gkatzia]@napier.ac.uk

Abstract

Neural Referring Expression Generation (REG) models have shown promising results in generating expressions which uniquely describe visual objects. However, current REG models still lack the ability to produce diverse and unambiguous referring expressions (REs). To address the lack of diversity, we propose generating a set of diverse REs, rather than one-shot REs. To reduce the ambiguity of referring expressions, we directly optimise non-differentiable test metrics using reinforcement learning (RL), and we show that our approaches achieve better results under multiple different settings. Specifically, we initially present a novel RL approach to REG training, which instead of drawing one sample per input, it averages over multiple samples to normalize the reward during RL training. Secondly, we present an innovative REG model that utilizes an object attention mechanism that explicitly incorporates information about the target object and is optimised using our proposed RL approach. Thirdly, we propose a novel transformer model optimised with RL that exploits different levels of visual information. Our human evaluation demonstrates the effectiveness of this model, where we improve the state-of-the-art results in RefCOCO testA and testB in terms of task success from 76.95% to 81.66% and from 78.10% to 83.33% respectively. While in RefCOCO+ testA we show improvements from 58.85% to 83.33%. Finally, we present a thorough comparison of diverse decoding strategies (sampling and maximisation-based) and how they control the trade-off between the quality and diversity.

Keywords: Referring Expression Generation, Natural Language Generation, Neural Models

1. Introduction

Referring Expression Generation (REG) aims at generating natural language descriptions for objects within scenes called referring expressions (REs) (Krahmer and van Deemter, 2012). The recently released datasets RefCOCO, RefCOCO+ and RefCOCOG (Yu et al., 2016; Mao et al., 2016a) which contain natural images of cluttered scenes impose new challenges to the task. Referring to objects in open domain images requires in depth understanding of the global concepts of the image, as well as their attributes and relationships. Deep learning approaches have yielded promising results on this task (Yu et al., 2016, 2017; Zarrieß and Schlangen, 2018; Castro Ferreira et al., 2019). Such approaches derive their inspiration from the recently introduced encoder-decoder paradigm (Narayan and Gardent, 2020) originally proposed for machine translation (Sutskever et al., 2014; Cho et al., 2014) and since have been widely used in various NLG sub-fields such as storytelling (Fan et al., 2018; Holtzman et al., 2018), summarization (Tan et al., 2017; Guo et al., 2018), dialogue systems (Vinyals and Le, 2015; Li et al., 2016), and image captioning (Vinyals et al., 2015; Xu et al., 2015). This architectural scheme utilizes a deep convolutional neural network (CNN) (Krizhevsky et al., 2012) to extract a vector representation of an image or image region, and a variation of recurrent neural networks (RNNs) (Jain and Medsker, 1999), e.g. a Long Short-Term Memory (LSTM) network (Hochreiter and Schmidhuber, 1997) to generate the output.

Despite the substantial progress in recent years, REG models are still far from being perfect. Our survey in Section 2 reveals that existing neural REG attempts focus mostly on the generation of unambiguous referring expressions. However, other essential natural language attributes such as *diversity* and *naturalness* have received less attention. Existing efforts focus on training objectives that promote resemblance to the ground truth sentences in order to reduce ambiguity. Secondly, due to their autoregressive nature, exact inference for generating the most likely output is intractable. Thus, it is necessary to resort to approximate search algorithms such as beam search (Koehn, 2004). However, despite the widespread adaptation of beam search, it has been found that the output decoded with beam search lacks in diversity (Vijayakumar et al., 2016a; Wang and Chan, 2019; Holtzman et al., 2020). As shown in Figure 1, beam search produces near identical expressions, with minor morphological variations (Vijayakumar et al., 2016b; Holtzman et al., 2020).

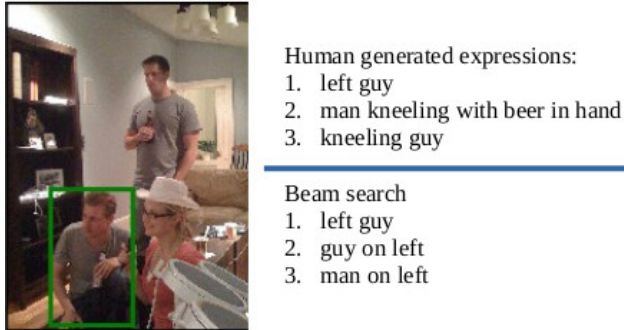


Figure 1: An example image associated with the top three referring expressions decoded with standard beam search and those provided by humans annotators. The target object is highlighted with the green box.

Diversity is important for a number reasons. First, an image contains multiple concepts at various levels of detail, and thus a RE describes a set of attributes that are interesting to the human speaker that uttered the expression. It has been shown that the content of a RE is speaker dependent (Viethen and Dale, 2010a). In other words, for the same referential environment (e.g. image), different speakers will often utter diverse expressions, a property that is reflected by the naturally existing human text. Interestingly, each of the REG datasets used in this study, namely RefCOCO and RefCOCO+, average 3 REs per object. Hence, from a machine learning standpoint, it is reasonable not only to evaluate the modes of the learned conditional distribution that reflect the accuracy, but also its variance which reflects the diversity of the generated output (Wang and Chan, 2019).

Therefore, in this work we explore an alternative approach as to what a “good” referring expression is. Our goal is to produce referring expressions that are: (1) *unambiguous*: the generated expressions should describe the object univocally; (2) *natural*: the referring expressions should be less distinguishable from the human ones; (3) *diverse*: the REG model should be able to produce a set of referring expressions for a given target object that are notably different.

We first propose to incorporate spatial attention to the standard RNN network that has been used so far in REG. Under the standard RNN framework, the generation of the next word is conditioned on the previously generated words. While this may suffice when the visual stimuli is relatively simple, for complex cluttered scenes a more fined-grained visual represen-

tation is required in order to generate high quality output. The attention mechanism bridges this gap by learning to focus on regions that are salient. In our case, the attention mechanism receives only the region of the target object, instead of receiving spatial features of the entire image. We find that the inclusion of an attention mechanism has significant benefits for REG. Our results on RefCOCO and RefCOCO+ show an increase, on average, of 0.24 in CIDEr scores (Vedantam et al., 2015) in both datasets compared to the state-of-the-art results (Yu et al., 2017).

To further demonstrate the benefits of attention in neural REG, we investigate a transformer-based architecture (Vaswani et al., 2017). Transformers have revolutionized NLG fields such as machine translation, where the machine generated translations surpass the performance of those produced by human experts (Vaswani et al., 2017). However, there are limited attempts to incorporate the transformer models in vision & language tasks. To bridge this gap, we investigate the effectiveness of the original architecture and we propose a different layer configuration in order to provide the network with a global “context” signal by connecting each layer of the encoder with the respective layer of the decoder. We show that the proposed transformer model is highly effective. We report significant improvements, both quantitative and qualitative, over baseline methods and our results compare favorably to the state-of-the-art results not only in automatic metrics but also in human evaluation.

The encoder-decoder models are trained mostly to maximize the likelihood of the generated word given the history of generated words that far. This approach has been coined in literature as “Teacher-Forcing” (Bengio et al., 2015). A limitation that stems from this approach is that the model is never exposed to its own predictions during training, while during generation the model uses its own predictions to generate the next word. Furthermore, there is a loss-evaluation metric mismatch coined as *exposure bias* (Ranzato et al., 2016). During training the model utilizes a word-level loss, while during generation its goal is to generate an expression that improves sequence-level metrics.

There is a large body of work that proposes solutions to the aforementioned exposure bias. Those approaches utilize reinforcement learning techniques (Sutton and Barto, 2018). For example, Ranzato et al. (2016) propose the use of the REINFORCE algorithm to directly optimize the non-differential evaluation metrics. A major limitation that stems from this method is that, the expected gradient exhibits high variance and without

careful normalization is often unstable (Rennie et al., 2017). An extension to the REINFORCE algorithm includes the bias correction with learned “baselines” (Schulman et al., 2016; Zaremba and Sutskever, 2015). Rennie et al. (2017) propose an alternative way to normalize the reward. Specifically, they propose the self-critical sequence training (SCST), where instead of approximating the reward signal with learned “baselines”, it uses the output of the current model at test-time to calibrate the observed reward. A limitation of this approach is that it utilizes only one sample per data point that might be insufficiently expressive for an observation. As a result, samples that poorly describe the observation will be heavily penalized, pushing the model to cover only high-probability zones. To minimize this effect, we propose a simple but effective way to calculate the baseline of the REINFORCE algorithm. Our approach, normalizes the reward by averaging over multiple-samples per observation. We hypothesize that drawing multiple diverse samples allows the construction of a robust baseline due to the diversity of the samples that are considered. In other words, averaging over multiple samples lifts the burden of having each sample to explain the observation well. We show that the proposed approach results in lower variance of the gradient than SCST.

Lastly, to overcome the lack of diversity we extend our investigation in generating sets of referring expressions. Specifically, we investigate the effect of different decoding strategies and training recipes by comparing their performance along the entire quality-diversity space. The importance that NLG systems place on these two criteria, is application dependent. For example, the goal of an open domain dialogue generation system is to be able to converse for a variety of topics and thus places more weight in the diversity of the output (Li and Jurafsky, 2016a). However, in REG the most important attribute of the output is to successfully identify the target object. Thus, generating a set of expressions is useful only if it does not come on the expense of the quality. Therefore, we present the first large-scale human evaluation to measure how the hyperparameters of each decoding algorithm, affect the diversity and the quality of sets of referring expressions.

Therefore, the contributions of this work are as follows:

- We propose an attention-based LSTM model which leads to significant improvements over the standard LSTM. Instead of letting the language model to hallucinate over the attributes that sound plausible, the attention mechanism enables the language model to be exposed to multiple salient regions of the object during generation. Thus, our

language model considers all the information pertaining to an object simultaneously.

- We propose a novel transformer-based language model for REG. Due to transformers’ limited adaptation to multi-modal tasks, we investigate their applicability to REG and we propose an architecture that injects context with different degrees of modification to the architecture, by connecting each layer of the encoder to the respective layer of the decoder.
- We present a novel optimization approach to REG based on the REINFORCE algorithm, that utilizes multiple samples per input to construct the baseline, rather than estimating the reward based on one sample. We found that the proposed RL objective reduces the variance of the gradient compared to SCST training.
- Finally, we extend our investigation to the generation of sets of referring expressions. We present the first detailed comparison of how the hyperparameters of commonly-used decoding strategies affect the quality-diversity trade-off. Specifically, we conduct the first large-scale human evaluation that measures the impact that diversity has on the quality of sets of referring expressions. We found that the recently proposed nucleus sampling (Holtzman et al., 2018), at equal points of diversity produces sets with higher quality compared to all other decoding algorithms evaluated in this work.

The rest of this work is organised as follows. Section 2 reviews existing approaches to Neural REG and inference in sequence to sequence models. Section 3 describes the proposed language models used in this work. Section 4 introduces the reinforcement learning strategy we propose to optimize REG models. Section 5 describes the decoding strategies that are compared in this work. Section 6 presents the implementation choices for each of the models used in this work, the datasets used, and the evaluation protocol that was followed. Section 7 demonstrates the effectiveness of the proposed approaches in generating one-shot referring expressions. Finally, in Section 8, we present a comparison of how existing decoding algorithms navigate the quality-diversity space when generating a *set* of REs, followed by the conclusions.

2. Related Work

Traditional view of REG: Traditionally, REG systems have been seen as a multi-step process that includes a number of choices in order to transform the input to a natural language description. The first choice is which form a referring expression will assume, i.e. whether the target object will be referred to with a proper name, a definite description or a pronoun. If the chosen form is a description, the second step is the determination of the content, that is the selection of properties that distinguish the target object from potential distractors (i.e. objects similar to the target) in a given context. The last step is the linguistic realisation of all the properties to a fully-fledged description. The large body of existing work in REG, focuses on the determination of content for definite descriptions (Krahmer and van Deemter, 2012). Content selection algorithms search for a combination of properties that distinguishes the target object univocally. The termination criterion of the search depends on the modeler’s interpretation of what constitutes a “good” referring expression. A large body of literature defines that a “good” referring expression is that which does not violate the Maxim of Quantity (Grice, 1975). In other words, a referring expression should convey just *enough* information to unambiguously identify the referent but no more. What constitutes “enough information” has led to a number of algorithmic definitions. First, the full brevity algorithm (Dale, 1989) exhaustively searches the space of possible properties of the referent in order to produce the smallest set that unambiguously identifies the referent. Due to its high algorithmic complexity, a greedy heuristic approach was proposed by Dale (1989), which incrementally chooses the properties that rule out the most distractors in the domain. One of the most influential algorithms, the incremental algorithm (Dale and Reiter, 1995) that serves as basis for a wide range of approaches, chooses the properties incrementally based on a domain-dependent preference order.

Early work in content selection did not take into consideration that speaker-dependent variation is one of the most important factors that governs the content selection process (Viethen and Dale, 2010a). However, there is a number of works that investigate the speaker-dependent variation in content selection. For instance, Bohnet (2008) extends the incremental algorithm by considering the recency of each speaker when the attributes are selected. Di Fabrizio et al. (2008) generate all possible descriptions for a given target, and then the most recent or frequent descriptions of each speaker are selected.

Other speaker-dependent REG models for content selection are presented by Viethen and Dale (2010b); Ferreira and Paraboni (2014). More recently, models that capture the speaker-dependent variation in the referential form, are introduced by Castro Ferreira et al. (2016a,b).

Neural REG: Neural REG approaches that follow the encoder-decoder paradigm (Mao et al., 2016a; Yu et al., 2016, 2017; Luo and Shakhnarovich, 2017; Zarri  and Schlangen, 2018), have seen a surge of interest due to the availability of larger and more complex REG datasets such as RefCOCO (+) (Yu et al., 2016) and RefCOCOg (Mao et al., 2016a). The underlying idea of the encoder-decoder is the following: a convolutional neural network (CNN) processes the image region in order to extract a vector representation that is used to initialize the decoder (e.g. a recurrent neural network). Given the previous generated words, the next word in the sentence is predicted sequentially. Neural REG approaches rely on incorporating contextual information by using visual features, appearance attributes (Yu et al., 2016), location features (Yu et al., 2016) and global image features as target object representation. Mao et al. (2016a) were the first to apply the encoder-decoder architecture. In particular, they use a convolutional neural network to extract visual features and an LSTM to generate the expression. The language model is trained to maximize the mutual information between the object and the associated expression through the Maximum Mutual Information objective. Yu et al. (2017) use a pre-trained comprehension module that serves as a “critic” to the language model in order to reduce the ambiguity of the produced referring expressions. Specifically, in order to guide the generation process towards unambiguous referring expressions, the language model is updated through reinforcement learning where the comprehension module plays the role of the reward function. To further reduce ambiguity, Yu et al. (2017) trained the language model jointly with the comprehension module. Similarly, Luo and Shakhnarovich (2017) utilize a comprehension module that steers the language model towards the generation of more informative expressions. Subsequently the listener module is used to re-rank the output.

Inference for conditional language models: Despite recent efforts in modeling context and learning, decoding has received little attention, with the notable exception of, for example, Zarri  and Schlangen (2018). During inference all proposed methods in REG utilize a standard decoding algorithm, e.g. greedy search or beam search. Specifically, words that maximize the likelihood are drawn sequentially. However, what is the best decoding strategy for NLG models still remains an open challenge. Although the maxi-

mization of the likelihood as training objective produces high quality models, the maximization-based decoding algorithms produce text that is repetitive (Ippolito et al., 2019a; Holtzman et al., 2020; Vijayakumar et al., 2016b). A number of diversity promoting variants of beam search have been proposed for different NLG tasks. Specifically, the noisy parallel approximate decoding was proposed by Cho (2016) for machine translation. Random noise is added to the hidden state of the decoder at each generation step. Diverse beam search (Vijayakumar et al., 2016b) was proposed for image captioning as it promotes diversity by penalizing new hypotheses that share same tokens with previously generated hypotheses. For machine translation and open dialog generation, top- g capping beam search was proposed by Li and Jurafsky (2016b), where only the top- g hypotheses from the same ancestor hypothesis are kept. The iterative beam search, that was originally proposed for dialog generation, runs multiple iterations of beam search while excludes any previously explored space.

A strand of research investigates the augmentation of beam’s search objective by training an additional network that provides a supplementary score to the likelihood. Specifically, Li et al. (2017) train an additional neural network to predict a reward for each partial hypothesis. Similarly, Zhang et al. (2018) train a network to predict dialog participants personality traits based on a partial conversation and re-ranks the candidate responses. Trainable decoding was attempted by Zarri  and Schlangen (2018) where they adopt the trainable decoder proposed by Chen et al. (2018). Specifically, an “actor” network is trained to manipulate the hidden state of the language model before it is passed to the decoding layer.

Another approach to the decoding step, is to sample from the model’s learned distribution. Under this scheme, at each time step, sampling-based decoding algorithms sample the next word by drawing a word from the conditional language model. While text generated by this method shows significant diversity, it can easily become incoherent because words from the model’s less robust confidence areas can be drawn (Holtzman et al., 2018). To the best of our knowledge, three different ways have been proposed to address this issue: (1) the use of temperature to reduce the entropy of the distribution leading to a more skewed distribution towards the high confidence zones; (2) top- k sampling (Fan et al., 2018), where a fixed number of k tokens is kept and the next word is sampled from this truncated vocabulary; (3) nucleus sampling (Holtzman et al., 2020), that keeps those tokens whose cumulative probability exceeds a pre-defined threshold.

3. Language models

In this section, we describe the language models used in this work. In Section 3.1, we describe our LSTM baseline. In Section 3.2, we describe our attention LSTM and in Section 3.3, we present our transformer model.

3.1. LSTM

The first model is a standard sequence encoder (Rennie et al., 2017; Vinyals et al., 2015). We first extract the representation of the target object with the use of a CNN, and then this representation is embedded through a linear projection W_I . Each word x_t is represented as one-hot vector, mapped to the same space as the object representation through a linear embedding. The start of each sequence is denoted by a special **BOS** token, while the special stop token **EOS** denotes the end of the sequence. For the generation of the sequence of words, we use an LSTM model. The image features are only used as an input to $t = 0$ in order to initialize the LSTM with visual features. Then, at each time step t , its output depends on the previously generated words and the hidden units, which encode the knowledge of the observed input up to this time step. More formally, the model is defined by the following update rules:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + b_i) \quad (\text{Input gate}) \quad (1)$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1} + b_f) \quad (\text{Forget gate}) \quad (2)$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1} + b_o) \quad (\text{output gate}) \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \sigma(W_{cx}x_t + W_{cm}m_{t-1} + b_c) \quad (\text{memory cell}) \quad (4)$$

$$m_t = o_t \odot \tanh(c_t) \quad (\text{hidden state}) \quad (5)$$

$$p_{t+1} = \text{softmax}(m_t) \quad (6)$$

where σ is the sigmoid function and p_{t+1} is the probability distribution over all words. The W , b matrices are learnable parameters and biases.

3.2. LSTM+ATT

Instead of utilizing a static visual representation, as the model described previously, attention-based models dynamically re-weight the spatial visual features to “attent” on specific visual regions at each time step. In this paper, we consider a modification of the architecture proposed for image captioning by Anderson et al. (2018). In particular, the attention model consists of two

LSTM layers. The first layer implements the attention mechanism, while the second plays the role of a language model and follows the update rules described in Section 3.1. The input to the LSTM attention layer is the following:

$$v_i = [r, \bar{o}, \bar{I}, h_{t-1}^L] \quad (7)$$

where \bar{o} is the concatenation of the mean-pooled object region features (i.e. $\bar{o} = \frac{1}{k} \sum_i o_i$); r, I are the CNN extracted features for the target object and image respectively and h_{t-1}^L is the previous hidden state of the language LSTM. We assume that this input representation is expressive enough for the context of the image and the state of language model in order to steer the model to information that is important for the target object.

We compute the attention weighted annotation vector $a_{i,t}$ for the uniform grid of the object region as follows:

$$a_{i,t} = \mathbf{w}_a^T \tanh(W_{oa}v_i + W_{ha}h_t^1) \quad (8)$$

$$\boldsymbol{\alpha}_t = \text{softmax}(\mathbf{a}_t) \quad (9)$$

where $W_{oa}v_i \in \mathbb{R}^{A \times D}$, $\mathbf{w}_a \in \mathbb{R}^A$ and $W_{ha} \in \mathbb{R}^{A \times d}$ are learnable parameters and A indicates the dimensions of the attention layer. Finally, the attention derived *object* visual features that will be used as input to the language LSTM is given by:

$$\hat{o}_t = \sum_{i=1}^K \alpha_{i,t} o_i \quad (10)$$

Specifically, the input to the language LSTM is the combination of the attended object features and the hidden state of the attention LSTM h_t^a . Formally the input i_t^l is the following:

$$i_t^l = [\hat{o}_t, h_t^a] \quad (11)$$

3.3. Transformer

Despite the success that attention models have achieved, there are two limitations that stem from such architectures. First, the attention mechanism only models the relationship between visual features and words, while neglecting the word-to-words interactions. Secondly, the LSTM-based models

are shallow, thus they may fail to capture abstract concepts and complex relationships due to the lack of a corresponding visual signal. The transformer model was proposed to fill this gap by simultaneously capturing the intra and inter modal interactions in a self-attention fashion using a deep stack of attention blocks. It can be conceptually divided into an image encoder and a decoder module. The encoder learns in a self-attention fashion visual representations, while the decoder makes use of the attention-derived visual representations to generate the output. In order to handle variable-length inputs, such as image regions and text sequences, the transformer employs two attention mechanisms: (1) the scaled dot-product attention; and (2) the multi-head attention. We first introduce the former type of attention since it is the most important function of the transformer model.

The scale-dot product function receives as an input: a query $q \in \mathbb{R}^d$, a set of keys $k_t \in \mathbb{R}^d$ and values $v_t \in \mathbb{R}^d$, where $t \in \{1, 2, \dots, n\}$. It outputs the weighted sum of value vectors v_t . For practical reasons, all the keys and values are packed into matrices $K = [k_1, \dots, k_n] \in \mathbb{R}^{n \times d}$ and $V = [v_1, \dots, v_n] \in \mathbb{R}^{n \times d}$ respectively. More formally, given a set of queries $Q = [q_1, \dots, q_m] \in \mathbb{R}^{m \times d}$ the scaled dot-product attention operator is defined by:

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (12)$$

where d is a scaling factor. We follow the implementation of Vaswani et al. (2017), and we use a scaling factor of $d = 64$, that indicates the cardinality of the value, key, and queries vectors. In order to attend different representation sub-spaces, the multi-head attention is introduced. It consists of h independent scaled dot-product operators named as ‘‘heads’’. Each attention head first calculates the queries, keys, and values that are projected into h sub-spaces as follows:

$$MultiHead(Q, K, V) = \text{Concat}(h_1, \dots, h_h)W^o \quad (13)$$

$$H_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (14)$$

where $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d \times d_h}$ are the projection matrices for the h independent heads, while $W^o \in \mathbb{R}^{h * d_h \times d}$ is the output projection matrix that aggregates the information from h heads. In this work, we empirically found that the optimal number of heads is eight. Therefore, all of the transformer-based architectures in this work employ eight heads.

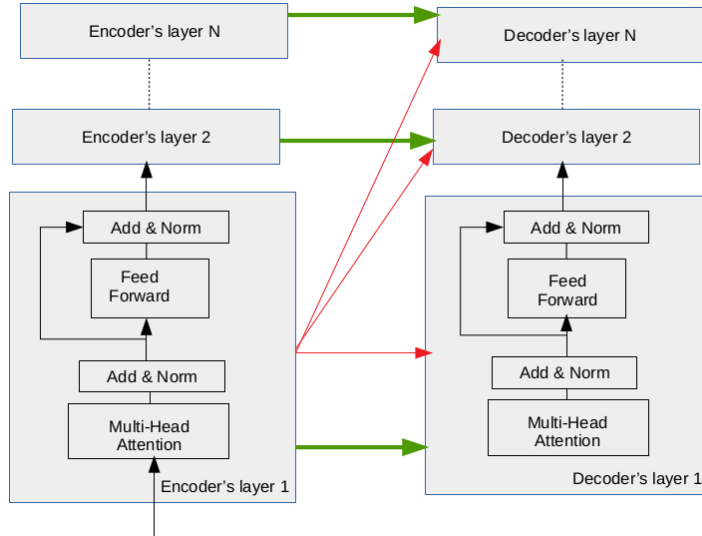


Figure 2: Overview of the transformer architecture (Vaswani et al., 2017). The red arrows illustrate the original connectivity between the encoder and decoder, while the green arrows illustrate the proposed connectivity.

The transformer leverages stacks of identical layers to mimic the encoder-decoder architecture. The overall architecture of a transformer-based model, is illustrated in Figure 2. Specifically, the encoder is a stack of N identical layers. Each layer is comprised of a multi-head attention mechanism given by the Equation 13. The second component is a position-wise feed-forward network that is applied to the output of the multi-head attention layer as follows:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (15)$$

where W_1, b_1, W_2, b_2 are the weights and biases of the two fully connected layers. Finally, residual-connections (He et al., 2016) that are followed by layer-normalization (Ba et al., 2016) are applied to the outputs of the self-attention and the feed-forward layer. The decoder's first layer receives as input the output of encoder's last layer. Similarly to the encoder, the decoder is a stack of N identical layers. However, in addition to the two sub-layers in each encoder layers, a third module is added to the decoder layers to perform multi-head attention over the encoder's output.

In this work, we propose a different connectivity pattern between the

encoder and the decoder. Specifically, we connect each layer of the encoder with the respective decoder layer. The proposed connectivity is illustrated in Figure 2 with the green arrows, while the original connectivity is shown with red arrows. Specifically, in order for a word to be predicted there should be a form of visual information that influences the likelihood. The original configuration utilizes a fixed representation throughout the network. However, fixed visual representations might be unable to capture the transitioning dynamics between the visual focus and words. Therefore, we incorporate visual features with different degrees of modification at each layer, to better model the interdependencies of different visual elements and words.

3.4. Token level objective

Let θ denote the parameters of the language models described in Sections 3.1, 3.2 and 3.3. Let $\{x_1^*, x_2^*, \dots, x_T^*\}$ be a ground-truth referring expression, the model parameters θ are trained to minimize the cross entropy loss as follows:

$$L(\theta) = - \sum_{t=1}^T \log(\pi_{\theta}(x_t^* | x_{1:t-1}^*, I, r)) \quad (16)$$

where $\pi_{\theta}(x_t | x_{1:t-1}, I, r)$ is the probability distribution of the token x_t given all the previous generated tokens $\{x_1, x_2, \dots, x_{t-1}\}$ and the visual features I, r . T denotes the length of the sequence.

4. Training REG with Reinforcement Learning

There are two limitations that stem from training a model with the cross entropy loss. The first is the exposure bias. Specifically, during training the model uses ground-truth words at each time step. However, during testing the model is fed with its own predicted words. This mismatch between training and testing, leads to error accumulation during testing, since the inferred words are different from the ground-truth. The second limitation is that during testing, the model is evaluated based on its ability to generate a high quality sequence by non-differentiable metrics, such as CIDEr. However, the model is trained to minimize a word-level objective, which leads to an inconsistency between the training objective and the evaluation metrics. Recently it has been shown that reinforcement learning techniques can bridge the gap between training and testing, by directly optimizing evaluation metrics (e.g. CIDEr) at training time.

In the classical reinforcement learning paradigm, the goal of an agent is to maximize the expectation of the reward r_t it receives for each action \hat{y}_t when interacting with its environment. More formally, an agent aims to maximize the following objective:

$$\mathbb{E}_{\hat{y}_1, \dots, \hat{y}_T \sim \pi_\theta(\hat{y}_1, \dots, \hat{y}_T)} [r(\hat{y}_1, \dots, \hat{y}_T)] \quad (17)$$

where \hat{y}_t is the word (i.e. action) sampled by the model at time t and $r(\hat{y}_1, \dots, \hat{y}_T)$ is the observed reward for the actions $\hat{y}_1, \dots, \hat{y}_T$. Each agent performs an action under a specific policy π_θ . The nature of the policy is application dependent. In the context of REG, the parameters of the agent (i.e. language model) define a policy. The agent selects an action, which is a candidate token from the vocabulary under the policy, until it generates the special token that denotes the end of the sequence. Once the agent reaches the end of the sequence, it compares the sequence of actions under the current policy \hat{y} against the ground-truth sequence y and calculates a reward based on any task specific metric (e.g. CIDEr). The goal of the training is to parameterize the agent in order to maximize the reward. Formally:

$$\mathcal{L}_\theta = - \mathbb{E}_{\hat{y}_1, \dots, \hat{y}_T \sim \pi_\theta(\hat{y}_1, \dots, \hat{y}_T)} [r(\hat{y}_1, \dots, \hat{y}_T)] \quad (18)$$

In practice, however, the expected gradient is computed with only one sample acquired from the policy π_θ as follows:

$$\nabla_\theta \mathcal{L}_\theta = - \mathbb{E}_{\hat{y}_{1..T} \sim \pi_\theta} [\nabla_\theta \log \pi_\theta(\hat{y}_{1..T}) r(\hat{y}_{1..T})] \quad (19)$$

Applying the chain rule we have:

$$\nabla_\theta \mathcal{L}_\theta = \frac{\partial \mathcal{L}_\theta}{\partial \theta} = \sum_t \frac{\partial \mathcal{L}_\theta}{\partial o_t} \frac{\partial o_t}{\partial \theta} \quad (20)$$

where o_t indicates the input to the softmax function. Thus, the estimate of the gradient \mathcal{L}_θ with respect to o_t is given by (Zaremba and Sutskever, 2015):

$$\frac{\partial \mathcal{L}_\theta}{\partial o_t} = \left(\pi_\theta(y_t | \hat{y}_{t-1}, h_t) - \mathbf{1}(\hat{y}_t) \right) (r(\hat{y}_1, \dots, \hat{y}_T) - r_b) \quad (21)$$

where r_b is a baseline reward. The role of the baseline is to guide the model towards actions with a reward $r > r_b$ and penalize those that have a reward $r < r_b$. Furthermore, subtracting a quantity from the learning signal

leads to lower variance, since it reduces its magnitude. This transformation leaves the gradient estimator unbiased because the baseline is a quantity that has zero expectation under the policy, since in this case:

$$\begin{aligned} \mathbb{E}_{\hat{y}_{1\dots T} \sim \pi_\theta} [\nabla_\theta \log \pi_\theta(\hat{y}_{1\dots T}) r_b] &= r_b \sum_{\hat{y}_{1\dots T}} \nabla_\theta \pi_\theta(\hat{y}_{1\dots T}) \\ &= r_b \nabla_\theta \sum_{\hat{y}_{1\dots T}} \pi_\theta(\hat{y}_{1\dots T}) = r_b \nabla_\theta 1 = 0 \end{aligned} \tag{22}$$

This shows that the subtraction of the baseline leaves the gradient estimator unbiased. This algorithm has been coined in literature as the REINFORCE with a baseline (Williams, 1992). The reward, for example, could be calculated as the mean of the N rewards that are observed.

Self-critical sequence training (SCST): An alternative way of reducing the variance was proposed by Rennie et al. (2017). In SCST, the reward is obtained by applying greedy search, the inference algorithm that is used at test-time. Thus, we obtain the following REINFORCE estimator:

$$\mathcal{L}_\theta = \frac{1}{N} \sum_{i=1}^N \log \pi_\theta(\hat{y}_i) \left(r(\hat{y}_{i,1}, \dots, \hat{y}_{i,T}) - r(\hat{y}_{i,1}^g, \dots, \hat{y}_{i,T}^g) \right) \tag{23}$$

where $\hat{y}_{i,t}^g$ is an action sampled with greedy decoding. In practice, however, only a single sample is used to compute the expectation. From a classic RL point of view, using a single sample is a reasonable strategy, since we might be unable to score multiple sampled actions for a state. However, from a data point of view, this is inefficient. Specifically, multiple samples can be evaluated without additional computational load. Secondly, optimizing a powerful model using one sample might have detrimental effect on its capacity. The assumption that one sample is sufficiently expressive does not always holds. As mentioned before, samples with higher rewards will be favored, while heavily penalizing samples that explain the observation poorly, leading to a lower bound of the likelihood. Therefore, the model will cover only the high-probability zones. An intuitive way to limit this crippling effect *is to average over multiple samples per data point*. The use of multiple samples per data point, provides sophisticated information leading to the construction of a more robust local baseline. Thus, we propose to use the REINFORCE with multiple-samples per input. A similar strategy, has been used on the travelling salesman problem presented by Kool et al. (2019) where they use REINFORCE without replacement and in variational inference presented by Mnih and Rezende (2016).

Algorithm 1 REINFORCE algorithm with multiple-samples per data point.

Require:A pre-trained policy (π_θ).**Input:** Input (X), ground-truth expressions (Y),**Output:** A fine-tuned policy with REINFORCE with multiple-samples.**Training Steps:****while** not converged **do** Produce a mini- batch of size N from X and Y . **for** each element in N **do** Generate K full sequences of actions:

$$\{\hat{y}_1, \dots, \hat{y}_T \sim (\hat{y}_1^{RS}, \dots, \hat{y}_T^{RS})\}_1^K.$$

 Observe the sequence rewards and calculate the baseline $b_i = \frac{1}{K-1} \sum_{i \neq j} r(\hat{y}^j)$. **end for**

Calculate the loss according to Eq. (24).

 Update the parameters of network $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}_\theta$.**end while**

REINFORCE with multiple-samples per data point: Granted that the samples within the set are independent, we can construct a baseline b for the i – th item of a set by averaging over the rest samples. We used both the arithmetic mean and the geometric mean $b_i = \frac{1}{K-1} \sum_{i \neq j} r(\hat{y}^j)$ and we empirically found a slight superiority of the latter. Therefore, the estimator in the Equation 24 becomes:

$$\mathcal{L}_\theta = \frac{1}{N} \sum_{i=1}^N \log \pi_\theta(\hat{y}_i) \left(r(\hat{y}_i) - \frac{1}{K-1} \sum_{i \neq j} r(\hat{y}_j) \right) \quad (24)$$

One of the advantages of the self-critical sequence training is that the baseline is based on the inference algorithm that is used at test-time without having to train an additional “critic” network. In particular, the greedy decoding was used (see section 5.1.1). However, greedy decoding can only produce one sample. Therefore, for generating a set of samples we resort to the use of sampling, which produces k independent samples by sampling from the model’s distribution that are diverse. As an alternative, one can use beam search that focuses on high-probability samples. However, we empirically found that the diverse sets produced by random sampling, are more informative to the gradient estimation compared to the less diverse sets produced by beam search. Algorithm 1 summarizes the required steps for the proposed approach.

5. Decoding Methods

This section presents the decoding algorithms used to decode the output. Section 5.1 describes the maximization-based decoding strategies, namely greedy decoding (§5.1.1), beam search (§5.1.2), and diverse beam search (§5.1.3). Section 5.2 presents the decoding strategies that rely on *randomness*, namely random sampling with temperature (§5.2.1), top- k sampling (§5.2.2) and nucleus sampling (§5.2.3).

5.1. Maximization-based decoding methods

5.1.1. Greedy Decoding

Greedy decoding (GD) can be seen as a naive inference method for conditional language models. It chooses the most likely token of the sequence, in a left to right manner, under the conditional probability:

$$\hat{x}_t = \arg \max_{x_t} P(x_t | x_{<t}, I, r)$$

The process continues until the end symbol is produced. Although it is computationally efficient, it can often lead to sub-optimal solutions (Cho, 2016). A significant drawback of this approach is that, the high-probability choices in earlier generation steps, can lead to an overall low likelihood sequence due to low probabilities choices later on.

5.1.2. Beam Search

Beam search (BS) is an inference algorithm that explores in a greedy left-right manner the search space. Instead of extending a single hypothesis, at each time step, it extends a set of K hypotheses H_t :

$$\mathcal{H}_t = \{(x_1^1, \dots, x_t^1), \dots, (x_1^K, \dots, x_t^K)\}. \quad (25)$$

The next set of partial hypotheses is created by expanding all the hypotheses in \mathcal{H}_t with each token from the vocabulary V . Then, each candidate hypothesis $h_{x_t^i}^i$ from H_t is scored as:

$$s(\tilde{h}_{v_j}^i) = s(h_{\tilde{y}_t^i}^i) + \log p(v_j | \tilde{x}_{\leq t}^i). \quad (26)$$

The K highest ranked hypotheses are selected as the new candidate set to be expanded in the next step. Among the top hypotheses, those whose the last token is the special EOS token are no longer expanding. The remaining

hypotheses continue to expand, however, with K reduced by the number of complete hypotheses. This process terminates until K reaches zero, and the best completed hypotheses are returned.

The space that beam search performs is the union of all the current hypotheses in \mathcal{H}^k . Thus, the K decoded sequences are from the same high-likelihood subspace. Consequently, generating a set of notable different expressions for a target object is not trivial.

5.1.3. Diverse Beam search

Diverse Beam Search (DBS) (Vijayakumar et al., 2016b) is a variant of beam search that tries to alleviate the redundancy of the search lists. DBS introduces a dissimilarity term θ that measures the difference between the current hypotheses with those produced in the previous step. It achieves that by augmenting the log-likelihood before re-ranking. More formally, each candidate hypothesis $\tilde{h}_{\leq t}^i$ is scored as:

$$s(\tilde{h}_{\leq t}^i) = s(h_{\leq t}^i) + \lambda\theta(h_{\leq t}^i, H_{t-1}).$$

where λ is a factor that regulates the strength of diversity. Another important hyperparameter is the dissimilarity function θ . We follow Vijayakumar et al. (2016a) and as dissimilarity function we use the Hamming distance that was reported to perform best.

A limitation that stems from this approach is that the fixed diversity strength is not optimal in every scenario. Vijayakumar et al. (2016b) reported that complex images benefit more from diversity-promoting inference than simpler images.

5.2. Sampling-based decoding methods

An alternative to decoding based on maximization is the introduction of some element of randomness by sampling from the model’s learned distribution. In this scenario, at each time step t the next word is randomly drawn from the conditional language model as:

$$x_i \sim P(x|x_{1:i-1}, I_i, r_i) \tag{27}$$

While output generated using this process avoids repetitions, it can become incoherent by sampling from model’s low confidence zones (Holtzman et al., 2020). REG is a low tolerance task; only one word is enough for an

unsuccessful referring expression (e.g. color or location words). To avoid sampling words from the tail of the distribution, which contains a large number of tokens assigned with low probability, three solutions have been proposed: (1) sampling with temperature; (2) top- k sampling; (3) and nucleus sampling.

5.2.1. Sampling with temperature

One common approach to control the entropy of the distribution is the use of temperature (Goodfellow et al., 2016; Ficer and Goldberg, 2017):

$$p(x = V_l | x_{1:i-1}, I, r) = \frac{\exp(u_l/T)}{\sum_{l'} \exp(u_{l'}/T)}. \quad (28)$$

The use of temperature $T \in [0, 1)$ reduces the risk of sampling words with very low probability, by skewing it towards high-probability zones (Holtzman et al., 2020).

5.2.2. Top- k Sampling

Top- k sampling that was proposed by Fan et al. (2018), is an intuitive solution that truncates the distribution by maintaining a subset of high-probability tokens. At each time step a fixed number of k words are selected that maximize $p' = \sum_{x \in V^{(k)}} P(x | x_{1:i-1}, I, r)$. Then, the next words are drawn from the top- k vocabulary $V^{(k)} \subset V$ based on their relative probabilities. Formally, the next words are drawn as follows:

$$P^*(x | x_{1:i-1}, I, r) = \begin{cases} P(x | x_{1:i-1}, I, r) / p' & \text{if } x \in V^{(k)} \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

5.2.3. Nucleus Sampling

An alternative to top- k sampling is nucleus sampling proposed by Holtzman et al. (2020). The fundamental difference between those two sampling strategies is that nucleus sampling instead of having a fixed number of tokens as subspace, it uses those tokens whose cumulative probability mass surpass a pre-define threshold q . Thus, the next words are drawn from the vocabulary $V^{(q)} \subset V$ which is the smallest set that:

$$\sum_{x \in V^{(q)}} P(x | x_{1:i-1}, I, r) \geq q. \quad (30)$$



Figure 3: Human written referring expressions for target objects (green box) in RefCOCO and RefCOCO+ images.

6. Experimental Setup

6.1. Datasets

We trained our models on RefCOCO and RefCOCO+ (Yu et al., 2016) which are built on MSCOCO dataset (Lin et al., 2014). The collection of the expressions for RefCOCO(+) datasets was based on the ReferIt Game (Kazemzadeh et al., 2014), an interactive game where two players alternate between two roles: (1) speaker: generating referring expressions; (2) listener: identifying the described object within an image. The RefCOCO(+) images contain on average 3.9 objects of the same category and they contain approximately 150k referring expressions for 50k objects. Although the images in both datasets are similar, the referring expressions for each dataset are quite different due to different data collection instructions. In particular, for RefCOCO+, the use of absolute location words (e.g. top right, bottom left, etc.) was not allowed and thus the referring expressions are *appearance* focused, while for the RefCOCO the use of *location* is essential in order for

the target object to be successfully identified. Examples of human written expressions of each dataset are shown in Figure 3. Furthermore, for each dataset different test splits are provided. The predefined test splits for both datasets are divided between person vs object splits. In particular, images containing people are in “testA” and images that contain all other object categories are in “testB”.

6.2. Implementation Details

Visual Features. The visual representation that was used is a 4096-dimensional vector that is a concatenation of: (1) a 2048-dimensional vector of the target object region; (2) a 2048-dimensional vector representation of the whole image that serves as context features. As main feature extractor we used ResNet-152 (He et al., 2016). In more detail, for the object region features, the aspect ratio of the region was kept constant and was scaled to 224×224 resolution. The margins were padded with the mean pixel value, following (Mao et al., 2016b). The attention features were extracted as follows. First, each target region was encoded with the final convolutional layer of ResNet-152. Then, bilinear interpolation was applied to resize the output to a fixed size representation of 7×7 , 10×10 and 14×14 . However, we empirically found that the 14×14 performs best. Both the object region and image features are pre-extracted and no fine-tuning was performed. The input visual representation was kept fixed across all the experiments.

Training. For our best performing LSTM and LSTM+ATT, we set the dimensions of the LSTM’s hidden state, image feature embeddings, and word embeddings to 512. The batch size is set to 128 objects. The learning rate is initialized to be 5×10^{-4} , and decays by a factor of 0.8 every three epochs.

Our best performing transformer model consists of 3 fully connected encoding and decoding layers. The dimensionality of each layer was set to 512 and 8 attention-heads were used. Every feed-forward layer is followed by a dropout with a rate of 0.1. The learning rate is initialized to be 5×10^{-4} and decays by a factor of 0.8 every three epochs, with 20000 warmup steps. The batch size was set to 10 objects.

All of our RL models are trained according to the following scheme. We first pre-train the REG models using MLE, optimized with Adam (Kingma and Ba, 2014). At each epoch, we evaluate the model on the validation set and we select the model with the best CIDEr score as an initialization for RL training. We then run RL training initialized with the MLE model to

optimize the CIDEr metric using ADAM with a fixed learning rate of 5×10^{-5} for the LSTM-based models, while for the transformer the learning rate was set to 1×10^{-5} .

6.3. Evaluation

Neural REG approaches are evaluated to produce one shot-referring expressions. However, multiple different referring expressions are often correct for a target object. Therefore, our evaluation is two-fold:

1. **Generation of a single RE:** Taking the traditional view of REG, where from one input, a single RE is generated. Under this setting, we evaluate the ability of our language models and training strategy to produce a single high quality referring expression per target object.
2. **Generation of a set of REs:** A set of REs is generated instead of a single RE for a target object. Different human speakers would probably utter referring expressions that are notably different with each other. However, this diversity is not equally reproduced by existing systems. Thus, we evaluate sets of REs in terms of quality and diversity.

Evaluation of one-shot referring expressions: We first focus on the evaluation of a system’s ability to generate a single RE. In the experiments below, we use the standard automatic metrics that have been used in REG (Mao et al., 2016a; Zarrieß and Schlangen, 2018; Yu et al., 2016) that compare the generated referring expression with the human ones. First we evaluate our models on $BLEU_1$ for uni-grams (due to the fact that models favoring shorter expressions), CIDEr and METEOR. However, previous work has shown that automatic evaluation metrics do not correlate well with human judgments (Yu et al., 2016, 2017; Zarrieß and Schlangen, 2016; Kilickaya et al., 2017). Therefore, we randomly selected 60 objects from each test set and we collected human judgments on Amazon Mechanical Turk. In all experiments, participants were presented with an image and an expression and were asked to draw a box around the referent object which they thought as the best match. In order for a RE to be considered successful, two annotators had to draw a box around the correct object. In addition to the evaluation of the success of referring expressions, annotators were asked to rate the statements below following Mitchell et al. (2012):

- **Q1-Grammaticality:** The description is grammatical correct.

- **Q2-Main aspects:** The description does not describe the main attributes correctly.
- **Q3-Correctness:** This description does not include extraneous or incorrect information.
- **Q4-Naturalness:** It sounds like a person wrote that description (Yes/No)

Evaluation of a set of referring expressions: In order to evaluate a set of referring expressions two criteria are required to be taken into consideration: accuracy and diversity. For the former, the commonly used approach is to average a similarity score (Ippolito et al., 2019b), e.g. CIDEr, over the set. Evaluating the accuracy of a particular system is not sufficient to reflect the overall performance of a model; the diversity of the output should also be considered. The diversity of a set is computed using Self-CIDEr (Wang and Chan, 2019), that computes a diversity score by calculating the eigenvalues of a kernel matrix that contains similarities scores (i.e. CIDEr) for all sentences pairs within the set (Wang and Chan, 2019).

Furthermore, we conducted human evaluation on Amazon Machine Turk and we asked humans to rate the diversity and the quality of a set of referring expressions. Specifically, we randomly selected 25 target objects and we generated 5 expressions for each object. For each expression in the set, three workers were asked whether or not the expression describes the object unambiguously. A referring expression was considered successful if two workers found that the expression unambiguously describes the object. We then required the workers to rate the diversity of the set on a 5-point Likert scale, where 1 indicates that the expressions are identical, and 5 that the expressions are significantly different with one another. In our instructions, diversity refers to different words, phrases, sentence structures, semantics or other factors that impact diversity. The diversity score for each set is the average score given by the 3 workers.

7. Generation of one-shot Referring Expressions

7.1. Attention-based REG

In order to demonstrate the advantages of the proposed object attention, we performed a detailed comparison between the attention model and the standard LSTM. For each of the considered metrics, we performed a

		RefCOCO				RefCOCO+			
		testA		testB		testA+		testB+	
Model Type	Decoding Method	$BLEU_1$	CIDEr	$BLEU_1$	CIDEr	$BLEU_1$	CIDEr	$BLEU_1$	CIDEr
LSTM	Greedy	0.490	0.762	0.523	1.332	0.444	0.633	0.373	0.710
	Beam	0.477	0.758	0.510	1.340	0.429	0.656	0.384	0.837
LSTM +ATT	Greedy	0.594	1.033	0.609	1.552	0.512	0.884	0.424	0.858
	Beam	0.577	1.013	0.599	1.573	0.491	0.881	0.424	0.857
p-value		<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

Table 1: Comparison of different automatic metrics for the attention model (denoted as “LSTM-ATT”) and the standard LSTM model. The proposed attention model results in significantly higher CIDEr and $BLEU_1$ scores in both datasets. The p-values are the result of two-tailed t-tests using paired samples.

two-tailed t-test with paired samples in order to determine whether the difference caused by incorporating the object attention was statistically significant. The results for the two considered datasets are shown in Table 1. We first note that the proposed attention model results in higher scores than the standard LSTM. The difference in scores was found statistically significant (using a significance level $\alpha = 0.05$). The significant improvements in CIDEr and $BLEU_1$ are in line with our expectation that adding the object attention mechanism would assist the model in determining both the relationship between objects, but also determine fine appearance details of the target object. This is due to the fact that our approach is able to consider all the information pertaining to an object simultaneously.

To illustrate the advantages of the proposed approach, we present examples of objects with the corresponding referring expressions generated by each model (see Figure 4 and Figure 5). The referring expressions presented here were generated using the following steps: both models were trained with MLE and were greedily decoded. We chose those examples for which there was a significant improvement between the CIDEr scores of the expressions generated by the attention model and those generated by the standard LSTM. The collection of objects and expressions for RefCOCO and RefCOCO+ is shown in Figure 4 and Figure 5 respectively. It should be noted that, during the collection of RefCOCO dataset, no restrictions were placed on the type of language that can be used in the referring expressions, while in RefCOCO+ dataset location words were not allowed. Thus, this dataset contains referring expressions that are based on appearance attributes. Specifically, the images in testA that are presented in Figure 4, illustrate an improvement in determining when a relationship between objects should be expressed, as well



Figure 4: Examples of objects and expressions drawn from RefCOCO dataset, for which the CIDEr scores of the attention model show an improvement over the standard LSTM. The target object is highlighted with a red box.



Figure 5: Examples of objects and expressions drawn from RefCOCO+ dataset, for which the CIDEr scores of the attention model show an improvement over the standard LSTM. The target object is highlighted with a red box.

Model	RefCOCO				RefCOCO+			
	testA		testB		testA		testB	
	CIDEr	$BLEU_1$	CIDEr	$BLEU_1$	CIDEr	$BLEU_1$	CIDEr	$BLEU_1$
Transformer 6	0.837	0.506	1.340	0.546	0.772	0.460	0.763	0.387
Transformer 6 (OURS)	0.852	0.513	1.355	0.552	0.798	0.467	0.791	0.395
Transformer 3	0.922	0.524	1.442	0.581	0.911	0.515	0.894	0.412
Transformer 3 (OURS)	0.938	0.586	1.464	0.586	0.938	0.529	0.913	0.424

Table 2: The impact of depth in the performance of the transformer model. Transformer 6 and 3 indicate that the decoder and the encoder consist of 6 and 3 layers respectively. The layer configuration follows the one proposed by Vaswani et al. (2017). “Ours” indicates that each layer of the encoder is connected with the respective layer of the decoder.

as in determining what that relationship should be. In addition, the images in testB presented in Figure 4, illustrate an improvement in including appearance and location attributes. The improvement in including appearance attributes can be further noticed in the referring expressions of RefCOCO+ dataset presented in Figure 5.

7.2. Transformer-based REG

Table 2 shows the results for our ablation study regarding the transformer model discussed in Section 3.4. We show the original configuration (denoted as Transformer 6 in Table 2) of the transformer (Vaswani et al., 2017) as our baseline. To determine whether the changes in the configuration of the model result in statistically significant differences for each of the considered metrics, we performed a two-tailed t-test with paired samples as described in Section 7.1.

We first investigate the effect of the number of layers. Our hypothesis is that, given the model was initially proposed for machine translation, a task with considerable longer sentences than REG and larger training sets, a shallower architecture might result in better performance. Table 2 shows that reducing the depth (Transformer 3 in Table 2) of the network leads to considerable improvements in both $BLEU_1$ and CIDEr scores. For instance, in RefCOCO+ testA, decreasing the number of layers leads to an improvement from 0.772 to 0.911 in CIDEr values. The score difference was statistically significant (using a significance level $\alpha = 0.05$).

We then investigate the effect of connecting each layer of the encoder to the respective layer of decoder. The results are shown in Table 2, where “Transformer (OURS)” stands for the proposed model. Specifically, for all of the considered metrics, the proposed transformer produces higher scores



Figure 6: Examples of objects and expressions drawn from both RefCOCO and RefCOCO+ datasets, for which the CIDEr score for the proposed transformer model show an improvement over the standard transformer. The target object is highlighted with a red box.

than the standard transformer.

Examples of generated REs are illustrated in Figure 6. The referring expressions presented here were generated using the following steps: both models were trained with MLE and were decoded using greedy decoding. In all images presented in Figure 6, we observe that the proposed model improves over the standard transformer in inferring fine appearance (e.g. “number 29” top left image in Figure 6) and location attributes of the target object. This is in line with our expectation that utilizing features with different degrees of modification at each layer, will better model the interdependencies of different visual elements and words.

7.3. Training REG with Reinforcement Learning

Next, we compare the proposed RL method (see Section 4) with self-critical sequence training. First, we explore which reward function to use to evaluate the sequences. We experimented with training directly with different evaluation metrics that have been used in neural REG literature, i.e. $BLEU_1$ and METEOR, as well as a combination of metrics. The results are shown in Table 3. As expected, optimizing towards a particular evaluation metric

RefCOCO						
	testA			testB		
Training Metric	CIDEr	$BLEU_1$	METEOR	CIDEr	$BLEU_1$	METEOR
MLE	0.762	0.490	0.177	1.332	0.523	0.208
CIDEr	0.978	0.556	0.211	1.498	0.536	0.229
$BLEU_1$	0.811	0.512	0.190	1.342	0.501	0.211
METEOR	0.762	0.489	0.178	1.331	0.522	0.209
CIDER+ $BLEU_1$	0.914	0.534	0.202	1.422	0.527	0.223
RefCOCO+						
	testA+			testB+		
Training Metric	CIDEr	$BLEU_1$	METEOR	CIDEr	$BLEU_1$	METEOR
MLE	0.633	0.444	0.167	0.710	0.373	0.159
CIDEr	0.847	0.500	0.203	0.980	0.288	0.169
$BLEU_1$	0.760	0.480	0.189	0.914	0.299	0.163
METEOR	0.729	0.442	0.179	0.932	0.321	0.169
CIDER+ $BLEU_1$	0.845	0.517	0.207	0.979	0.299	0.171

Table 3: Performance of different reward functions for the LSTM model. When the language model is optimized with the CIDEr metric, we observe an significant increase to all other evaluation metrics. All models were decoded using greedy decoding. The performance of the seed model is also reported. The best overall values for each metric are emphasized with bold.

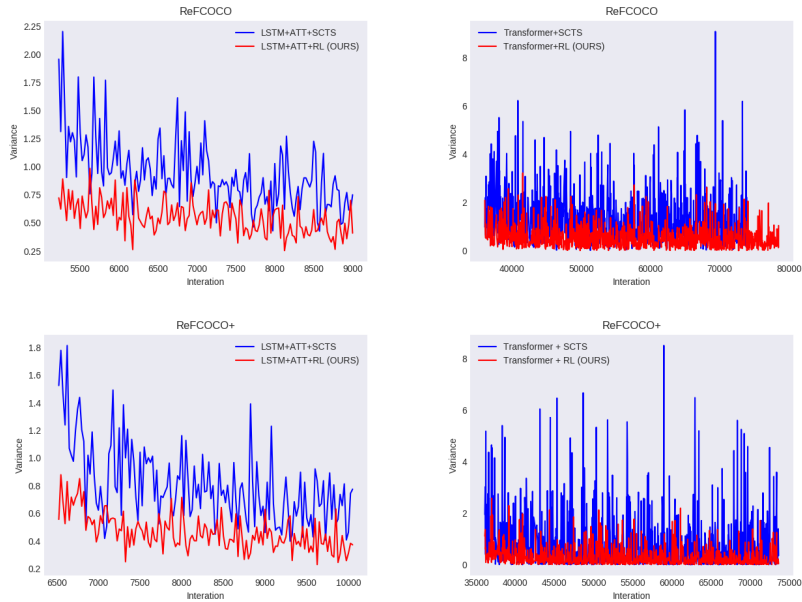


Figure 7: Gradient variance of the proposed RL objective compared to the SCST for the proposed attention and transformer model.

Model	RefCOCO				RefCOCO+			
	testA		testB		testA		testB	
	CIDEr	$BLEU_1$	CIDEr	$BLEU_1$	CIDEr	$BLEU_1$	CIDEr	$BLEU_1$
LSTM +ATT + MLE	1.033	0.594	1.552	0.609	0.884	0.512	0.858	0.424
LSTM + ATT + SCST	1.089	0.597	1.565	0.570	1.065	0.563	1.054	0.323
LSTM + ATT+ RL(OURS)	1.204	0.636	1.646	0.605	1.077	0.563	1.074	0.323
p-value	<0.001	0.01	<0.001	0.01	0.01	0.63	0.01	0.15
Transformer	0.938	0.529	1.464	0.586	0.938	0.529	0.913	0.424
Transformer + SCST	1.255	0.650	1.710	0.650	0.967	0.532	0.974	0.308
Transformer + RL(OURS)	1.261	0.665	1.732	0.656	1.020	0.546	1.003	0.294
p-value	0.01	0.01	0.01	0.01	<0.001	<0.001	<0.001	<0.001

Table 4: Performance of the best attention (denoted as LSTM +ATT) and transformer model (denoted as Transformer) trained with maximum likelihood estimation (denoted as MLE), self-critical sequence training (denoted as SCST) and the proposed RL objective (denoted as RL (OURS)). The p-values are the result of two-tailed t-tests using paired samples.

during training leads to an increase on that particular metric during testing. However, we found that CIDEr optimization increases the performance of all other metrics considerably. We further noticed that when a model is optimized with either $BLEU_1$ or METEOR, produces shorter sentences than a model that is optimized with CIDEr. Hence, we hypothesize that the brevity penalty in $BLEU_1$ (Papineni et al., 2002) and METEOR’s length penalty (Banerjee and Lavie, 2005) adversely affect the score. Therefore, for the rest of this work, all RL models are based on CIDEr optimization.

Next we evaluate whether the proposed RL objective reduces the variance of the gradient compared to self-critical sequence training. We hypothesize that using multiple samples to estimate the expectation will reduce the variance. Figure 7 compares the variance of the two methods. Although both techniques lead to unbiased estimators of the gradient, our proposed method results in lower gradient variance for both language models that were tested. Interestingly, SCST has much higher gradient variance than the proposed RL objective during the first epoch of training. We hypothesize that the samples drawn from the model’s distribution score lower than the sentences produced by greedy decoding.

Table 4 presents the results on RefCOCO and RefCOCO+ for the proposed attention model and the transformer model optimized with the proposed RL training strategy and SCST. Both RL based models are fine-tuned from the same pre-trained model. Again, for each of the considered metrics, we performed a two-tailed t-test in order to determine whether the difference in scores between the two RL methods was statistically significant. We first

Model	RefCOCO		RefCOCO+	
	testA	testB	testA	testB
	CIDEr		CIDEr	
speaker+listener+MMI+rerank (Yu et al., 2017)	0.763	1.306	0.500	0.734
speaker+reinforcer+MMI+rerank (Yu et al., 2017)	0.748	1.311	0.499	0.729
speaker+listener+reinforcer+MMI+rerank (Yu et al., 2017)	0.775	1.320	0.520	0.735
LSTM+ ATT	1.033	1.552	0.884	0.858
Transformer	0.938	1.464	0.938	0.913
LSTM+ ATT + RL	1.204	1.646	1.077	1.074
Transformer + RL	1.261	1.732	1.020	1.003

Table 5: Comparative analysis to existing state-of-the-art approaches.

note that when a model is optimized with the proposed RL objective achieves higher CIDEr scores than SCST. The score difference was statistically significant. Second, the $BLEU_1$ score difference was statistically significant in RefCOCO and RefCOCO+ testB. Third, we observe that the attention model achieves higher scores when trained with MLE in both datasets compared to transformer. However, when both models are trained with RL, the transformer presents higher scores than the attention LSTM in RefCOCO.

Finally, we compare our best performing models against the three best models presented by Yu et al. (2017). Table 5 shows the scores as reported by the authors along with our best performing models. The model “speaker+listener+MMI+rerank” is a generative model that receives as additional input a listener-aware representation. The second model “speaker+reinforcer +MMI+rerank” utilizes a pre-trained comprehension module to update the parameters of the generative module through reinforcement learning. The last model is the combination of all aforementioned modules. Additionally, all three models use a comprehension model as a post-ranking tool to rank a set of referring expressions. First we note that, the systems when trained with MLE outperform the three best models reported by Yu et al. (2017). Further improvements are noticed when the models are trained with the proposed RL method.

7.4. Human evaluation of one-shot Referring Expressions

Previous work has shown that automatic evaluation metrics do not correlate well with human judgments (Yu et al., 2016, 2017; Zarri  and Schlangen, 2016; Kilickaya et al., 2017). Unlike other generation tasks such as image captioning, here a referring expression is successful if it describes the target object unambiguously. Thus, we conduct human evaluation on 60 randomly

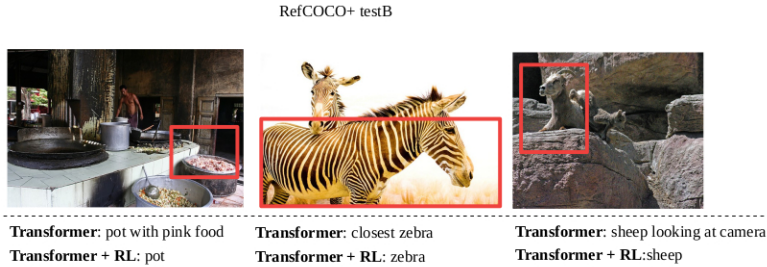


Figure 8: Examples of objects and expressions for which human annotators successfully identified the described object when the model was trained with MLE, while failed to identify the object when the model was fine-tuned with RL.

RefCOCO testA					
	Task success	Naturalness	Grammaticality	Main Aspects	Correctness
LSTM+ATT	71.66%	92.85%	4 (3.41, 0.68)	2 (2.23, 0.92)	3 (3.30, 0.77)
Trasformer	78.33%	96.42%	4 (3.69, 0.62)	3 (2.87, 0.88)	4 (3.64, 0.88)
Transformer+ RL	81.66%	85.71%	3 (3.39, 0.64)	2 (2.14, 0.87)	4 (3.83, 0.78)
Best by Yu et al. (2017)	76.95%	-	-	-	-

RefCOCO testB					
	Task success	Naturalness	Grammaticality	Main Aspects	Correctness
LSTM+ATT	66.66%	98.92%	4 (3.71, 0.83)	2 (2.23, 0.92)	3 (3.30, 0.77)
Transformer	73.33%	98.21%	3 (2.78, 0.61)	3 (2.28, 0.79)	3 (2.85, 0.71)
Transformer+ RL	83.33%	96.42%	4 (3.80, 0.54)	2 (2.12, 0.85)	3 (2.96, 0.49)
Best by Yu et al. (2017)	78.10%	-	-	-	-

RefCOCO+ testA					
	Task success	Naturalness	Grammaticality	Main Aspects	Correctness
LSTM+ATT	76.66%	93.64%	4 (4.12, 0.92)	2 (1.91, 0.93)	3 (3.25, 0.96)
Transformer	80.00%	95.44%	4 (3.82, 0.38)	3 (2.11, 0.68)	4 (3.78, 0.55)
Transformer+ RL	83.33%	92.85%	3 (3.30, 0.49)	2 (1.23, 0.87)	4 (3.92, 0.25)
Best by Yu et al. (2017)	58.85%	-	-	-	-

RefCOCO+ testB					
	Task success	Naturalness	Grammaticality	Main Aspects	Correctness
LSTM+ATT	55.00%	71.42%	4 (4.17, 0.92)	2 (1.91, 0.93)	3 (3.25, 0.96)
Transformer	58.33%	92.85%	4 (3.62, 0.51)	3 (2.89, 0.64)	3 (3.07, 0.59)
Transformer+ RL	51.66%	90.78%	4 (3.67, 0.84)	3 (2.26, 0.99)	3 (2.91, 1.31)
Best by Yu et al. (2017)	58.20%	-	-	-	-

Table 6: Human Evaluation results. Median scores for systems, mean and standard deviation in parentheses.

selected objects for each test set. We ask Amazon Mechanical Turk workers to draw a box around the object that they believe is best described by a given expression. If two workers chose the correct object, then the expression was considered successful. Furthermore, we extend the existing human evaluation protocol, by collecting ratings (from strongly disagree to strongly agree) for

naturalness, grammaticality, main aspects and correctness (see Section 6.3). We report the scores for the systems in Table 6.

We first evaluate the proposed attention model against the transformer. The results presented in Table 6 demonstrate that the proposed transformer model is more effective in task success compared to the attention model. Additionally, we observe that it scores higher in naturalness of the produced referring expressions across datasets. Then, we evaluate whether the proposed RL objective further improves the performance. The results are shown in Table 6. We note that the proposed RL objective improves the success of the produced referring expressions in RefCOCO dataset and in RefCOCO+ testA. However, it reduces the performance in RefCOCO+ testB. In order to better understand the failure modes of our model, we present example objects with the corresponding referring expressions generated by each model in Figure 8. We chose those examples for which human annotators successfully identified the target object described by an expression produced by the transformer model trained with MLE, but failed to identify the same object described by an expression produced by the transformer model optimized with RL. For all three expressions generated by the MLE trained model presented in Figure 8, the human annotators found that the main aspects of the objects were not described accurately but the objects were successfully identified. Thus, we hypothesize that optimizing a model towards CIDEr, concentrates the probability distribution to words that improve CIDEr, while suppress those that are not beneficial for the metric. However, this concentration of the probability around words that are beneficial for CIDEr might adversely affect the success of REs.

8. Generation of a set of REs

This section aims to explore how decoding algorithms, training procedures affect the accuracy-diversity trade-off. As our language model we chose the transformer model (see Section 3.3) that achieved state-of-the-art results in human evaluation for the one-shot generation. As training objective we use: (1) the cross-entropy loss (see Section 3.4); and (2) the proposed RL method (see Section 4). Table 7 shows the diversity parameter that controls the accuracy-diversity trade-off for each of the employed decoding strategies.

Secondly, most of the published models are trained to generate a single referring expression, thus we adapt the decoding strategies to generate a set of REs. In particular, we use two approaches to generate a set: (1) for the

Decoding method	Hyperparameter
Random Sampling (RS)	Temperature T
top- k Sampling	The number k of tokens to be kept.
Nucleus Sampling (NS)	The probability threshold q
Diverse beam Search (DBS)	The diversity strength parameter λ
Beam search (BS)	Temperature T

Table 7: The hyperparameter that controls the quality-diversity trade-off for each of the decoding strategies used in this work.

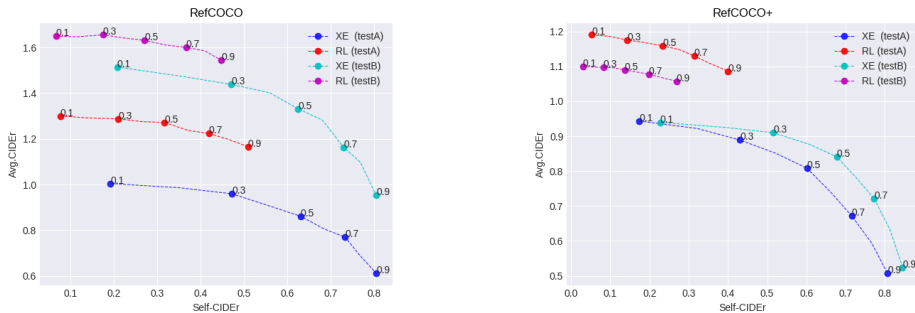


Figure 9: Self-CIDEr and average CIDEr scores for random sampling with different temperature values. The language model used is the proposed transformer trained with cross-entropy (XE) and fine-tuned with the proposed RL objective.

randomization-based algorithms (e.g. random sampling), a set of referring expressions is constructed by randomly sampling from the model’s learned distribution; and (2) for normal and diverse beam search we use the beam width to generate the set.

8.1. Random Sampling-based Decoding Methods

We first investigate how temperature affects the accuracy-diversity trade-off for random sampling. As illustrated in Figure 9, higher sampling temperatures result in both an increase in Self-CIDEr scores and a reduction in average CIDEr scores. Interestingly, using CIDEr reward to fine-tune the model will drastically reduce Self-CIDEr, while will increase the average CIDEr score. Optimising the CIDEr reward encourages syntactic similarity between the generated expressions and the ground truth expressions which leads to low diversity. To illustrate the differences between the two objectives

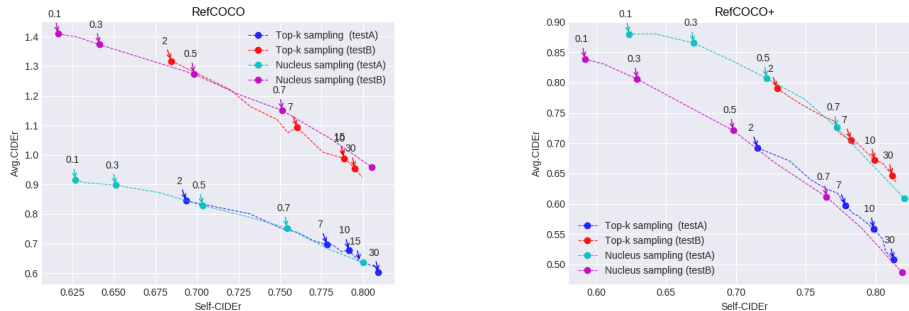


Figure 10: Self-CIDEr and average CIDEr scores for top- k and nucleus sampling for varying k and q values. The temperature was set to $T = 1$. The language model used is the proposed transformer trained with cross-entropy.

and how different temperature values affect the output, we present example objects with the corresponding expressions generated by each model in Figure 11. First, we observe that for both objectives when the sampling temperature is set to 0.1, the text is highly repetitive, mimicking greedy search. Furthermore, when the temperature is set to 0.9, the model trained with cross entropy produces output that is less fluent and incoherent (see Figure 11). However, minor changes are noticed to the output of the RL-trained model. One interpretation of this behavior is that, the RL-trained model is concentrated around words that benefit CIDEr. In other words, the learned distribution of a RL-model is already skewed towards a few high probable tokens and thus invariant to the effect of temperature. Hence, we focus mainly on the XE trained models.

Top- k and nucleus sampling have become an alternative to random sampling. Both strategies sample from a truncated distribution. The difference between the two is how they truncate the distribution; top- k sampling keeps a fixed number of k tokens that have been assigned high-probability, while nucleus sampling keeps those tokens whose cumulative probability mass exceeds a pre-defined threshold q . Figure 10 illustrates how the two strategies affect the accuracy-diversity trade-off. We observe that nucleus sampling achieves higher avg. CIDEr compared to top- k sampling. We hypothesize that the reason for which top- k results in lower avg. CIDEr scores is that, the distribution is truncated to a fixed number of tokens regardless of the input. There might be cases that there are too many or too few probably tokens. Thus, the fixed number of tokens could potentially lead to sub-optimal solu-





RefCOCO testA	RefCOCO+ testA	RefCOCO testB	RefCOCO+ testB
			
Human written expressions: 1) red shirt 2) woman wearing red of the left 3) red shirt RL + random sampling (t=0.1) : 1) red shirt 2) red shirt 3) red shirt 4) red shirt 5) red shirt XE + random sampling (t=0.1) : 1) red shirt 2) red shirt 3) red shirt 4) red shirt 5) red shirt RL + random sampling (t=0.9) : 1) woman in red 2) red shirt 3) woman in red shirt 4) red shirt 5) red shirt XE + random sampling (t=0.9) : 1) red 2) sitting woman in red hoodie on left 3) leftmost person red guy 4) red shirt left 5) red shirt	Human written expressions: 1) girl 2) girl 3) the one with a hot on in all black RL + random sampling (t=0.1) : 1) woman in black 2) woman in black 3) woman in black 4) woman in black 5) woman in black XE + random sampling (t=0.1) : 1) man in hat 2) man in hat 3) man in hat 4) man in hat 5) man in hat RL + random sampling (t=0.9) : 1) woman in hat 2) white hat 3) woman in black 4) woman in black 5) woman in hat XE + random sampling (t=0.9) : 1) man behind closest man 2) person with white hat not stove hat 3) man with cut off with coat on grapefruit hat in 4) white hat 5) girl in hat	Human written expressions: 1) cup of coffee 2) coffee 3) coffee RL + random sampling (t=0.1) : 1) coffee 2) coffee 3) coffee 4) coffee 5) coffee XE + random sampling (t=0.1) : 1) coffee 2) coffee 3) coffee 4) coffee 5) coffee RL + random sampling (t=0.9) : 1) coffee cup 2) coffee mug 3) coffee 4) coffee 5) right cup XE + random sampling (t=0.9) : 1) coffee cup with coffee on top 2) coffee cup top of plate 3) coffee cup 4) coffee 5) white coffee mug	Human written expressions: 1) darkest auto 2) partial end of vehicle 3) back end of van RL + random sampling (t=0.1) : 1) back of van 2) back of bus 3) back of van 4) back of van 5) back of van XE + random sampling (t=0.1) : 1) back of the van 2) back of the truck 3) back of the truck 4) back of the car 5) back of the truck RL + random sampling (t=0.9) : 1) back of van 2) back of bus 3) back of truck 4) dark van 5) back of van XE + random sampling (t=0.9) : 1) closest van 2) further away car 3) half bus 4) truck 5) between women blue van and white half

Figure 11: Examples of objects and sets of expressions drawn from RefCOCO and RefCOCO+ datasets decoded with random sampling with varying temperature values. Human written expressions are also presented.

tions. On the contrary, nucleus sampling addresses this issue by dynamically distilling the learned distribution.

8.2. Maximization-based Decoding Methods

The cross-entropy loss (see Equation 16) is minimized when the learned distribution concentrates to the correct ground-truth token. This, ideally, leads to a peaked probability distribution. Hence, the maximization-based decoding methods assume that the model assigns higher probability to higher quality output, and thus they strive to find the sequence with the highest probability tokens. However, the model’s high-confidence over regions of the vocabulary overestimates the use of frequent words resulting in repetition of common words and phrases. Thus, we first investigate how the model’s confidence affects the trade-off between diversity and accuracy for beam search

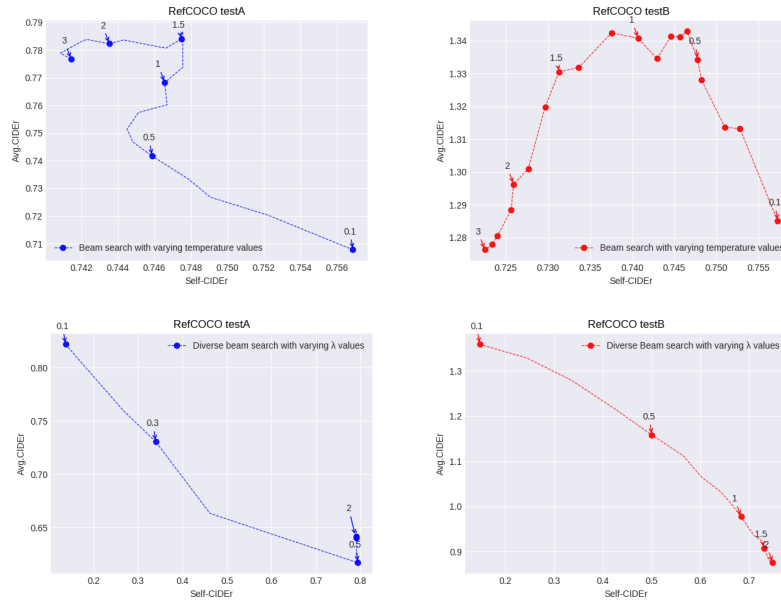


Figure 12: Self-CIDEr and average CIDEr scores for beam search and diverse beam search with varying temperature and diversity strength values for RefCOCO dataset.

by varying the softmax temperature. Figure 12 (top left and right) shows how the temperature modulates the quality-diversity trade-off for RefCOCO dataset. We first observe that unlike the random-based methods, lowering the temperature increases the diversity. As temperature increases (≤ 1.5), beam search generates sets with higher average CIDEr. Further increase in temperature (> 2) hurts both accuracy and diversity.

Next we investigate how diverse beam search (see Section 5.1.3), a diversity-promoting variant of beam search modulates the accuracy-diversity trade-off. The trade-off is controlled by the diversity strength parameter λ , which we vary between $[0.1, 2]$. We follow Vijayakumar et al. (2016b) and we set the number of groups equal to the beam width (i.e. 5). In Figure 12 (bottom left and right) we observe that as in sampling with temperature, lowering the λ values decreases the diversity. Comparing the DBS with BS with temperature, we notice that for same average CIDEr values BS achieves higher diversity.

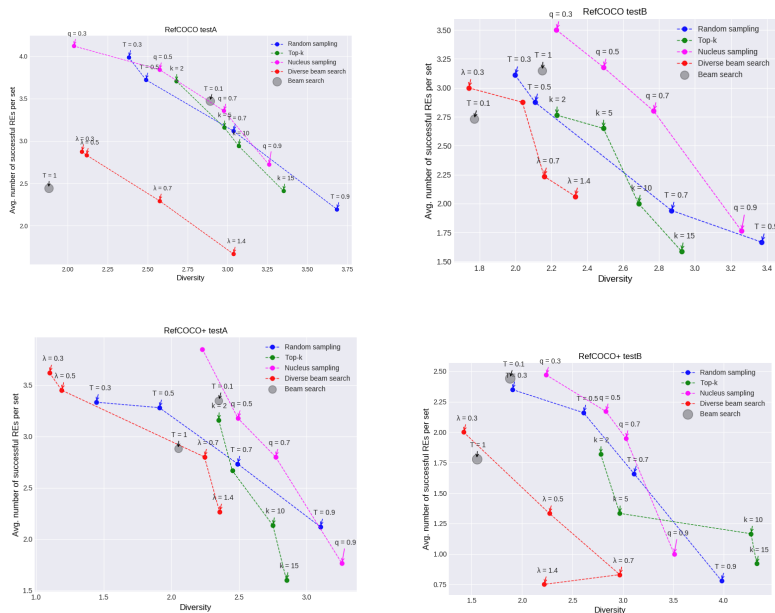


Figure 13: Human judgment scores for quality and diversity for different hyperparameter configurations.

Random Sampling	$T \in [0.3, 0.5, 0.7, 0.9]$
top- k Sampling	$k \in [2, 5, 10, 15]$
Nucleus Sampling	$q \in [0.3, 0.5, 0.7, 0.9]$
Diverse beam Search	$\lambda \in [0.3, 0.6, 0.9, 1.4]$
Beam search	$T \in [0.1, 1]$

Table 8: Hyperparameter configurations used in our human evaluation for each of the decoding strategies.

8.3. Human evaluation of sets of Referring Expressions

The analysis performed in the previous sections gave us vital insights into how the different decoding methods move in the accuracy-diversity space. However, human evaluation is still required to measure the quality and the diversity of the generated expressions. Thus, we conducted human evaluation in order to evaluate the performance of the decoding algorithms along the entire quality-diversity space. In other words, the objective of our human evaluation is to measure the effect that diversity has on the quality of

sets of referring expressions. The decoding strategies used along with the chosen hyperparameters are shown in Table 8. As language model we used the transformer model described in Section 3.3 trained with MLE. Our human evaluation protocol is the following. First, we randomly selected 25 objects and for each object we created a set of 5 expressions. Second, we showed each object along with a set of referring expressions to three human annotators. For each of the expressions within a set, human annotators were asked to evaluate whether or not the expression describes the referent object unambiguously. An expression was considered successful if two annotators agreed that the object is described unambiguously by the expression. The quality score of a set is the number of successful referring expressions it contains, while the overall quality score for a hyperparameter configuration is the average number of successful referring expressions of all sets. Furthermore, human annotators were asked to give a diversity score (from 1 to 5, the higher the better) for each set. The diversity score of a set is the average score given by the three human annotators, while the diversity score for each hyperparameter configuration is the average diversity score of all sets.

Figure 13 presents the results of our human evaluation study. We first note that beam search and diverse beam search do not produce sets with the highest generation quality. Nucleus sampling with $q = 0.3$ consistently produces sets that have the highest quality ratings in both datasets. A natural question that arises is why maximization-based algorithms underperform when it comes to the generation of a high quality set. Figure 14 shows examples of referent objects and the associated sets of referring expressions for both decoding strategies for different hyperparameters. We observe that both decoding strategies generate duplicate expressions within a set that contain incorrect or shorter expressions that do not convey enough information to facilitate the identification of the target object. Thus, reducing the overall quality of the set. Furthermore, comparing the default softmax temperature for beam search ($T = 1$) with a sharper distribution ($T = 0.1$), we observe that the latter produces sets that have higher quality and diversity. We hypothesize that reducing the temperature, leads to the exploration and expansion of hypotheses that do not stem from one predominant root hypothesis. This is consistent with the examples presented in Figure 14.

Furthermore, we observe that the quality of the sets varies significantly for different levels of diversity for all decoding algorithms. The diversity of the sets when aligned with the quality is comparable between all the randomization-based decoding algorithms. It is at the extremes of their hy-





RefCOCO testA	RefCOCO+ testA	RefCOCO testB	RefCOCO+ testB
			
Beam search (t=0.1) : 1) white shirt 2) gray shirt 3) man in gray shirt 4) girl in white 5) girl standing Beam search (t=1) : 1) white shirt 2) white shirt standing 3) man in white 4) gray tshirt 5) gray shirt Diverse beam search (λ=0.3) : 1) white shirt 2) white shirt 3) white shirt 4) white shirt 5) standing Diverse beam search (λ=0.6) : 1) white shirt 2) white shirt 3) guy standing 4) standing 5) far right person Diverse beam search (λ=0.9) : 1) girl in gray 2) gray shirt standing 3) guy standing 4) white shirt standing 5) man on left Diverse beam search (λ=1.4) : 1) white shirt 2) gray tshirt 3) white shirt guy 4) man in white 5) far right person	Beam search (t=0.1) : 1) big elephant 2) elephant on left 3) elephant in front 4) front elephant 5) bigger elephant Beam search (t=1) : 1) big elephant 2) front elephant 3) elephant 4) elephant 5) elephant on left Diverse beam search (λ=0.3) : 1) big elephant 2) big elephant 3) big elephant 4) elephant 5) elephant Diverse beam search (λ=0.6) : 1) big elephant 2) big elephant 3) the big elephant 4) elephant 5) front elephant Diverse beam search (λ=0.9) : 1) big elephant 2) big elephant 3) the big elephant 4) elephant 5) front elephant Diverse beam search (λ=1.4) : 1) big elephant 2) the big elephant 3) elephant 4) front 5) right	Beam search (t=0.1) : 1) man with glasses 2) black shirt 3) guy with glasses 4) the man with glasses 5) bald guy Beam search (t=1) : 1) man in black 2) man in black 3) black shirt 4) man in black 5) man in black Diverse beam search (λ=0.3) : 1) black shirt 2) black shirt 3) black shirt 4) black shirt 5) man with glasses Diverse beam search (λ=0.6) : 1) man with glasses 2) black shirt 3) glasses 4) black shirt 5) glasses Diverse beam search (λ=0.9) : 1) sunglasses 2) glasses 3) black shirt 4) black shirt 5) glasses Diverse beam search (λ=1.4) : 1) sunglasses 2) glasses 3) glasses 4) black shirt 5) guy	Beam search (t=0.1) : 1) brown cow 2) brown cow 3) the one with the white face 4) closest cow 5) closest cow Beam search (t=1) : 1) brown cow 2) brown cow 3) cow 4) cow 5) cow Diverse beam search (λ=0.3) : 1) brown cow 2) brown cow 3) brown cow 4) cow 5) cow Diverse beam search (λ=0.6) : 1) brown cow 2) brown cow 3) brown cow 4) cow 5) cow Diverse beam search (λ=0.9) : 1) cow 2) brown cow 3) brown cow 4) brown 5) brown Diverse beam search (λ=1.4) : 1) closest cow 2) white 3) brown 4) brown 5) light brown cow

Figure 14: Examples of objects and sets of expressions drawn from RefCOCO and RefCOCO+ datasets. The expressions were decoded with beam search and diverse beam search.

perparameters range, where the decoding algorithms heavily affect sampling that their performance diverges. Based on the results shown in Figure 13 the following observations can be made:

- Higher diversity results in lower human judgement scores for quality.
- Nucleus sampling produces sets with higher quality for the same level of diversity between all the decoding strategies, with random sampling performing second best, followed closely by top- k sampling.

- Diverse beam search produces consistently sets with the least diversity.
- Beam search produces higher quality and diversity sets when the soft-max temperature is set to $T = 0.1$ compared to the default value. Interestingly, it produces sets with higher diversity than diverse beam search.

9. Conclusions

There are three building blocks for neural REG models that follow the encoder-decoder architecture: (1) the network architecture; (2) the decoding algorithm; and (3) the learning strategy. In this work, we explored how the choices for each of the three building blocks affect the generation of one-shot expressions as well the generation of sets of referring expressions. First, we demonstrated the benefits of incorporating an object attention mechanism in the language model. Our approach allows the attention mechanism to be calculated at the level of the referent object. We demonstrated that applying this approach to REG results in significant benefits compared to the standard LSTM model. Our results on RefCOCO and RefCOCO+ shows an increase, on average, of 0.26 and 0.12 in CIDEr scores respectively. Our qualitative analysis showed that the attention mechanism results in an improvement in determining fine appearance attributes of the target object as well as an improvement in expressing the absolute and relative location of the target object. Unlike the standard LSTM, the proposed attention mechanism allows the language model to consider all the information pertaining the referent object at once. In other words, instead of letting the language model to hallucinate over the attributes of the target object, the attention mechanism enables the language model to take multiple glimpses of the salient parts of the object’s region during generation. Our human evaluation study showed that the proposed model performs comparable with the state-of-the art (Yu et al., 2017). In particular, in RefCOCO+ testA it achieves an increase from 58.85% to 76.66% in task success.

Furthermore, to demonstrate the benefits of attention in REG, we carefully devised a transformer architecture that is noticeably effective in REG. We showed that reducing the depth of the network from 6 layers to 3 results in an improvement in automatic metrics. Moreover, we proposed a different connectivity pattern between the encoder and the decoder, by connecting each layer of the encoder with the respective decoder layer. Our results on

RefCOCO and RefCOCO+ datasets demonstrate significant improvements over the standard architecture. We also presented qualitative examples of how the proposed connectivity improves the spatial awareness and the inference of fine appearance attributes. Our human evaluation study showed that the proposed transformer produces expressions that are more human-like, accurate and describing the main aspects of the target object better than the proposed attention LSTM. In addition, our results in task success improves over the state-of-the-art results in RefCOCO testA from 76.95% to 78.33% and in RefCOCO+ testA from 58.85% to 80.00%.

Next, we presented a simple and efficient approach to effectively train our language models on non-differentiable sequence metrics. Our approach is a variation of the popular REINFORCE algorithm that utilizes multiple samples per input to normalize the reward that it observes. We showed that the proposed approach reduces the variance of the gradient more effectively compared to the self-critical sequence training. Empirically we found that directly optimizing the CIDEr metric is highly effective. Our human evaluation results on RefCOCO and RefCOCO+ dataset establish a new state-of-the art. We improved the results in RefCOCO testA and testB from 76.95% to 81.66% and from 78.10% to 83.33% respectively. While in RefCOCO+ testA we improved the best results from 58.85% to 83.33%.

Finally, the choice of the decoding strategy is critical in controlling the trade-off between quality and diversity. Thus, we evaluated the ability of existing decoding algorithms to generate sets of referring expressions by comparing their performance along the entire quality-diversity space. We introduced the first large-scale human evaluation study in REG, that compares the quality of sets of referring expressions at the same levels of diversity. We found that beam search produces sets with higher quality and diversity when the softmax temperature is set to $T = 0.1$ compared to the default value $T = 1$. Second, both beam search and diverse beam search result in less successful expressions per set compared to the rest decoding algorithms at equal points of diversity. We showed that duplicate wrong expressions within the sets reduce the quality significantly. Finally, our findings suggest that nucleus sampling, produces higher quality sets at the same levels of diversity amongst the compared decoding strategies, with random sampling performing second best, followed by top- k sampling.

Acknowledgements

We wish to thank NVIDIA for its kind donation of the GPU used in the experiments. Dimitra Gkatzia is supported by the EPSRC grant CiViL: EP/T014598/1.

References

- P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, L. Zhang, Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering, in: CVPR, 2018.
- J. L. Ba, J. R. Kiros, G. E. Hinton, Layer Normalization, 2016.
- S. Banerjee, A. Lavie, METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments, in: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Association for Computational Linguistics, Ann Arbor, Michigan, 65–72, 2005.
- S. Bengio, O. Vinyals, N. Jaitly, N. Shazeer, Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks, in: Advances in Neural Information Processing Systems, 2015.
- B. Bohnet, The Fingerprint of Human Referring Expressions and their Surface Realization with Graph Transducers, in: Proceedings of the Fifth International Natural Language Generation Conference, 2008.
- T. Castro Ferreira, E. Krahmer, S. Wubben, Individual Variation in the Choice of Referential Form, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 423–427, doi:10.18653/v1/N16-1048, URL <https://www.aclweb.org/anthology/N16-1048>, 2016a.
- T. Castro Ferreira, E. Krahmer, S. Wubben, Towards more variation in text generation: Developing and evaluating variation models for choice of referential form, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics

- (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 568–577, doi:10.18653/v1/P16-1054, URL <https://www.aclweb.org/anthology/P16-1054>, 2016b.
- T. Castro Ferreira, C. van der Lee, E. van Miltenburg, E. Krahmer, Neural data-to-text generation: A comparison between pipeline and end-to-end architectures, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 552–562, doi:10.18653/v1/D19-1052, URL <https://www.aclweb.org/anthology/D19-1052>, 2019.
- Y. Chen, K. Cho, S. R. Bowman, V. O. Li, Stable and Effective Trainable Greedy Decoding for Sequence to Sequence Learning, URL <https://openreview.net/forum?id=rJZ1KFkvM>, 2018.
- K. Cho, Noisy Parallel Approximate Decoding for Conditional Recurrent Language Model, CoRR abs/1605.03835, URL <http://arxiv.org/abs/1605.03835>.
- K. Cho, B. van Merriënboer, D. Bahdanau, Y. Bengio, On the Properties of Neural Machine Translation: Encoder–Decoder Approaches, in: Proceedings of the Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8), 2014.
- R. Dale, Cooking up Referring Expressions, in: Proceedings of the 27th Annual Meeting on Association for Computational Linguistics, 1989.
- R. Dale, E. Reiter, Computational interpretations of the Gricean maxims in the generation of referring expressions, *Cognitive Science* 19 (2) (1995) 233–263, ISSN 0364-0213, doi:10.1016/0364-0213(95)90018-7.
- G. Di Fabbri, A. Stent, S. Bangalore, Trainable Speaker-Based Referring Expression Generation, in: Proceedings of the Twelfth Conference on Computational Natural Language Learning (CoNLL), 2008.
- A. Fan, M. Lewis, Y. Dauphin, Hierarchical Neural Story Generation, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018.

- T. C. Ferreira, I. Paraboni, Referring Expression Generation: Taking Speakers' Preferences into Account, in: Proceedings of the International Conference on Text, Speech and Dialogue, 2014.
- J. Fidler, Y. Goldberg, Controlling Linguistic Style Aspects in Neural Language Generation, in: Proceedings of the Workshop on Stylistic Variation, Association for Computational Linguistics, Copenhagen, Denmark, 94–104, doi:10.18653/v1/W17-4912, URL <https://www.aclweb.org/anthology/W17-4912>, 2017.
- I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, The MIT Press, ISBN 0262035618, 2016.
- H. P. Grice, Logic and Conversation, in: P. Cole, J. L. Morgan (Eds.), Syntax and Semantics: Vol. 3: Speech Acts, Academic Press, New York, 41–58, URL <http://www.ucl.ac.uk/ls/studypacks/Grice-Logic.pdf>, 1975.
- H. Guo, R. Pasunuru, M. Bansal, Soft Layer-Specific Multi-Task Summarization with Entailment and Question Generation, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), 2018.
- K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 770–778, doi:10.1109/CVPR.2016.90, URL <https://doi.org/10.1109/CVPR.2016.90>, 2016.
- K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Comput. 9 (8) (1997) 1735–1780, ISSN 0899-7667, doi:10.1162/neco.1997.9.8.1735, URL <https://doi.org/10.1162/neco.1997.9.8.1735>.
- A. Holtzman, J. Buys, L. Du, M. Forbes, Y. Choi, The Curious Case of Neural Text Degeneration, in: International Conference on Learning Representations, URL <https://openreview.net/forum?id=rygGQyrFvH>, 2020.

- A. Holtzman, J. Buys, M. Forbes, A. Bosselut, D. Golub, Y. Choi, Learning to Write with Cooperative Discriminators, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018.
- D. Ippolito, R. Kriz, J. Sedoc, M. Kustikova, C. Callison-Burch, Comparison of Diverse Decoding Methods from Conditional Language Models, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019a.
- D. Ippolito, R. Kriz, J. Sedoc, M. Kustikova, C. Callison-Burch, Comparison of Diverse Decoding Methods from Conditional Language Models, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019b.
- L. C. Jain, L. R. Medsker, Recurrent Neural Networks: Design and Applications, CRC Press, Inc., Boca Raton, FL, USA, 1st edn., ISBN 0849371813, 1999.
- S. Kazemzadeh, V. Ordonez, M. Matten, T. Berg, ReferItGame: Referring to Objects in Photographs of Natural Scenes, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.
- M. Kilickaya, A. Erdem, N. Ikizler-Cinbis, E. Erdem, Re-evaluating Automatic Metrics for Image Captioning, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Association for Computational Linguistics, Valencia, Spain, 199–209, URL <https://www.aclweb.org/anthology/E17-1019>, 2017.
- D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, URL <http://arxiv.org/abs/1412.6980>, cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015, 2014.
- P. Koehn, Pharaoh: A Beam Search Decoder for Phrase-Based Statistical Machine Translation Models, in: R. E. Frederking, K. B. Taylor (Eds.), Machine Translation: From Real Users to Research, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-540-30194-3, 115–124, 2004.

- W. Kool, H. van Hoof, M. Welling, Buy 4 REINFORCE Samples, Get a Baseline for Free!, in: Deep Reinforcement Learning Meets Structured Prediction, ICLR, 2019.
- E. Kraehmer, K. van Deemter, Computational Generation of Referring Expressions: A Survey, *Comput. Linguist.* 38 (1) (2012) 173–218.
- A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in: Proceedings of the 25th Conference on Advances in Neural Information Processing Systems, 2012.
- J. Li, D. Jurafsky, Mutual Information and Diverse Decoding Improve Neural Machine Translation., *CoRR* abs/1601.00372, URL <http://dblp.uni-trier.de/db/journals/corr/corr1601.html>LiJ16.
- J. Li, D. Jurafsky, Mutual Information and Diverse Decoding Improve Neural Machine Translation., *CoRR* abs/1601.00372, URL <http://dblp.uni-trier.de/db/journals/corr/corr1601.html>LiJ16.
- J. Li, W. Monroe, D. Jurafsky, Learning to Decode for Future Success, *CoRR* abs/1701.06549, URL <http://arxiv.org/abs/1701.06549>.
- J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, J. Gao, Deep Reinforcement Learning for Dialogue Generation, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2016.
- T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common Objects in Context, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), *Computer Vision – ECCV*, 2014.
- R. Luo, G. Shakhnarovich, Comprehension-Guided Referring Expressions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
- J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, K. Murphy, Generation and Comprehension of Unambiguous Object Descriptions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016a.

- J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, K. Murphy, Generation and Comprehension of Unambiguous Object Descriptions, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016b.
- M. Mitchell, J. Dodge, A. Goyal, K. Yamaguchi, K. Stratos, X. Han, A. Mensch, A. Berg, T. Berg, H. Daumé III, Midge: Generating Image Descriptions From Computer Vision Detections, in: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Avignon, France, 747–756, URL <https://www.aclweb.org/anthology/E12-1076>, 2012.
- A. Mnih, D. J. Rezende, Variational Inference for Monte Carlo Objectives, in: Proceedings of the 33rd International Conference on International Conference on Machine Learning, 2016.
- S. Narayan, C. Gardent, Deep Learning Approaches to Text Production, *Synthesis Lectures on Human Language Technologies* 13 (1) (2020) 1–199, doi:10.2200/S00979ED1V01Y201912HLT044, URL <https://doi.org/10.2200/S00979ED1V01Y201912HLT044>.
- K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: A Method for Automatic Evaluation of Machine Translation, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL), ACL '02, 2002.
- M. Ranzato, S. Chopra, M. Auli, W. Zaremba, Sequence Level Training with Recurrent Neural Networks, in: Proceedings of the 4th International Conference on Learning Representations ICLR, 2016.
- S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, V. Goel, Self-Critical Sequence Training for Image Captioning, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
- J. Schulman, P. Moritz, S. Levine, M. Jordan, P. Abbeel, High-Dimensional Continuous Control Using Generalized Advantage Estimation, in: Proceedings of the International Conference on Learning Representations (ICLR), 2016.
- I. Sutskever, O. Vinyals, Q. V. Le, Sequence to Sequence Learning with Neural Networks, in: Proceedings of the 27th International Conference on Neural Information Processing Systems, 2014.

- R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, The MIT Press, second edn., URL <http://incompleteideas.net/book/the-book-2nd.html>, 2018.
- J. Tan, X. Wan, J. Xiao, Abstractive Document Summarization with a Graph-Based Attentional Neural Model, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), 2017.
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is All you Need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 5998–6008, URL <http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>, 2017.
- R. Vedantam, C. L. Zitnick, D. Parikh, CIDEr: Consensus-based image description evaluation., in: CVPR, 2015.
- J. Viethen, R. Dale, Speaker-Dependent Variation in Content Selection for Referring Expression Generation, in: Proceedings of the Australasian Language Technology Association Workshop 2010, 2010a.
- J. Viethen, R. Dale, Speaker-Dependent Variation in Content Selection for Referring Expression Generation, in: Proceedings of the Australasian Language Technology Association Workshop, 2010b.
- A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee, D. J. Crandall, D. Batra, Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models, CoRR abs/1610.02424, URL <http://arxiv.org/abs/1610.02424>.
- A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee, D. J. Crandall, D. Batra, Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models, CoRR abs/1610.02424, URL <http://arxiv.org/abs/1610.02424>.
- O. Vinyals, Q. V. Le, A Neural Conversational Model, CoRR abs/1506.05869, URL <http://arxiv.org/abs/1506.05869>.

- O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
- Q. Wang, A. B. Chan, Describing like humans: on diversity in image captioning, CoRR abs/1903.12020, URL <http://arxiv.org/abs/1903.12020>.
- R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, in: Machine Learning, 229–256, 1992.
- K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, in: Proceedings of the 32nd International Conference on Machine Learning, 2015.
- L. Yu, P. Poirson, S. Yang, A. C. Berg, T. L. Berg, Modeling Context in Referring Expressions, in: Proceedings of the 14th European Conference on Computer Vision (ECCV), 2016.
- L. Yu, H. Tan, M. Bansal, T. L. Berg, A Joint Speaker-Listener-Reinforcer Model for Referring Expressions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR, 2017.
- W. Zaremba, I. Sutskever, Reinforcement Learning Neural Turing Machines, CoRR abs/1505.00521, URL <http://arxiv.org/abs/1505.00521>.
- S. Zarrieß, D. Schlangen, Easy Things First: Installments Improve Referring Expression Generation for Objects in Photographs, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 610–620, doi:10.18653/v1/P16-1058, URL <https://www.aclweb.org/anthology/P16-1058>, 2016.
- S. Zarrieß, D. Schlangen, Decoding Strategies for Neural Referring Expression Generation, in: Proceedings of the 11th International Conference on Natural Language Generation, Association for Computational Linguistics, Tilburg University, The Netherlands, 503–512, 2018.
- S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, J. Weston, Personalizing Dialogue Agents: I have a dog, do you have pets too?, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018.